PolicyForge: A Collaborative Environment for Formalizing Privacy Policies in Health Care

Andras Nadas,
Laszlo Juracz
and Janos Sztipanovits
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee 37212, USA
{andras.nadas, laszlo.juracz, janos.sztipanovits}@vanderbilt.edu

Mark E. Frisse
and Ann J. Olsen
Vanderbilt University Medical Center
Vanderbilt University
Nashville, Tennessee 37212, USA
mark.frisse@vanderbilt.edu
and ann.olsen@vanderbilt.edu

***Abstract*—The vision of PolicyForge.org is that it becomes an open repository for privacy policies at local, state and national level; provides collaboration services for discussing, interpreting, and tracking policies; and by embedding formal policy models with relevant ontologies, it provides a wide range of services for authoring, composing, analyzing policy models, and for exporting executable version of the models for Health Information Exchange platforms.**

***Index Terms*—Formal Specification, Privacy, Collaborative tools, Metamodeling, Health information systems.**

I. Introduction and Background

As with all aspects of modern digital life more and more data is stored, transferred and shared about individuals by large computer systems with or without adequate the protection of the privacy of the individual. Electronic Medical Record (EMR) and Health Information Systems (HIS) have particularly sensitive privacy concerns that are governed by policies from several different entities. In the United States EMR and HIS are governed by a confusing number of laws, regulations and policies from the federal, state and local governments as well as institutional policies and business agreements. The federal statutes include the Health Insurance Portability and Accountability Act (HIPAA) [1] and the Health Information Technology for Economic and Clinical Health (HITECH) Act [2]. Many states have issued different and sometimes contradicting regulations governing the use and disclosures of sensitive Patient Health Information (PHI) [3]. Ensuring consistent harmonization of policies across federal, state, and institutional levels remains a major challenge [4]. To further complicate the issue, the policies and regulations are changed and revised from time to time. Ever more stringent enforcement requirements with increasing penalties for non-compliance add to the considerable challenge for the development of health information systems.

To conquer these problems researchers have been focusing on formalization of the policies that regulate health information systems. All of these formalization strategies can be traced back to the philosophical framework of Contextual Integrity [5], that provides a clear definition of what privacy is and enables the description of it using context dependent transmission channels and dissemination rules. Based on Contextual Integrity researchers developed many formalisms that cover different slices and aspects of the policy formalization problem [6]–[12]. While all these researchers were successful and made meaningful contributions, they did not achieve wide adoption outside the research community. One reason for this is the lack of a platform where researchers and others can share their models, results and thoughts while making it available for others. Another reason for the lack of adoption is that researchers usually focus on narrow subsets of large problems and the result of this is the lack of sufficient coverage of the problem domain to be usable for outside entities. The particular reason for focusing on a narrow subset in the case of policies is the problem of translation and formalization of the natural language policies. One thing the researchers agree on is that this formalization is a slow and difficult process [13].

The PolicyForge framework we present can help to scale these efforts by the introduction of crowdsourcing. Crowdsourcing can extend the user base from just computer science researchers to a broad range of health care professionals, as well as, Institutional Privacy Officers and their internal policy development teams, inter-organizational policy development teams (e.g., for Health Information Exchanges (HIE), clinically integrated networks, Accountable Care Organizations (ACO), etc.) and legislative and regulatory agency staff at state, federal and regional levels.

II. Design of PolicyForge

The architecture of PolicyForge, as shown on Figure 1, will offer an authoritative reference source for policy makers and users. Its policy repository, the Policy Exchange, captures standard format policies, policy templates, and ontologies, includes both the textual and formal representations, their version history, the various interpretations and open issues and the provenance information. The Policy Exchange includes a taxonomy-driven search engine for all artifacts. Policy modeling, analysis and export is supported by a wide range of authoring, collaboration, and analysis tools. An essential feature of PolicyForge.org will be the requirement for the existence

