Towards a Semantic Web Testbed for Collaborative Policy Development

Li Ding[1], K Krasnow Waterman[2], Jie Bao[1], Lalana Kagal[2] and Deborah L. McGuinness[1]

[1]Tetherless World Constellation, Rensselaer Polytechnic Institute, 110 8th St, Troy, NY12180, USA

[2] Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139, USA

dingl@cs.rpi.edu, kkw@mit.edu, baojie@cs.rpi.edu, lkagal@csail.mit.edu, dlm@cs.rpi.edu,

Abstract

We aim to improve accountability of distributed data usage by creating tools to support collaborative online development of scenarios and policy infrastructure. This paper reports on our design and development of a simple adaptive semantic wiki-based testbed and describes its usage in privacy protection scenarios.

1. Introduction

The Transparent Accountable Datamining Initiative (TAMI) [1] project is aimed at building a policy infrastructure that (i) allows users to run compliance checks of data usage policies on the transaction logs of Web-based data flow among distributed data custodians, and (ii) generates user friendly justifications for the results of compliance checks. Our work on the policy infrastructure includes the AIR language [2], the AIR policy reasoner, and the justification user interface. The development of the policy infrastructure is driven by a series of increasingly complicated scenarios that embody real world privacy protection requirements. Developing these scenarios requires a team of physically distributed researchers with diverse skill sets; therefore, a supporting testbed is needed for the collaborative policy infrastructure development for example, to construct hypothetical scenarios (content and ontology) with variations.

In this paper, we describe our recent work on a Semantic Mediawiki (SMW) [3] based testbed for policy infrastructure development. The testbed lets us collaboratively construct, evolve, browse, and review scenario data in a manner suitable for those who are accustomed to Web 2.0 applications. We describe its use in our TAMI project and show how it can be evolved to adapt to the new challenges introduced by the latest TAMI scenarios. Live demos can be found at http://tw.rpi.edu/proj/tami/.

During the development of our testbed, we made the following observations: (i) increasingly complicated knowledge can and should be organized into a collection of semantically inter-linked web pages using semantic wiki; (ii) each page should integrate both text (which is suitable for capturing deep semantics) and semantic annotations (which is suitable for capturing frequent shallow semantics); (iii) changes of semantic annotation should be automatically reflected in its text version (for human consumption) and semantic version (for machine consumption) to reduce the cost of synchronizing the two versions; and (iv) new challenges introduced by complex scenarios, such as ontology evolution and hypothetical testing, can be accommodated by adding some small extensions to Semantic Wiki.

This paper is structured as follows: section two reviews the background of TAMI scenarios and the requirements to our policy testbed; section three shows the design highlights of our SMW-based policy testbed; section four discusses new challenges introduced by TAMI scenario 11 and how our testbed can be evolved to meet the challenges; section five discusses related work; and section six concludes our work with future plans.

2. TAMI Background and Requirements

2.1 TAMI Project and AIR Language

The TAMI project focuses on creating technical, legal, and policy foundations for transparency and accountability in heterogeneous distributed information systems, e.g., the federal/state information exchange systems for law enforcement and national security. Large-scale, decentralized systems like the Web provide ease of data flow; this revolution, however, is threatened by the challenges of inappropriate use of information. Although access control mechanisms have been adopted by information providers to prevent unauthorized data access, they are simply not enough to address the privacy risks in a world where information is ever more easily copied and improperly passed on even by authorized users, and where automated correlations and inferences across multiple databases can uncover information even when it has not been explicitly revealed. Therefore, the TAMI project investigates privacy protection solutions by: (i) designing expressive and precise rule languages that are able to express policy constraints; (ii) building a policy reasoner to check whether the transaction logs generated by the distributed and heterogeneous information systems are compliant with the applicable laws and polices, (iii) providing methods and technologies to enhance the accessibility of the policy compliance results to end users; and (iv) developing scenarios to embody challenges to the policy infrastructure. The results can support real-world systems that want to increase the transparency of data handling and accountability for meeting policy requirements. More details about the TAMI project can be found in [1] and http://dig.csail.mit.edu/TAMI/.

