The Semantic Web and Policy ⋆

Lalana Kagal [a] James Hendler [b] Tim Berners-Lee [a]

[a] *Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA*
[b] *Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY*

1. Introduction

As Semantic Web technologies mature and become more accepted by researchers and developers alike, the widespread growth of the Semantic Web seems inevitable. However, this growth is currently hampered by the lack of well-defined security protocols and specifications. Though the Web does include fairly robust security mechanisms, they do not translate appropriately to the Semantic Web as they do not support autonomous machine access to data and resources and usually require some kind of human input [9]. Also, the ease of retrieval and aggregation of distributed information made possible by the Semantic Web raises privacy questions as it is not always possible to prevent misuse of sensitive information [12]. In order to realize it's full potential as a powerful distributed model for publishing, utilizing, and extending information, it is important to develop security and privacy mechanisms for the Semantic Web. Policy frameworks built around machine-understandable policy languages, with their promise of flexibility, expressivity and automatable enforcement appear to be the obvious choice.

The term **policy** is very broad and can mean different things in different contexts. We consider a policy to be a declarative set of rules that guide the behavior of entities within a system. Policies are a way to implement flexible security for dynamic and distributed environments where the system behavior may need to be modified without re-implementing the system or reconfiguring the requirements. As distributed information systems become more ubiquitous, autonomous, and complex, there is a stronger need for grounding them on common models of data and knowledge such as those provided by Semantic Web technologies. The entities (software and human) in such systems need to be able to exchange information, queries and requests with some assurance that they share a common meaning. This is critical not only for the data but also for the security policies. This is especially important if the policy is shared between multiple domains that must adhere to or enforce the policy even though they have their own native schemas or data models. Employing Semantic Web techniques for modeling and reasoning about information in policy frameworks will provide the required shared semantics.

It is clear that these two technologies - Semantic Web and Policy - complement each other and together will give rise to security infrastructures that provide more flexible management, are able to accommodate heterogeneous information, have improved communication, and are able to dynamically adapt to variations in the environment. These infrastructures could be used for a wide spectrum of applications ranging from network management, quality of information, to security, privacy and trust. This special issue of the *Journal of Web Semantics* is focussed on the impact of Semantic Web technologies on policy management, and the specification, analysis and application of these Semantic Web-based policy frameworks.

⋆ Partial support provided by NSF award 0524481 and IARPA award FA8750-07-2-0031

Email addresses: lkagal@csail.mit.edu (Lalana Kagal), hendler@cs.rpi.edu (James Hendler), timbl@csail.mit.edu (Tim Berners-Lee).

2. Papers

The five papers that were selected for this issue represent an overview of emerging research in Semantic Web-based policy languages and reasoning in different application areas. One paper deals with policies for information quality assessment, two with controlling access to resources and information in mobile computing environments, one with obligation policies for processing event streams, and one with preferences for the selection of Web services.

In the first paper, Bizer and Cyganiak explain that the Web allows massive amounts of information to be easily accessed and consumed making the identification of high quality information of utmost importance. They suggest that information to be used for a certain task be assessed based on task-specific criteria, the provenance of the information and the attributes of the information provider such as their reputation and intentions. Bizer and Cyganiak define the process of information quality assessment as measuring the quality dimensions that are relevant to an information consumer and comparing the assessment results with the consumers quality requirements. In order to support this assessment, they propose that user-specific policies be used to filter out data of low quality and describe how their framework, WIQA, enables quality-driven information filtering by enforcing these policies. The WIQA framework consists of two components - a graph store and a filtering and explanation engine. WIQA supports named graphs, which are an extension of the RDF data model where RDF graphs are named with URI references. Using named graphs allows the framework to maintain and track provenance and other meta-information of the data that would have been otherwise difficult to manage. WIPA-PL, which is based on SPARQL Query Language for RDF (SPARQL) [11], is used to express policies over these named graphs and the user's preferences. The engine makes decisions about information quality by reasoning over these policies. The WIQA framework is able to provide an explanation for the assessment enabling users to understand and trust the results. The authors suggest that integrating WIQA framework with search engines would improve searching by providing information quality assessment of the search results.