978-1-4673-6282-5/13 © 2013 IEEE

SEHC 2013, San Francisco, CA, USA

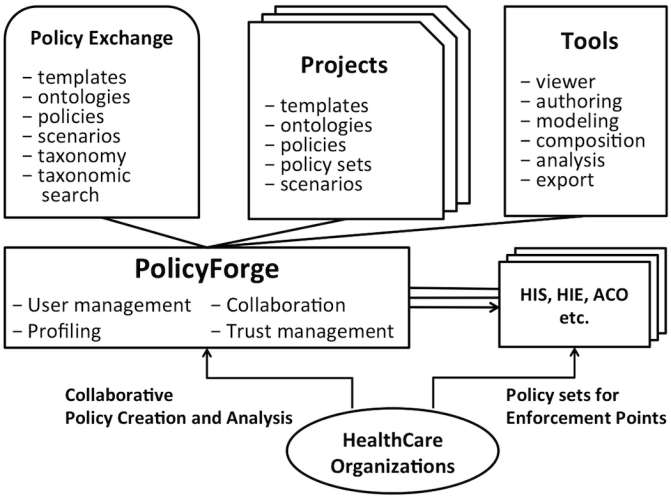

Fig. 1. The architecture of PolicyForge.org.

of formal policy models. While the formalism is not expected to be standardized, the formal specification of semantics and methods for translating formal policy models into a common semantic domain will be required. PolicyForge.org includes extensive support for security and user management. While artifacts in the Policy Exchange are public, users and user groups are able to create projects with restricted visibility and access control.

A. The Framework

PolicyForge is being developed on the novel cloud based collaboration framework of VehicleFORGE [14]. The VehicleFORGE platform is designed to step beyond the software-only forges. It is designed and maintained to host the Defense Advanced Research Project Agency (DARPA) Fast, Adaptable, Next-Generation Ground Vehicle (FANG) series of prize-based design competitions as part of the Adaptive Vehicle Make (AVM) portfolio. VehicleFORGE provides the virtual environment which enables the management of the competitions, competitors and the collaboration of geographically distributed design teams, as well as various cloud-based analysis services and tools for the design work.

The success of crowdscourcing and the model of distributed problem-solving and software production was essentially enabled by widely popular open-source software forges such as SourceForge.net in the past decade. Based on the success of SourceForge.net, their Allura framework was an obvious choice after investigating the open technologies for the development of VehicleFORGE.

B. Policy Modeling Tools

PolicyForge will come with a set of tools that enables the organization and formalization of health care privacy policies. There are tools to focus on organization, creation and sharing of all artifacts related or constructed during the formalization of privacy policies including policy texts, ontology models, policy models and scenarios.

1) Artifact Taxonomy: PolicyForge is designed to enable the sharing and editing of the artifacts connected to a policy formalization problem. These artifacts are specified and organized using an Artifact Taxonomy. This taxonomy is extensible with new artifact types to enable scaling and adaptability to new formalisms and problem areas. The artifacts are grouped into three main categories file, document and model as shown on Figure 2. The file artifacts are stored in PolicyForge but are not processed or used by any of the tools directly. The files are stored for reference only to help the users. The documents are the processable and usable abstractions of the files. The content of the documents is plain text with mark-ups to enable simple but universal processing by tools. The most extensive category of artifacts is the models, which contain the formal descriptions of the policies, scenarios and concept ontologies. Policy template models are special artifacts. These templates are used to define the formal language for the policy formalisms together with the formal semantics of the language anchored for different purposes, such as execution, analysis or verification using scenarios.

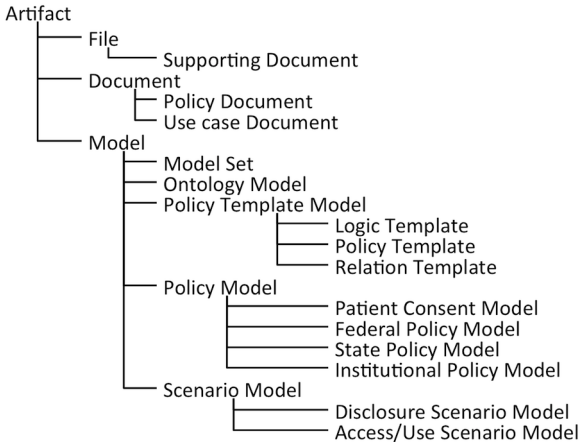

Fig. 2. The Artifact Taxonomy of PolicyForge.org.