The AIR language is a general-purpose rule language, grounded in Semantic Web technologies, aimed at enabling policy compliance checks in open, decentralized information infrastructures. The AIR policy reasoner is a production rule system and uses a RETE algorithm for pattern matching. Along with policy compliance checking, the AIR policy reasoner also produces a justification for the compliance check results. It uses a Truth Maintenance System (TMS) in order to maintain the set of premises for every conclusion in the knowledge base. These premises are the set of facts and rules from which the conclusion is inferred. The justification is produced by retrieving relevant information from these results in the form of a proof tree. The justification user interface is part of the Tabulator [5], a Semantic Web browser; and it allows users to view the explanation provided by the AIR policy reasoner in different ways: (i) in a simple rule language, N3, and (ii) in a graphical layout that highlights the results of the inference and allows the explanation to be explored. More details about the AIR language, the AIR policy reasoner and the justification UI can be found in [2].

2.2 TAMI Scenario Development Process

Current TAMI development is running in a scenario-driven mode. A collection of fictional scenarios have been developed in the past few years to drive TAMI policy infrastructure development.

- Scenario 0: The records about a student's use of a door access card cannot be used for a disciplinary proceeding because the school's rule only permits use for a criminal investigative proceeding. This scenario offered a simple transaction and a single policy for testing the policy reasoner.
- Scenario 3: A flight passenger being screened by the Transportation Security Administration might be a match with a possible terrorist; further investigation by the US Marshals reveals he is the subject of an outstanding warrant for unpaid child support. Here it must be determined if the information can be passed to the US Marshals. The scenario required representation of more transactions and more complex policies. It caused the development of an algebraic method of representing and comparing purpose of use against original purpose for collection; it also drove the development of truth maintenance technology so that the system can express the limits of its ability to reason.
- ARL Scenario: A military unit uses Red Cross information, in violation of Universal Declaration of Human Rights and UN Resolution, to locate a person of interest. The scenario provided the impetus for the initial semantic user interface, providing multiple layers of information from the AIR code representation of the policy and RDF representation of the transaction to a user-facing expression of the reasoning.
- Scenario 9: Because a TB patient is in a coma, the CDC has to conduct a digital investigation to determine the patient's contacts/potential spread of the disease. A phone company later denies a service visit to a person included in the CDC's investigatory net. This scenario creates the challenge of identifying an inferred fact. It also provided a driver for an enhanced user interface with an additional layer that provides near-grammatical explanation of the reasoner's conclusion and reasons therefore.
- Scenario 11: Local law enforcement receives suspicious information from a private enterprise, merges it with state information and provides it to a state fusion center. A federal agency trying to decide whether to share related information needs to understand how the state's disclosure laws for merged information differ from its own. The scenario introduced the opportunity to aggregate transaction log data from distributed locations and the need for temporal reasoning.

The process for developing a TAMI scenario typically involves the following steps:

- Identify some desired policy language features, and outline a reality-based scenario that will require the identified features
- Compile a detailed storyline for the scenario including the text versions of relevant policies, synthetic transaction logs, and specific policy checking tasks with expected results.
- Generate Semantic Web versions of the scenario data as the input to the AIR policy reasoner by (i) encoding the transaction log using RDF, (ii) evolving the transaction log ontology; and (iii) expressing the identified policies in AIR.
- Evaluate the experimental results by comparison with expectations. Tools for browsing the interlinked transaction log and policies are needed to debug the results.

2.3 Requirements for the TAMI Policy Testbed

In order to support the TAMI policy infrastructure development, our testbed focuses on hosting TAMI *scenario data* including (i) the transaction log ontology; (ii) the transaction log data that details the story covered by the scenario, and (iii) the policies to be tested in the scenario. The testbed should be designed as an online environment with the following features:

- Collaboration support for distributed team members to revise scenario data
- Integrated editing and publishing facilities for users to integrate deep semantics and shallow semantics at the same time, and synchronize the changes to the text version and Semantic Web version of the scenario data
- Smart data access mechanisms for users to effectively browse, locate and debug their information

When new TAMI scenarios introduce new challenges, for example, to evolve the transaction log ontology and to export the semantic web version of variations of a transaction log, the testbed should also be able to evolve to adapt to the challenges.

