Prototyping Fusion Center Information Sharing

K. Krasnow Waterman, Samuel Wang and William D. Cattey
Decentralized Information Group, CSAIL
Massachusetts Institute of Technology
Cambridge, MA, USA
Email: {kkw, samuelsw, wdc}@mit.edu

Abstract—In 2004, the White House and then Congress determined there should be an “Information Sharing Environment” that facilitates the flow of critical information for counter-terrorism, related law enforcement, and disaster management activities. That work has been progressing but major challenges include how to ensure compliance with laws and policies of the federal government, 50 states, and individual agencies; how to pass appropriate data that would support access control and privilege decisions in different jurisdictions; and how to achieve accountability and transparency for this activity. We have built a prototype of Fusion Center information sharing that shows significant progress in the representation of law in a policy language, the reasoning of that law over data transactions occurring in a web environment (internet or intranet), acquiring necessary information from authoritative sources wherever they reside in the decentralized environment, and providing both a binary response suitable for workflow implementation and a detailed justification suitable for validating the conclusion. In this paper, we briefly describe the technologies employed for serializing the data and policy, reasoning over the rules contained in the policy, and displaying the results to users. These combine to provide a powerful tool to support a range of necessary governmental functions including access control, privilege management, audit, periodic reporting, or risk modeling.

I. INTRODUCTION

After 9/11, within the US, a cry arose that the terrorist attack could have been averted if government agencies had shared what they knew with each other. While the accuracy of that claim remains in debate, there is significant evidence that agencies were sharing less than expected and that they would operate more effectively if they did share information. Three years later, having not made significant progress towards that goal, the White House issued an Executive Order [1] mandating the creation of an Information Sharing Environment; this goal was reinforced by Congress later the same year when it was mandated in a new statute [2].

In the years since the goal was set, an impediment to implementation has been identified. The sharing is mandated to be performed “[t]o the maximum extent consistent with applicable law.” However, a gap exists between the law and policies enacted by government to regulate the handling of information and the ability to enforce those policies in systems. There is a strong desire to bridge that gap as more data is or is desired to be collected, shared, and manipulated. Responsible managers and interested citizens alike are seeking the means to ensure that systems more effectively implement rules about privacy, security, and the appropriate conduct of government business. But, while people can express rules with complex reasoning, context, and reference to information not contained in the subject data, systems historically have not handled such challenges.

For example, consider the following snippet of legislation enacted by the state of Maryland:

A. Subject to the provisions of Regulation .12B, the Central Repository and other criminal justice agencies shall disseminate CHRI, be it conviction or nonconviction criminal history record information, to a criminal justice agency upon a request made in accordance with applicable regulations adopted by the Secretary. A criminal justice agency may request this information from the Central Repository or another criminal justice agency only if it has a need for the information:

(1) In the performance of its function as a criminal justice agency; or

(2) For the purpose of hiring or retaining its own employees and agents.

It is clear that the intent of this legislation is to regulate the transmission of sensitive criminal history record information so that it is only used for appropriate purposes. However, the interactions between this specific policy and other policies at the organization, state, and federal level could potentially be very complex, and it is not feasible for humans to be able to reason over all of them simultaneously. In addition, the rules and terms used in policies often reference other policies and pieces of information located in different databases or organizations, which makes it difficult to efficiently verify compliance by hand. Finally, given the number of transactions that happen per day, if a violation does occur, it is difficult to verify exactly which information sharing transaction was non-compliant with the applicable policies.