2) Policy Text and Ontology Model Authoring: The Policy Text and Ontology Model Authoring tool enables users to add or edit the text of the rules and regulations from their organization(s) to their project in the PolicyForge. The addition of the policy texts to the PolicyForge is the first step in the process of formalization. Users can store and arrange an arbitrary number of policy texts in each project. The policy texts that are already available in the project can be browsed and reviewed anytime. The policy texts are copied from documents already existing outside PolicyForge or users can opt to upload these documents as references into the cloud storage of PolicyForge.

Parts of the texts can be marked as members of an ontology and be highlighted in the text. The marked up text is linked to the ontology model that is stored in the background together with the other models. The ontology association can later be reused during the formalization of the policy. The ontologies that the user can use to markup the text can also be visualized as a reference, as well as edited and extended. Beyond editing

the ontology models inside PolicyForge, users can opt to include ontology models in a standard RDF-OWL format [15].

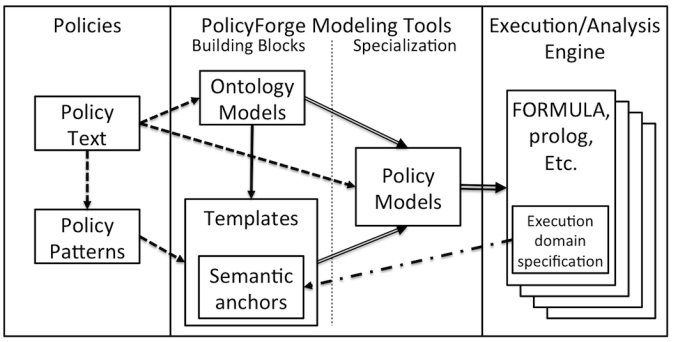

Fig. 3. The Models and their relation in PolicyForge.

3) Policy Model Construction: The Policy Models are constructed using Ontology Models and Policy Model Templates as shown in Figure 3. The Policy Model Templates are predefined templates or forms that provide a generalized structure for policy model development. The templates provide a form-like structure where instantiated ontology terms can be filled in into each field. The templates provide the structure and glue to go from Ontology Models to Policy Models.

More formally speaking, the Policy Model Templates provide a reusable structure with structural semantics enforced upon specialization of the template into Policy Models. The operational or denominational semantics are given to the templates by anchoring their semantics with a formal specification. This separation of the structural and behavioral semantics enables use of the same patterns and their instantiated models in different target domains such us analysis, verification and/or execution [16].

Policy Models are instantiations of Policy Templates with entities (actors, classes etc.) derived from Ontology Models and filled in each field of the template. Each Policy Model can contain a hierarchy of instantiated templates to describe details, relations and constraints found inside the policies. A simple Policy Model example is shown in Figure 4. It shows a simplified state policy template that is instantiated to describe a policy saying: "*Mental Health Record* of a *Mental Health Patient* (who is classified as a *patient* from an Ontology Model) can only be disclosed to a *Psychiatrist* (who is classified as a *doctor* from an Ontology Model) if there is an established *Treatment Relation* between the *Mental Health Patient* and the *Psychiatrist*."

4) Policy Verification and Analysis: The Policy Verification and Analysis tool enables users to check the consistency of designated policy sets and analyze the impact of policies on established scenarios.

Policy Analysis enables users to test a composition of policies for contradictions and entailment that could cause problems if the policies are be put into effect. The analysis can be run on policy model sets. These model sets are artifacts built hierarchically from policy models and policy model sets. This hierarchical organization enables the simple extension and

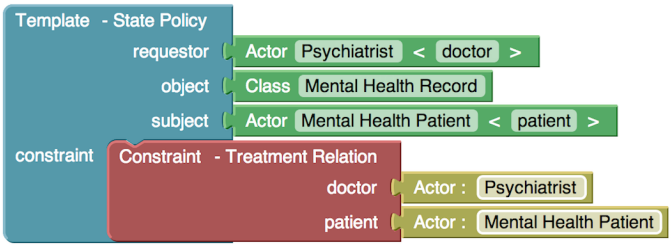

Fig. 4. A simple example of a Policy Model in PolicyForge.

specialization of policy sets in different projects using shared artifacts. The analysis is done by instantiating the analysis semantics of all the policy models in the policy model set using their policy templates. The instantiation process generates an executable verification code that can be run on a compatible execution environment, such as the widely adopted Prolog or the novel Formula [17], residing in the PolicyForge cloud.