3. TAMI Policy Testbed

Although the process of developing TAMI scenarios can be approached by conventional text and Web authoring tools, our SMW-based testbed has proven more effective. In what follows, we show some highlights from building our light-weight testbed for TAMI policy infrastructure development.

3.1 Wiki-based Collaborative Environment

Wikis are well-known platforms for users to collaboratively organize knowledge on the Web. As a popular wiki platform, MediaWiki offers both collaborative editing functions and many data access functions such as search, list recent changes, list in-links to a page, and list revisions of a page. It is used to power wikipedia.org, the famous collaborative online encyclopedia.

Semantic MediaWiki is an extension of MediaWiki and provides many enhanced knowledge management features:

- It allows hybrid wiki pages with both text and semantic annotations. In addition to page categorization, users can annotate semantic relations between wiki pages
- It renders semantic annotations in both text and RDF/XML, which is a standard syntax of Semantic Web languages.
- It supports semantic query: users can query and display semantic content from other pages on any wiki page. Semantic query is more flexible than page transclusion[1].

While MW and SMW natively enable a collaborative online environment, the above features are heavily used to build our testbed as described in the rest of this section.

3.2 Integrated Data and Policy Editing

The data of each TAMI scenario typically mixes text and semantic annotations; moreover, we are required to maintain both the text version (for discussion) and the semantic annotation in the testbed. Although we could maintain them in different files, it became increasingly costly to synchronize versions of the same data especially when the scenario data is incrementally revised. Benefiting from SMW's hybrid content support, our testbed integrates text and semantic annotations in one wiki page, and

[1] Transclusion refers to the inclusion of part of a document into another document by reference. See http://en.wikipedia.org/wiki/Transclusion

automatically synchronizes the text version and semantic web version of the semantic annotations.

Figure 1 shows an example wiki page encoding both text and semantic annotations about an event in a transaction log. "[[Mr. Parker]]" generates a wiki link; "[[Category:Event]]" not only generates a wiki category but also generates a semantic annotation indicating "the entity described by this page is an instance of event"; "[[coordinator::William Parker]]" is encoded in SMW syntax to generate a semantic annotation indicating "the entity described by this page has a coordinator called William Parker".

```
[[Mr. Parker]] observed a glint of light in the window of a parked station wagon across the street, and then identified a male in the driver's seat apparently taking photographs of the gate and/or guard shack.
* [[coordinator::William Parker]]
* [[datetime::2008/10/26 06:44:00 AM]]
[[Category:Event]]
…
```

Figure 1 Example wiki page for an event in transaction log

Figure 2 shows a wiki encoding identifying a Florida state law. Besides the text and semantic annotations, our testbed additionally allows users to include the policy content in N3 format enclosed by "__BEGIN_ANDMAR_N3__" and "__END_ANDMAR_N3__".

```
…
[[Category:Policy]]
__BEGIN_LANDMARK_N3__ <pre>
@prefix air: <http://dig.csail.mit.edu/TAMI/2007/amord/air#> .
@prefix tami: <http://tw.rpi.edu/proj/tami/Special:URIResolver/> .
....
:FS_119_01_1  a  air:Policy;
  rdfs:label "Fla. Stat. Ch. 119.01(1)";
  ....
</pre>__END_LANDMARK_N3__
```

Figure 2 Example wiki page for an AIR policy

3.3 Ontology-based Template and Forms

An interoperable transaction log ontology is critical to effective policy checking. Our work developed a simple upper ontology for an interoperable transaction log (see Figure 3).