Given that computers are already ubiquitous in data sharing
environments due to the ease of sharing and aggregating information, it is worthwhile to investigate whether or not they can also solve the problems listed above. We built a prototype of an “accountable system” to address this challenge by using Semantic Web technology. An accountable system is one which can reason over complex policy over the details of data transactions, so that organizations can be accountable for their compliance and be transparent about it. In this project, we modeled transactions between Fusion Centers – locations where state and federal agencies work cooperatively to address terrorism, crime, and emergency response. Our prototype associates RDF tags with meaning about data, actors, and context, expresses policies in AIR, and reasons over them with a forward-chaining RETE engine, cwm. It uses dependency tracking in a Truth Maintenance System to reveal the bases for it’s conclusions, and it displays the component parts and results (justifications) using multiple pane views of Tabulator, a Semantic Web browser. Our prototype shows that the authoritative sources of information needed to make policy-based decisions can remain and be accessed wherever they reside in the decentralized environment (internet or intranet), making it a quintessential example of policy-controlled Linked Data.

This paper presents a prototype system that models the data sharing workflow in a Fusion Center environment, with the following features:

1) An effective way to represent real legislation and policies in a computer-readable language that can be reasoned over.
2) Leverages semantic web technologies so that data can be placed in disparate databases and servers which the reasoner can access on-the-fly during reasoning.
3) A reasoner that can analyze transactions with rules, and then present a justification of why the transaction is or is not compliant.
4) A user interface that analysts can use to verify that their transactions are compliant with the applicable policies. The user interface is designed for end-users that have neither a legal nor a technical background, with justifications that are presented in natural language to the user.

The prototype demonstrates that such a reasoning system can be used to increase the amount of transparency and accountability in real data-sharing environments. Given any data sharing event, the reasoner can produce a transcript that shows exactly which pieces of data went into the reasoner, and which parts of the law justifies the transaction as legitimate.

II. BACKGROUND
A. History of Information Sharing Environment
B. Goals of Accountable Systems

Accountable systems are an alternate way to consider privacy and security in computer systems. Almost all existing systems consider data security to be the problem of safeguarding private information within certain predefined boundaries. However, it is unrealistic to create a world where all of your private data is meticulously stored and safeguarded by all of the organizations that need that information. In reality, there will always be incidents where security boundaries are breached and previously private data is released. Assuming this is the case, it is worthwhile to consider designing policies and technology that emphasize accountability rather than impenetrability. Rather than trying to prevent breaches from happening, we should design our systems so that once a breach occurs, it is easy to determine the source of the problem and deal with the data release after the fact.

Specifically, in this case, we want to use the ideas of accountable systems to make data sharing more transparent to citizens. If data about a person is shared between two parties, it should be clear to that person why the data sharing event is compliant under the policies governing the transaction. Instead of giving a binary assertion about the validity of the transaction, it should be possible to show exactly why the compliant was valid under the law. Similarly, if a non-compliant transaction occurs, it should be possible to pinpoint exactly what part of the transaction was fraudulent, and resolve the manner accordingly.

A more comprehensive treatment of accountable systems can be found in the paper by Weitzner et al [3].

III. WORKFLOW OVERVIEW

The developed system is designed to be queried with hypothetical situations, where a user asks if a document can be sent between two parties under any specific policy. It is assumed that both users have profiles detailing their various affiliations and other relevant information cited in the law at hand. In addition, the document is assumed to be annotated with information that describes the content of the document. Finally, it is assumed that there is a transcription of the law into computer-readable policy.

The user gives URIs (Uniform Resource Identifiers) for each of these components to the system through a web interface. The system then returns to the user a justification of whether or not the hypothetical transaction is valid. Regardless of the validity of the scenario, the user can see exactly which pieces of the transaction fit together with which clauses of the policy to cause the compliant or non-compliant result.

IV. COMPONENTS OF A DATA TRANSACTION
A. Rules

The rules of a data transaction are whatever policies are applicable to the given transaction. In reality, it’s often the case that many different rules from different domains apply simultaneously to any given transaction. For example, if one shares data between two different states, it is the case that data protection laws from both states need to be applied to the transaction. Not only that, but these policies may disagree about the vocabulary used to describe the transaction, and these policies may need to pull from completely different data sources in order to reach a sound justification.
In general, each rule has its own ontology and data sources associated with it. For any given transaction, the rule needs to be encoded such that every rule can understand the parts of the transaction, and then execute whatever reasoning it needs to do over its data sources to judge compliance.

