Rule-Based Trust Assessment on the Semantic Web

Ian Jacobi[1], Lalana Kagal[1], and Ankesh Khandelwal[2,*]

[1] MIT CSAIL,
Cambridge, MA 02139
{jacobi,lkagal}@csail.mit.edu
[2] Rensselaer Polytechnic Institute,
Troy, NY 12180
ankesh@cs.rpi.edu

Abstract. The Semantic Web is a decentralized forum on which anyone can publish structured data or extend and reuse existing data. This inherent openness of the Semantic Web raises questions about the trustworthiness of the data. Data is usually deemed trustworthy based on several factors including its source, users' prior knowledge, the reputation of the source, and the previous experience of users. However, as rules are important on the Semantic Web for checking data integrity, representing implicit knowledge, or even defining policies, additional factors need to be considered for data that is inferred. Given an existing trust measure, we identify two trust axes namely data and rules and two trust categories namely content-based and metadata-based that are useful for trust assignments associated with Semantic Web data. We propose a meta-modeling framework that uses trust ontologies to assign trust values to data, sources, rules, etc. on the Web, provenance ontologies to capture data generation, and declarative rules to combine these values to form different trust assessment models. These trust assessment models can be used to transfer trust from known to unknown data. We discuss how AIR, a Web rule language, can be used to implement our framework and declaratively describe assessment models using different kinds of trust and provenance ontologies.

1 Introduction

The rise of the Semantic Web and Linked Data, and the machine-understandable interlinked data they promise, has led to an increased reuse of data. Vocabularies such as RDF Schema (RDFS) [7] and the Web Ontology Language (OWL) [2,25] have been developed to enable consumers of Semantic Web data to exchange data with some knowledge of the meaning of this data, allowing not only for the reuse of data, but the reuse of schemas and terminology as well.

Given the inherent openness of the Semantic Web, where anyone can say anything, the reliability and usefulness of web data depends on evaluating its

[*] Author names are arranged in alphabetic order.

trustworthiness. Users need to make decisions about their subjective belief of whether the data is true; such decisions may be based on a number of factors, including which sources to rely on, their prior knowledge, the reputation of the source, and their experience [1]. However, trust assessment becomes challenging when the consumers of this data are applications and agents. In order to automate the assignment of "truthfulness" or trustworthiness measures, it must be possible for trust values to be associated with different aspects of the data such as the actual content of the data, the data sources, recency of updates, the schemas being used, and the creator, and for these, trust values be combined together to evaluate trust in the actual data. For example, there might be multiple Friend Of a Friend (FOAF)[1] files for Tim Berners-Lee that describe his social profile in Resource Description Framework (RDF), but the one that is most trusted is the one available at the W3C website. This is because the trustworthiness of the source, W3C, is higher than that of the other sources. Different trust levels may also be assigned to sources relative to their contents. For example, a hospital may be trusted with information about a potential virus outbreak but may not be trusted with respect to its economic predictions. We suggest that the trust associated with any Web data is some combination of these different trust values associated with the content of the data as well as meta-data about the data such as its source, creator, etc.

Rules are often used with Semantic Web data to check its integrity or represent implicit knowledge as well as to define policies and business logic. As the machine-understandability of Semantic Web data encourages the use of this data by software agents, mechanisms for the automated evaluation of inferred data becomes important. The trustworthiness of inferred data may be evaluated from its provenance, metadata describing how data came to be known. This provenance could be as simple as the source of the data or contain the entire deduction trace. These justifications or deduction traces may provide detailed provenance information, including the data sources, facts used, and rules applied, to allow the evaluation of the particular result. Any number of reasons may exist to assign different levels of trust in a rule or rule set (or the derivations produced therefrom), including differing levels of expertise, familiarity with domain knowledge, or even malicious intent. Thus, the ability to assign or determine a level of *trust in a rule* and its conclusions are required for trust on the Semantic Web. In case of inferred data, the trust value is a combination of trust values associated with its data as well as its rules that in turn can be calculated from the trust associated with their content as well as meta-data.

In this paper we focus on developing models that involve non-statistical functions to assess trust in Semantic Web data in terms of the content and meta-data (source, creator, recency, provenance, reputation, etc.) of data and rules. Along with the trust values associated with the data used, we propose that the trust assignments of rules used in data generation is an important factor in the trust evaluation of the generated data. We suggest that there are two main trust categories for Semantic Web data from which different trust assessment models may

[1] http://www.foaf-project.org/

be derived: **content-based trust** and **metadata-based trust**. In content-based trust, we derive our trust values from the contents of the data or rule itself. The metadata-based trust category is more helpful when calculating trust based on circumstantial facts about the data or rule such as reputation assignments, user ratings, and provenance, rather than the content itself. We propose a *meta-modeling framework* that uses trust ontologies to assign metadata-based and content-based trust values to data, sources, rules, etc. on the Web, provenance ontologies to capture justification/deduction traces, and declarative rules to combine these values to form different trust assessment models. We show how this meta-modeling system can be used to define a range of trust assessment strategies in AIR, a Semantic Web rule language.