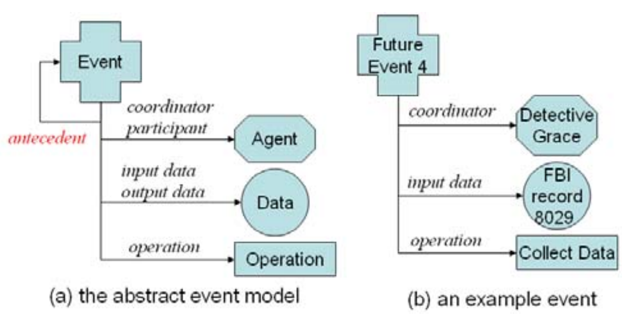


Figure 3 A simple event model and example

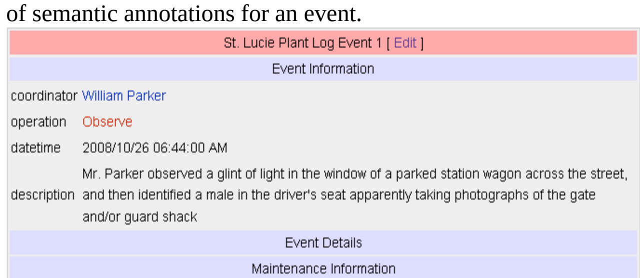
- Event: the audit logs of an event where an operation has been executed by someone over some data. E.g. "FBI record 8029 was collected by Detective Grace." Events are partially ordered by the "antecedent" relation.
- Agent: the actionable entities (people, organizations, systems) involved in an event. Each has a name and some optional properties such as homepage, location, and purpose.
- Data: the data records involved in an event. Each instance constitutes a reference to a definitive piece of data, and can be annotated by category (e.g. healthcare, commercial-use, criminal investigation).
- Operation: the actions taken in an event.

We use the "semantic template" to support and enforce the use of the transaction log ontology. It recognizes the minimal commonalities of the annotated events among distributed transaction logs. Users can still keep or add their own semantic annotations outside the semantic templates. Figure 4 illustrates an "Event" template that substitutes the semantic annotations shown in Figure 1, and one more customized semantic annotation is added to indicate the gender of the suspect.

```
{{Tami.event
|coordinator=William Parker
|datetime=2008/10/26 06:44:00 AM
|description=[[Mr. Parker]] observed a glint of light in the window of a parked station wagon across the street, and then identified a male in the driver's seat apparently taking photographs of the gate and/or guard shack    }}
[[suspect gender::male]]
…
```

Figure 4 Example wiki page using semantic template

Besides generating semantic annotations, a semantic template also synchronizes the text version and semantic version of semantic annotations. Figure 5 shows the table-based rendering of semantic annotations for an event.

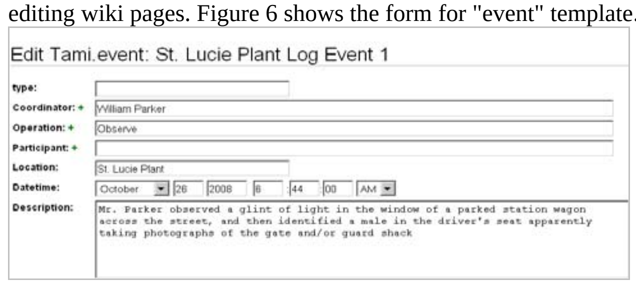


St. Lucie Plant Log Event 1 [Edit]	
Event Information	
coordinator	William Parker
operation	Observe
datetime	2008/10/26 06:44:00 AM
description	Mr. Parker observed a glint of light in the window of a parked station wagon across the street, and then identified a male in the driver's seat apparently taking photographs of the gate and/or guard shack
Event Details	
Maintenance Information	

Figure 5 Example rendering of semantic template

Moreover, we can derive new semantic annotations using the combination of semantic templates and semantic queries, e.g. inferring "antecedent" relations by querying pages that have asserted "antecedent of" relations to the current page.

We also use the SemanticForms extension to provide users with a form based editing interface to further reduce the cost of editing wiki pages. Figure 6 shows the form for "event" template.



Edit Tami.event: St. Lucie Plant Log Event 1

type:	
Coordinator: *	William Parker
Operation: *	Observe
Participant: *	
Location:	St. Lucie Plant
Datetime:	October 26 2008 6 44 00 AM
Description:	Mr. Parker observed a glint of light in the window of a parked station wagon across the street, and then identified a male in the driver's seat apparently taking photographs of the gate and/or guard shack

Figure 6 Example form for editing "event" template

3.4 Semantic Linking and Browsing

As the size and complexity of the transaction log grows, effective data access is needed in our testbed. In complementing the native text data access features offered by MediaWiki, we investigate the benefits of semantic annotation based data access.