While Bizer et al. are mostly concerned about quality of information, Agarwal, Lamparter and Studer explore algorithms that facilitate the automated selection and negotiation of services. They suggest that, as service-oriented architectures are made up of services from different organization and might be offered under different configurations, service-oriented computing requires a mechanism for coordination between service requesters and providers. This co-ordination is provided by selection and negotiation algorithms, and service matchmaking that depend on the preferences of both the service providers and requesters. Agarwal et al. present a formalism for specifying policies on Web service properties in order to facilitate this co-ordination. Their criticism of current policy languages is that they cannot express constraints on the behavior (choreography, orchestration and communication) of Web services and are not able to express detailed preference information that captures trade-offs between the different possible Web service configurations. An interesting notion the authors describe is that of "soft constraints". Usually requirements are considered boolean constraints, meaning if a service does not fulfill the constraint it is not matched. However, often users' constraints are not so hard and users should be able to represent these soft constraints by specifying their preferences with respect to different alternatives. Utility function policies represent the functional relationship between alternatives and their values. The utility value associated with a Web service description is used in the matchmaking and negotiation algorithms. The authors define a policy ontology that supports utility functions and is augmented with temporal logic that enables soft constraints to be expressed. Web service policies are represented in a DL-safe subset of Semantic Web Rule Language (SWRL) [7] over the policy ontology and Web service description.

In the next paper, Perich and McHenry address security concerns of cognitive software-defined radios. Usually radios are manufactured for specific applications and to satisfy requirements defined by regulators, rendering them useless for other applications. Cognitive software-defined radios (SDR) overcome this problem as they can be dynamically configured according to current regulatory, user, application and environmental restrictions. However, they also pose security risks as they could be modified by a malicious user or through a malfunction to interfere with services such as aviation or global positioning. The authors propose the use of declarative policies for controlling the scope of operations of a SDR. These policies limit spectrum access opportu-

nities that are currently available such as frequencies, bandwidth, power level, or modulation techniques that the SDR can use to transmit given its current environment. The policy conformance and enforcement components, which ensure that the device does not violate the policies, consist of the policy manager, database, policy conformance reasoner and policy enforcer. The policy manager maintains the database of policies and responds to status requests and commands to modify the policies. The policy enforcer uses the policy conformance reasoner to evaluate policies and ensures that the device's configuration conforms with regulatory and system policies. These policies are expressed in the XG policy language, which is based on Semantic Web Rule Language (SWRL) [7] and are expressed over data annotated with the XG ontology, which is in Web Ontology Language (OWL) [2]. As policy conflicts are possible, the authors also provide a meta-level ontology for defining absolute and relative prioritization of policies. One of the main challenges, which they overcame successfully, was to implement all the policy components on a small, general-purpose processor embedded within an XG radio and tightly integrate them with the accredited kernel on the device. The authors also field-tested policy-enabled XG radios, which were dynamically configured in the presence of other radio systems, with promising results.

As opposed to access control policies that Perich and McHenry focus on, the next paper in this issue discusses obligation policies that determine actions that must be performed under a certain set of circumstances. Liu, Ranganathan and Riabova propose the use of obligation policies to perform appropriate processing on streams of data. Their focus is on large-scale distributed systems that produce high-volume streams of unstructured low-level data including text and multimedia from sources such as surveillance video cameras and emergency radio broadcasts. These streams usually require additional processing such as filtration, aggregation and classification to extract useful information from them. Obligation policies, the authors suggest, provide the right mechanism. However, the authors argue that existing policy frameworks are unsuitable and propose the Eagle policy language. Eagle policies describe high-level events in the form of RDF patterns that are associated with conditions, which need to be checked, and actions, which need to be performed once the pattern is matched. The policies are described over high-level events so there is still the need to obtain high-level events from the low-level events in the stream. This is done by processing graphs that are automatically constructed from Eagle policies by the enforcement framework. This approach exploits the semantics of policies, sources and processing elements (that perform operations on the event stream) that are all expressed in OWL [2]. The policy framework uses Description Logic Programs (DLP) [1] to compare the requirements of the policy with the processing elements and uses planning techniques to generate processing graphs that consist of event sources and processing elements interconnected by event streams. Stream monitors are responsible for enforcing the policy. They monitor the events on the high-level streams and if the conditions of a policy are satisfied they perform the associated actions.