B. Data

Historical approaches to applying rules to data focus on the category of data. For example, in the government arena, rules will be applied broadly across categories such as criminal case investigations or sub-categories describing the type of investigation: e.g., drugs, kidnapping, tax fraud. But, the rules we implemented had very different sorts of descriptors. Consider this one segment of a sentence in the Massachusetts law:

"Information shall be provided or made available pursuant to the preceding paragraph only if the individual named in the request or summary has been convicted of a crime punishable by imprisonment for a term of five years or more, or has been convicted of any crime and sentenced to any term of imprisonment, and at the time of the request: is serving a sentence of probation or incarceration, or is under the custody of the parole board..." [4].

This requires a system implementing the rule to know not only that the general class of data being acted upon falls into the broad class of criminal record, but it also requires the ability to represent information from within the data itself such as: the name of the criminal subject, the specific statute(s) under which convicted, the length of the sentence imposed, and the current status of the convict.

1) Entities Described in the Data: This short sub-rule requires the system to be able to identify at least three different kinds of people – people who provide information, people who are the recipients of information, and people who are the subjects of the information. Many systems can handle the first two as system users (discussed more below), but have no mechanism to easily communicate the details of a person within target data. In the Semantic Web, we could say

```xml
<mdccl:convicted_pursuant
```

and by including a second tag for the maximum allowable sentence under that statute and the actual value:

```xml
<mdccl:maximum_allowable_sentence_length>20</mdccl:maximum_allowable_sentence_length>
```

As we progressed in the project, we improved our ability for re-use such that today we could identify the statute and have the policy call to the representation of that conviction statute to find and return the value of the maximum sentence length.

3) Temporal Reasoning: Another determinative fact about the data may require the ability to perform date calculations. Sharing of information is permitted if “at the time of the request: is serving a sentence of probation or incarceration.” We represented this as an instance of the subclass which represents custody status:

```xml
<mdccl:has_custody_status
 rdf:resource="mdccl:Parole"/>
```

but we also included the instance of the sentence imposed:

```xml
<mdccl:sentence_imposed>5</mdccl:sentence_imposed>
```

so that we could later show the ability to calculate the end of the sentence based on the date it was imposed and compare that to the “current” date - the date of the request for a data transaction.

C. Actors

So, too, real rules require the ability to represent details about actors at a finer granularity. Again, Semantic Web technology is well suited to this purpose because it is possible to represent any fact about a user – from the more traditional fixed values of name, organization, and role, to the discoverable or computable ones such as a person’s security keys or deriving the authorized purpose of the employer or comparing the definitions of an attribute from two different jurisdictions (e.g., sending and receiving). Quite frequently, perhaps more often than not, rules about data handling are dependent upon the context of what the individual is doing at that moment. For example, the Maryland law [5] we modeled permits access to information if:

1) In the performance of its function as a criminal justice agency; or

2) For the purpose of hiring or retaining its own employees and agents.

This sort of information may not be inferable from within a system and may need to be collected as an assertion from the actor.

D. Actions

In order to have meaning, a data usage rule must in some way reference what action is being taken vis-a-vis the target
data; it must say an actor can or cannot [do something] with particular data. These rules refer to actions as “collect”, “retain”, “copy”, “share” and “delete”. Often the action is described using common words, such as “disseminate” or “share”, without any definition - for example, whether these terms. Because we were modeling the information sharing environment, we focused on sharing rules for the prototype, but could readily represent other actions.

V. Modeling the Component Parts in RDF

A. Rules (AIR)

The rules in our prototype are represented in the AIR (Accountability in RDF) policy language, as described by Kagel et al [6]. AIR permits the expression of policies as a series of patterns representing criteria to be met for compliance with a particular rule; this works well with legal rules which often are referred to as having “elements”, such as the five fair uses of copyright. The prototype was built using version [x.x] which now permits if-then-else logic (which the earlier version did not). In order that the prototype be accessible for evaluation and validation by a broad array of interested parties (e.g., government executives, policy leaders, lawyers, and the professionals who need to share the information), the sub-rules are coded in the order in which they appear in statute and annotated with their legal citations. This is particularly challenging because law is generated through negotiation and does not generally follow formal logic structures.