Since we have added semantic annotations into wiki pages using SMW syntax, semantic template and simple inference, the asserted semantic annotations can be browsed bi-directionally via SMW's browse interface (see Figure 7).

Figure 7 Example SMW browsing UI for an event's wiki page

We also used the "semantic query" to help users dynamically link to other pages. Figure 8 shows a table generated by semantic query listing the details (with many links) of events in the transaction log owned by "St. Lucie Plant". Here, a semantic query functions like a simple SQL query. Similarly, we can list events related to a policy by querying the policy's pattern, e.g. "list events whose coordinator is a US government agency".


List of Events

	Datetime	Coordinator	Operation	Participant	Input data	Output data	Antecedent
St. Lucie Plant Log Event 1	26 October 2008 06:44	William Parker	Observe				St. Lucie Plant Log Event 1
St. Lucie Plant Log Event 2	26 October 2008 06:50	William Parker	Compile Data		FPL Security Policy 3-1	IMAGE 293-1 IMAGE 293-2 St. Lucie Plant Incident Report 293	St. Lucie Plant Log Event 1 St. Lucie Plant Log Event 2
St. Lucie Plant Log Event 3	26 October 2008 06:52	William Parker	Disclose Data	Port St. Lucie Police Department			St. Lucie Plant Log Event 2 St. Lucie Plant Log Event 3

Figure 8 Example linked table generated by semantic query

It is notable that the RDF dump generated by SMW is intensively inter-linked. Therefore, external tools, such as Tabulator, can browse the transaction log from page to page. Figure 9 shows the Tabulator view of the semantic web version of the same event displayed in Figure 7.

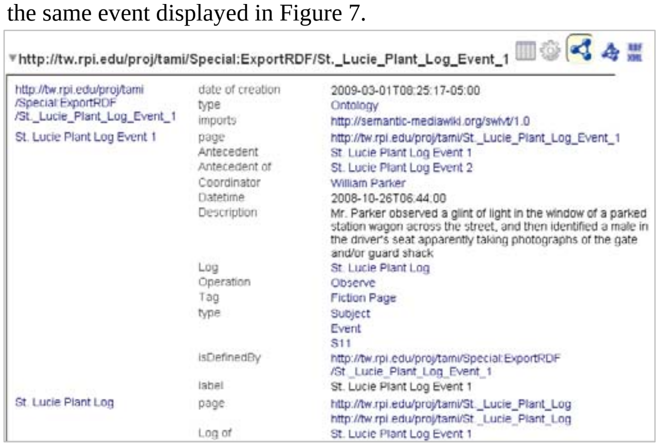


Figure 9 A policy page rendered by Tabulator Firefox extension

3.5 Enhanced Semantic Publishing

In order to prepare the input for the AIR policy reasoner, the policy testbed needs to provide (i) a single N3 dump of the entire transaction log, and (ii) one or more N3 dumps of the related AIR policies. Unfortunately, the current distribution of SMW does not support (i) exporting a single N3 dump of the semantic annotations from an arbitrary selection of wiki pages (e.g. pages belong to one transaction log); and (ii) exporting AIR policies encoded in N3.

Since these two features are critical for connecting the testbed with the policy reasoner, we enhanced SMW with the two new functions: the first one was implemented by minimally revising SMW source code; and the second one was implemented by adding a new SMW extension called SimpleExport (Figure 2 shows the landmark for identifying the N3 data to be exported).

4. Discussion

Beyond the functions described in the previous section, we show some examples where our testbed adapted to challenges from new scenarios.