Rao, Sardinha and Sadeh stress the importance of being able to dynamically identify and access relevant information while enforcing context-sensitive policies in open environments. They suggest that users require increasingly richer policies for both security and privacy and that the ability to include context such as current activity, user location, and relationships is becoming essential for policies. The authors define context-sensitive policies as policies whose conditions are not tied to static considerations but conditions whose satisfaction, given the very same actors, will likely fluctuate over time. This variance makes enforcing these policies challenging as relevant sources of information vary from user to user, information sources for the same user might vary over time, and it might not be possible to pre-determine sources of information. The authors propose the use of Web services with rich profiles to represent information sources allowing for dynamic discovery of relevant information. Their framework, which has been successfully deployed on Carnegie Mellon's campus for pervasive applications, consists of Web services representing information sources and a model for dynamically interleaving policy reasoning and information selection. In their framework, a Web service is protected by an information disclosure agent that enforces both access control policies, which control who can access the service, and obfuscation policies, which manipulate the level of accuracy or inaccuracy of information being disclosed. The agent can use both local and external information via other Web services to make policies decisions. The agent also contains a meta-controller that allows different orchestration strategies to be used, from simple control flows to

more sophisticated processes. The architecture does not require a specific policy language or reasoner and supports different policy reasoning engines. Rao et al. demonstrate this flexibility by describing their implementation that consists of both Jess reasoners for ROWL (OWL-Lite extension for rules) policies [5] and reasoners for policies in eXtensible Access Control Markup Language (XACML) [6].

3. Conclusion

Though our authors were unified in their belief that using Semantic Web technologies for policies had significant benefits, interestingly enough each paper (other than Rao et al.) proposed a unique approach to representing policies with the authors arguing that their application area had specific requirements that could not be met by existing policy languages. Standardization efforts, like those involved in the development of XACML [6] have generally not provided a formal semantics, making it difficult to use rule languages and Semantic Web reasoners. While there have been efforts to formalize these models [8], there is much work left to be done in this area. We believe that new Web standards such as the Rule Interchange Format (RIF) [10] and rule languages such as N3Logic [3] will significantly impact current research by encouraging more focus on previously under-appreciated topics such as policy interoperability and re-use. This in turn will help researchers understand the requirements of [4] and appreciate the importance of standardization of policy languages. The five papers in this issue provide a exciting look at the current Semantic Web and Policy landscape and we look forward to further developments in this research area.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. PatelSchneider (eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, New York, NY, USA, 2003.

[2] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, L. Stein, P. Patel-Schneider, Web Ontology Language Reference (OWL), W4C Recommendation 10 February 2004, http://www.w3.org/TR/owl-ref/ (2004).

[3] T. Berners-Lee, D. Connolly, L. Kagal, J. Hendler, Y. Schraf, N3Logic: A Logical Framework for the World Wide Web, Journal of Theory and Practice of Logic Programming (TPLP), Special Issue on Logic Programming and the Web 2008, http://arxiv.org/abs/0711.1533.

[4] P. A. Bonatti, C. Duma, N. Fuchs, W. Nejdl, D. Olmedilla, J. Peer, N. Shahmehri, Semantic Web Policies - A Discussion of Requirements and Research Issues, in: 3rd European Semantic Web Conference (ESWC), 2006.

[5] F. L. Gandon, M. Sheshagiri, N. M. Sadeh, ROWL: Rule Language in OWL and Translation Engine for JESS, http://www.cs.cmu.edu/~sadeh/MyCampusMirror/ROWL/ROWL.html (2004).

[6] S. Godik, T. Moses, OASIS eXtensible Access Control Markup Language (XACML), OASIS Committee Secification cs-xacml-specification-1.0, November 2002.

[7] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission 21 May 2004, http://www.w3.org/Submission/SWRL/ (2004).

[8] V. Kolovski, J. Hendler, B. Parsia, Analyzing web access control policies, in: WWW '07: Proceedings of the 16th international conference on World Wide Web, ACM, New York, NY, USA, 2007.

[9] D. Olmedilla, Security and Privacy on the Semantic Web, in: M. Petkovic, W. Jonker (eds.), Security, Privacy and Trust in Modern Data Management, Data-Centric Systems and Applications, Springer, 2007.

[10] RIF W3C Working Group, Rules Interchange Format (RIF), http://www.w3.org/2005/rules/wiki/RIF_Working_Group (2008).

[11] W3C, SPARQL RDF Query Language (SPARQL), W3C Recommendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/ (2008).

[12] D. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, G. Sussman, Information accountability, Tech. rep., MIT, http://dspace.mit.edu/handle/1721.1/37600 (June 2007).