Policy Verification makes it possible to verify consistent policy sets against established scenarios of health care activities such as disclosures, information exchanges or data access. The scenarios are relatively simple data flow models specialized to the domain. The data in this model is the health information of the patient as well as other administrative data elements such as visit history, appointment schedules and consent documents. The actors in the data flow model are the patient, the care provider and other entities that are involved in the communication or want to have access to the patients data. The actors and the data are derived from the ontology models similarly to the policy models. The verification engine first matches the actors and documents to the similar parts of the policies in the policy set using the shared ontologies. After the scenario and policies are composed using the ontologies, the logic expressions from the semantic anchors of the policy templates are instantiated. The instantiated logic expressions are then executed and solved by constraint satisfaction algorithms in the same execution environments as used for the policy analysis. The result of the verification can be twofold. First, it can tell whether the scenario is valid or not in view of the policy set. Second, if the execution environment has the capability it is also possible to infer where the contradiction between the policies and the scenario lies and possibly offer a solution to incorporate into the scenario to mend the contradiction.

5) Policy Export and Integration: After the Policy Models are analyzed, verified and tested, they can be exported from PolicyForge into an HIS to regulate or audit the workflow and the execution. Similarly to analysis and verification the semantics of the execution has to be anchored to the Policy Model Templates from which Policy Models are composed. Together with the Policy Models the Ontology Models can also be exported and used as configuration parameters to an HIE or HIS.

C. Collaboration Tools

1) Projects and Neighborhoods: The organization of the fundamental forge concepts in PolicyForge is derived from the

Allura core. Projects embody the collaboration spaces where members of a team of users can collaborate. There are tools available in the Project space for the collaborative design work. Registered users can create new Projects or acquire membership in an existing one. Projects are created based on pre-configured templates but in general, each team controls how it utilizes the Project for its work. PolicyForge implements role-based access control. Privileged administrator users of each project can freely provision new tools and administer the tools and members of the Project.

PolicyForge supports the concept of Neighborhoods. Neighborhoods are collections of projects, usually representing institutions or domains with which the teams of the member Projects are affiliated in the physical world. Neighborhoods also offer similar collaboration functionalities to the Project spaces: they can have members, customized roles and selected tools installed for Neighborhood-level collaborative work.

2) User Management: The framework that the PolicyForge is being built upon enables very flexible user management. Each project in the PolicyForge can create Permissions and User Groups to match its requirements. The tools enabled in the project can use these groups to determine the permissions of each user in the project.

3) Message Boards and Ticketing: PolicyForge also comes with standard forums and message boards as well as an issue and ticket tracking service. These collaboration tools are associated to projects and can be read and written by members of the project, unless the projects administrators open these up to other projects and members.

D. Policy Exchange

The artifacts in the Policy Forge are only accessible to the users associated with the project where the artifact was created unless the artifact is explicitly shared. This feature enables very flexible control over the access of the artifact by users outside the working group that created the artifact. The Policy Exchange tool enables browsing and discovery of the artifacts that were made available by the projects in the PolicyForge.

All the artifacts in the PolicyForge are organized into a taxonomy (Figure 2) to enable efficient but flexible discovery of the artifacts. Each type artifact in the taxonomy may have unique properties over the general properties it inherits from their parent type. The properties can also be used to filter the search results using intelligent filters.