4.1 New Challenges from TAMI Scenario 11

The latest TAMI Scenario 11 covers the following: two law enforcement agencies need to understand each other's data release policies so that they can decide what data they're willing to share with each other. This scenario introduces two new challenges:

- The status of data may change over time; therefore, accessing the same data may be allowed or disallowed by the same policy depending on when the data access was done.
- Different agencies (federal, state, city) may draw different conclusions from the same event because they are regulated by different rules, different access to the event data, and etc.

4.2 Ontology Evolution

In order to meet the first challenge, we added (i) two classes - "Snapshot" (e.g. content of a file at a certain time) and "Data Record" (e.g. a file) as sub-classes of "Data"; and (ii) let the properties "input data" and "output data" link to instances of "Snapshot", which then links to instances of "Data Record". This approach helps us to maintain the transaction log in a monotonic growth mode, i.e., the log grows without deleting or updating existing knowledge. Such an ontology can be used to address "temporal" compliance checks, for example, evaluating a policy which depends upon whether criminal intelligence information (CII) was "active". Before the change of the ontology, we could not simply determine whether the "status" property of an instance of "Data" was "active" because the value of the property could change over time. After the change, we can determine that by comparing the timestamps of the instances of "Snapshot" and "Event": events accessing CII data between the opening event and closing event of a criminal investigation case are accessing "active" CII data.

The corresponding updates in our testbed are fairly small: (i) to add two wiki pages for the two classes respectively; (ii) to add a new form for "Snapshot" (copied from "data" template/form and added two more form fields, i.e., "timestamp" and "data-record"), and (iii) change the auto-completion category of the "input data" and "output data" fields of the "Event" form to "Snapshot".

4.3 Hypothetical Testing

In order to adapt the testbed to meet the second challenge, users need more freedom in constructing hypothetical testing via (i) variations of transaction logs for different context, and (ii) incrementally updating scenario data with a restore option.

Hypothetical testing can be done by checking policies against several different variations of a transaction log. The variations share many common events in the log, and each variation is dumped into one RDF/XML file as the input of the policy reasoner. While we can always duplicate the shared events in the variations, it becomes increasingly difficult to maintain the variations when we update the shared events. The versatile RDF export capability mentioned in section 3.5 makes it easy for a user to produce any number of variations of a transaction log by selecting fine-grained elements of transaction log data and ontology. For example, one variation may include the ontological axiom that "antecedent is a transitive property", but the other may not. Similarly, one may create a variation by excluding an event that "a criminal investigation is closed by the local police" to see the impact of missing knowledge in policy checking.

Hypothetical testing can be done at even finer-grained level than a wiki page, i.e. users can try variations of semantic annotations within the same wiki page. The test process works as follows: (i) a user selects a wiki page and changes the semantic annotations and/or semantic content; (ii) the changes immediately affect the corresponding RDF or N3 dump, (iii) the user then uses the policy reasoner to detect the changes of policy checking results; (iv) the user browses changes of the updated wiki page using the MW native "diff" computation over the corresponding revision history, and then associates such changes with the changes of policy checking results; and (v) the user chooses to either keep the current version or to restore the previous revision of the wiki page after the test. Figure 10 shows the changes between two versions of a policy "FS 119 01 1-body", and the changes can be used to explain why the older version failed after the upgrade of AIR policy reasoner. We are also working on semantically encoding the revision history using a simplified version of the Proof Markup Language [6] that will help us automate the above process.