A brief note about architecture. We know that some organizations will have the resources and interest to create their own representation of every rule, but that many will opt for a baseline available from a rules library; even in the latter case, there will be law, legal counsel opinion, or policy that is unique to an organization. For this reason, and to demonstrate operation in a decentralized environment, we modeled each organization having a rules library somewhere within the organization’s “cloud”. Further, while we could have triggered rules based upon the person or the action, our prototype identifies the rules associated with the data and triggers the rules when there is an attempt to perform any transaction on the data.

B. Actors (FOAF)

Although LDAP will allow more expansive descriptions of user attributes, the set in a user profile is normally quite limited. We wanted to be able to express essentially anything that might come up in a rule and so chose to adapt FOAF (Friend of a Friend) [7] profiles to represent actors. The FOAF ontology, too, is a relatively short list of attributes, but it is possible to add an unlimited number of additional attributes so long as they are given a URI and, preferably, associated with a definition in an ontology.

Architecturally, we assumed that each organization would continue to control the user profiles of its employees, members, etc. We did not build, but assumed that each organization would ultimately add a security layer which determines how much of a profile to reveal to a requesting system. For example, if person A from organization A asks reasoner A for permission to send data A to recipient B, the rule in organization A is likely to say that data may only be able to send to persons with attributes C; it will be up to organization B’s system to decide whether to reveal that information to reasoner A.

C. Data (PDF/XMP)

Data can be retained in many forms so in the wild we might need to apply RDF to annotate the data in structured or unstructured form. For our prototype, we modeled a series of memos – a request for information about a possible criminal suspect and responses – in PDF with RDF in an embedded XMP file.

VI. Reasoning Over the Transaction

A. Reasoner

The transaction is evaluated against the policy by a backward chaining reasoner. Because of this design choice, the reasoner itself cannot issue calls for more information. Pre-processing must deliver all the necessary data to the reasoner. For example, the prototype automatically identifies to the reasoner the URL for the sender’s profile, the proposed recipient’s profile, and the target data; it also pre-processes by crawling those files for references to other policies or ontologies and delivers those URLs to the reasoner as well. Also, as alluded to earlier, the system searches rules for any assertions that it will need, queries the user, and delivers the result to the reasoner.

B. TMS

The reasoner has incorporated a Truth Maintenance System [7], a dependency tracking mechanism. This allows the system to retain the dependencies upon which it relied to form its conclusions. For our prototype, this is extremely useful because it allows users to see the basis for a decision, a function not available from some other Policy Decision Points (PDPs). Also, it is an efficient mechanism for storing the necessary information for aggregate reporting, risk modeling, or auditing at a later time.

VII. Visibility to Users

A. Input

1) Transaction Simulator: People act on data using many systems and platforms. Rather than separately model transactions in email, various portals, databases, etc., we created a user interface that is intended to provide a [shortcut??] into the middleware, allowing the user to identify the minimal data that would be identified to the accountable system regardless of application or platform – the sender, the target data, and the recipient.

[graphic here?]

In the UI, the sender and receiver are identified by email address, a commonly known identifier, and presumed to be readily linked to the URL for the user profile (FOAF file) by [can’t remember the term]; the individual’s picture and URL are automatically populated on the page. So, too, choosing the
data to be sent causes the UI to find and auto-populate the URL for the applicable policy. If necessary to model a variant, the user can override the policy linked to the data with a different policy.

2) **Tabulator: Views:** Many potential users or evaluators of the technology will not have the skill to read program code. Using a Semantic Web browser, Tabulator, to our input code provides an opportunity for those users to glimpse the meaning of the native RDF. Tabulator has multiple viewing panes including a “FOAF View” which makes it possible to see the user profiles in a visualization that looks more like a list of attributes and a “Table View” which makes it possible to see an AIR policy’s if-then-else structure in a nested chart.