III. Conclusion

PolicyForge.org is envisioned to be an open, community-driven platform, offering an authoritative reference source for policy makers and users in an online space. It enables both private and open collaboration among teams and individual users at many levels (institutional, network, state, federal) for viewing, reviewing, discussing, developing, interpreting, comparing, and tracking privacy policies. It integrates policy tools developed by different communities in a consistent, widely accessible framework, and brings crowdsourcing capabilities to the process of authoring, interpreting, analyzing and implementing privacy policies in health care. This complex process require participation from groups of individuals offering heterogeneous knowledge, input from multiple stakeholder institutions, and extensive collaboration and consensus building at local, state and national scales.

Acknowledgments

The work presented in this paper was funded through National Science Foundation (NSF) TRUST (The Team for Research in Ubiquitous Secure Technology) Science and Technology Center Grant Number CCF-0424422 and Office of National Coordinator for Health Information Technology (ONC) Grant Number HHS 90TR0003/1. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the HHS or NSF.

References

[1] U. S. Congress, "HIPAA: Health insurance portability and accountability act," 1996.

[2] U. S. Congress, "Health information technology for economic and clinical health (HITECH) act," February 2009.

[3] J. Pritts, S. Lewis, R. Jacobson, K. Lucia, and K. Kayne, "Report on state law requirements for patient permission to disclose health information," *RTI International report*, aug 2009.

[4] M. E. Frisse, K. B. Johnson, H. Nian, C. L. Davison, C. S. Gadd, K. M. Unertl, P. A. Turri, and Q. Chen, "The financial impact of health information exchange on emergency department care," *Journal of the American Medical Informatics Association*, 2011.

[5] H. Nissenbaum, "Privacy as contextual integrity," *Washington Law Review*, no. 79, pp. 119–158, 2004.

[6] A. Barth, J. Mitchell, A. Datta, and S. Sundaram, "Privacy and utility in business processes," in *Computer Security Foundations Symposium, 2007. CSF '07. 20th IEEE*, july 2007, pp. 279 –294.

[7] A. Datta, J. Franklin, D. Garg, and D. Kaynar, "A logic of secure systems and its application to trusted computing," in *Security and Privacy, 2009 30th IEEE Symposium on*, may 2009, pp. 221 –236.

[8] P. E. Lam, J. C. Mitchell, and S. Sundaram, "A formalization of hipaa for a medical messaging system," in *Proceedings of the 6th International Conference on Trust, Privacy and Security in Digital Business*, ser. TrustBus '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 73–85.

[9] M. Y. Becker, C. Fournet, and A. D. Gordon, "SecPAL: Design and semantics of a decentralized authorization language," *Journal of Computer Security*, vol. 18, no. 4, pp. 619–665, 2010.

[10] R. Craven, J. Lobo, E. Lupu, J. Ma, A. Russo, M. Sloman, and A. Bandara, "A formal framework for policy analysis," *Imperial College London, Tech. Rep*, 2008.

[11] G. Simko and J. Sztipanovits, "Active monitoring using real-time metric linear temporal logic specifications." in *HEALTHINF*, 2012, pp. 370–373.

[12] A. Nadas, M. E. Frisse, and J. Sztipanovits, "Modeling privacy aware health information exchange systems," in *1st International Workshop on Engineering EHR Solutions (IWEES)*. Amsterdam Privacy Conference 2012, 2012.

[13] K. Waterman, "Preprocessing legal text: Policy parsing and isomorphic intermediate representation," 2010. [Online]. Available: http://www.aaai.org/ocs/index.php/SSS/SSS10/paper/view/1112

[14] *VechicleForge platform*, Institute for Software Integrated Systems, Vanderbilt University, http://vehicleforge.org/faq.

[15] J. H. I. H. D. L. M. P. F. P.-S. L. A. S. Sean Bechhofer, Frank van Harmelen, *OWL Web Ontology Language Reference*, W3C, http://www.w3.org/TR/owl-ref/.

[16] A. Nadas, T. Levendovszky, E. K. Jackson, and J. Sztipanovits, "A model-integrated authoring environment for privacy policies," *Science of Computer Programming*, vol. Special Issue on Success Stories in Model Driven Engineering, 2012, submitted.

[17] E. K. Jackson and W. Schulte, *FORMULA (Formal Modeling Using Logic Programming and Analysis)*, Microsoft Research, http://research.microsoft.com/en-us/projects/formula/.