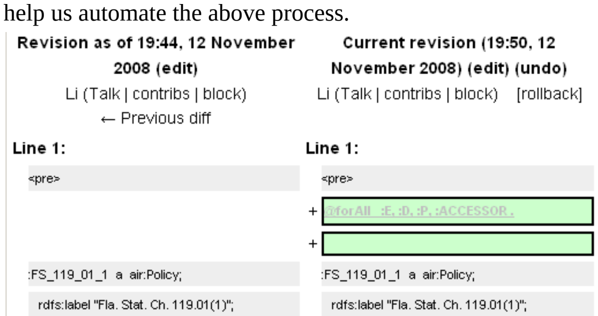


Figure 10 example changes of wiki page

5. Related Work

Research on policy management investigates authoring, storing and querying policies, especially privacy policy. Karat et al. evaluated the effectiveness of both natural language based and form-based policy authoring interfaces [7]. Collaborative policy management has also been investigated [8, 9]. Feeney et al. [8] proposed a community policy system where users can use a local graphical UI or an online form-based UI to access policy management web services. Zeiss et al. [9] proposed a Semantic Web based policy management environment with an online form-based authoring interface for N3 policies. Although our testbed has not yet provided a policy editor like the above work, it supports users who need to (i) record and query the metadata (such as provenance and classification) of policies, (ii) track versions of policies by semantic history, and (iii) embed both the original text encoding and the AIR encoding of a policy. Moreover, our tested uniquely lets users edit test cases for validating user contributed policies.

Semantic Wiki has been used for knowledge management in various domains such as personal information, mathematical formulae, workflow data, and ontologies [10]. Our testbed has been used to investigate the new domains of transaction logs and policy. It includes several wiki extensions and contributes to the best practices for semantic wiki-based applications.

6. Conclusion

We built a semantic wiki based testbed to support the development of our policy infrastructure. The testbed has been used to manage hypothetical scenario data, including policies and transaction logs, for testing the expressiveness of policy language, the functionality of a policy reasoner, and the supporting justification user interface. The testbed has allowed us to quickly prototype a knowledge management system and supported the required evolution as the complexity of scenarios increased. In future work, we will evolve this testbed into a policy management system with special focus on a policy authoring interface, provenance of policy and version-related computations.

Acknowledgments

This work is partially supported by NSF Cybertrust Award #0524481, IARPA Award #FA8750-07-2-0031, and DARPA award #FA8750-07-D-0185, #55-002001, #FA8650-06-C-7605. We would like to thank our colleagues Hal Abelson, Tim Berners-Lee, James Hendler, Ian Jacobi, Ankesh Khandelwal, Yosi Scharf, Oshani Seneviratne, Gerry Sussman, and Daniel Weitzner for their participation in this project.

REFERENCES

[1] Daniel Weitzner, Hal Abelson, Tim Berners-Lee, Christ Hanson, Jim Hendler, Lalana Kagal, Deborah L. McGuinness, Gerry Sussman, K Krasnow Waterman, *Transparent Accountable Inferencing for Privacy Risk Management*. Proc. of AAAI Spring Symposium on The Semantic Web meets eGovernment, 2006.

[2] Lalana Kagal, Chris Hanson, Daniel Weitzner, *Integrated Policy Explanations via Dependency Tracking*, IEEE Policy 2008.

[3] Markus Krötzsch, Denny Vrandecic, Max Völkel, *Semantic MediaWiki*, ISWC'06, 2006

[4] Jie Bao, Li Ding, Deborah L. McGuinness, James Hendler, *Towards Social Webtops Using Semantic Wiki*, ISWC'08 poster & demo, 2008

[5] Tim Berners-Lee and Y. Chen and L. Chilton and D. Connolly and R. Dhanaraj and J. Hollenbach and A. Lerer and D. Sheets. *Tabulator: Exploring and Analyzing linked data on the Semantic Web*. Proc. SWUI06 Workshop, 2006.

[6] Deborah L. McGuinness, Li Ding, Paulo Pinheiro da Silva, Cynthia Chang. *PML 2: A Modular Explanation Interlingua*, Proc. of the Workshop on Explanation-aware Computing (ExaCt-2007), 2007

[7] C.-M. Karat , J. Karat, C. Brodie, J. Feng. *Evaluating Interfaces for Privacy Policy Rule Authoring*, CHI. 2006

[8] K. Feeney, C. Tsarouchis, D. Lewis, *Policies as Signals in Collaborative Policy Engineering*, Proc. of 1st International Workshop on Policy-Based Autonomic Computing, 2007

[9] J. Zeiss, R. Gabner, A.V. Zhdanova, S. Bessler. *A Semantic Policy Management Environment for End-Users*. Proc. of International Conference on Semantic Systems, 2008.

[10] Christoph Lange, Sebastian Schaffert, Hala Skaf-Molli, Max Völkel (Eds.). *Proc. of 3rd Semantic Wiki Workshop*, 2008