**B. Output**

1) **Tabulator: Views:** The accountable system’s results can also be viewed in special Tabulator panes. The “Justification” pane first opens to a single sentence that indicates if the proposed transaction is compliant or non-compliant with the policy; it builds the sentence by inserting specific instances into an assertion pattern in the code. Pressing the “Why?” button provides the valuable payload from the TMS, expressing each important belief and all of the known facts on which the formation of the belief depended.

The “Lawyer” pane provides a short form of the analysis, auto-generating a series of near-grammatical sentences explaining the requirement of the rule (a significant pattern in the rule), the relevant fact instances that meet or fail to meet that pattern, and the citation for the subsection of the law being applied; the first two are now represented as hyperlinks to the URLs to which they refer.

**VIII. RESULTS**

Our defined goal was to model and execute six scenarios through the reasoner; we accomplished that goal and built a system with sufficient capability and flexibility that it is possible to run previously undefined scenarios (mixing and matching the component pieces in unplanned ways) and also achieve results.

From the research perspective, this exercise served primarily to confirm expectations. First, it proved a certain type of scalability. In our earlier work, we fed to the reasoner only the input necessary to reach a correct conclusion. In this work, we fed the reasoner a significant number of rule patterns and facts that were unnecessary to the conclusion and confirmed, so long as the rule is expressed correctly, that the correct result will be produced – only the appropriate sub-rules will be found to support relevant beliefs and that only the relevant facts will be reported as dependencies. As the “so long as” clause implies, the work showed the importance and necessity of validation, the ability to determine that the rule patterns have been expressed correctly in their entirety – both the pattern and its relationship to all other patterns (e.g., conditions, exceptions, order). We also proved that “broken” or undefined bits did not keep the system from reaching a conclusion (although there was some difficulty keeping the diligent from fixing them). For example, if we run a particular scenario under the Massachusetts criminal records release law and the recipient has a malformed tag which was intended to identify him as a member of a criminal justice agency but fails to do so, the system will correctly determine that he is, by a later sub-rule in the policy, entitled to receive such information as any member of the public may receive.

As part of this research, we also demonstrated the prototype to a variety of relevant persons – ranging from Fusion Center analysts to Intelligence Community management, both technical and operational. The reactions were very positive that such an accountable system could fulfill government obligations to ensure that information sharing is handled in a policy compliant manner and to provide a level of transparency about that.

The most significant resistance received was from an analyst supervisor who perceived this as questioning the ability of individual analysts to know and comply with all rules; however, even that individual believed that the mechanism would be quite helpful when necessary to apply the rules of another jurisdiction (e.g., not one’s own) and/or for use as a workflow management tool. Conversely, the analysts at a demo the next day were so enthusiastic that they wanted to know if they could build and use the FOAF-based user profiles immediately.

**IX. FUTURE WORK**

The reasoner is relatively fast, (processing x, y, and z in aa-bb seconds), but we learned that it cannot produce the millisecond response necessary to use the system as a real-time processor for programs that handle millions of transactions daily, such as border applications for customs and screening passengers. However, the Fusion Centers we spoke to indicated that they were producing sufficiently small numbers of analytical reports per day that waiting some seconds for the evaluation would not be prohibitive. We would like to test other reasoning strategies to reduced the time for throughput.

We also are quite interested in coordinating with other test components and systems. Because the US Constitution establishes state sovereignty, the individual states do not have to follow a federal mandate on the standards for an accountable system and, as a result, we expect that for accountability to be viable there will always be more than one platform, policy language, and tagging scheme in effect. Research is needed to determine the feasibility and strategies for interchange among them.

**X. CONCLUSION**

The conclusion goes here.

**ACKNOWLEDGMENT**

The authors would like to thank...
REFERENCES

[5] “Maryland CMCCL 12.15.01.11.”