This paper is structured as follows: we begin by introducing existing work dealing with trust on the Web. In Section 3, we discuss trust problems on the Semantic Web and discuss different possible trust representations and assessment strategies. The next section provides an overview of the AIR language. In Section 5 we describe how AIR can be used for different trust assessment strategies on the Web. Section 6 identifies the contributions of our work and finally, Section 7 provides a summary and directions for future work.

2 Related Work

Well-known trust management systems such as PolicyMaker [19], KeyNote [6], REFEREE [8], and Delegation Logics [17] view trust management as an authorization problem. That is, they define mechanisms for inferring whether a requester (software or human agent) is permitted to perform a certain action or access a certain resource based on a set of constraints defined by the action/data owner. Our goals are different in that we look at the role of trust in rule based reasoning. Our framework is focused on expressing trustworthiness of data on the open Web, evaluating trustworthiness of inferred data and on allowing mechanisms for decisions based off trust and trust computations to be declaratively specified.

[22,11,12,16] discuss how trust values for users and data sources can be computed. Richardson et al. enable users to maintain trust for other users and provide functions to merge these values into trust values for all users by leveraging the path of trust between users [22]. Kuter et al. allow users to maintain trust values or trust estimates for data sources and provide a probabilistic technique to use that information to compute a trust estimate for a data source [16]. Our approach can be thought of as a meta-modeling approach that allows different trust frameworks to be declaratively developed and possibly combined. It provides a rule language, mechanisms for accessing the Web and cryptographic, math, string and other related functions that may be used to specify how trust is assigned and calculated.

The WIQA framework [5] is also related to the trust assessment framework that we've developed using AIR, however, it is much simpler. The WIQA framework allows RDF data to be filtered according to policies expressed as graph patterns (Please refer to Section 3.3 for more information about patterns) and provides an explanation for this filtering by identifying the matched patterns. Our framework supports more than just filtering as graph patterns are part of rule definitions and filtered data take part in some rule based reasoning. Furthermore, data may be filtered not just based on some patterns but also based on trust assigned (or assessed) to data as well as patterns.

SAOR [13] and Straccia et al. [24] incorporate trust in rule based reasoning. They employ different trust representations and use trust differently. While Straccia et al. assume that every triple is annotated with trust (and other annotations such as fuzziness), SAOR doesn't consider trust valuation of triples in isolation. At the rule application time, trust on a triple is evaluated based on what it is being used to prove, and the trust value is binary, i.e. a triple is either considered authoritative for that instance of rule application or is not used for derivation. In contrast every triple is used for derivation in the framework proposed by Straccia et al. and the inferred triples are associated with trust derived from that of triples required for inferring it. AIR reasoner is not trust aware in the sense that SAOR and the framework proposed by Straccia et al. are, but we show that the language features of AIR give a lot of freedom to reason about knowledge base annotated with trust. In the approach proposed, the trust valuations are incorporated in the rule conditions and may be used for different affects. For instance a pattern with trust smaller than 7 may be rejected for one rule, and accepted with lower trust value of 6 for other (because it is rejected for trust value less than 5). Furthermore, we treat rules as part of the knowledge that people may have varying degrees of trust.

3 Semantic Web and Trust

The reliability and usefulness of Web data depends on evaluating its trustworthiness, the subjective measure of the belief which a user has that the data is "true". Our approach supports the vision provided by the Semantic Web layer-cake[2] by building trust out of rules about provenance and proofs/justifications related to data on the Semantic Web. Thus, in order to better understand how trust might be modeled in the Semantic Web, it is important to understand the underlying concepts employed by the Resource Description Framework (RDF), which serves as the foundation for all Semantic Web data.

3.1 Resource Description Framework (RDF)

Resource Description Framework (RDF) is the data modeling framework for the Semantic Web and it uses 3-tuples, or triples, to represent facts. RDF is described in more detail in [3], but we will proceed to give a short outline below.

Each triple in an RDF model consists of a subject, predicate, and object, much like the subject, predicate, and object of a natural language sentence. For example, one possible triple representing the rating of a movie, **:citizenkane**

[2] http://www.w3.org/2004/Talks/0412-RDF-functions/slide4-0.html

:stars "5".[3], consists of a subject **:citizenkane**, a predicate **:stars** and an object **"5"**. Triples may also be thought of as logical predicates taking two arguments, such as $stars(citizenkane, 5)$.

In the RDF data model, each triple may be thought of as a directed edge in a labelled *RDF graph*, where the subject and object are nodes in the graph, and the predicate is a labelled edge. Nodes may be uniquely identified by a Uniform Resource Identifier (URI), and thus, subjects and objects may be specified by a URI. Nodes may also be made anonymous, without such an identifier. Such anonymous nodes are called blank nodes, or bnodes.

Predicates are also identified by a URI. Unlike subjects and objects, however, these URIs do not identify a unique edge, but rather identify the type/meaning of the edge linking the two entities. Objects may also be a *literal*, that is, a string or a simply-typed object (such as an integer or date), but these do not uniquely identify a node.

3.2 Models of Trust for the Semantic Web

When speaking of a trust metric, $\mathcal{T}$ (a quantitative measurement of trust), which is applicable to the RDF data derived from rules, we must ground such trust in the two inputs needed to draw such conclusions: trust in the data from which the conclusion was drawn, $\mathcal{T}(f)$, and trust in the rule which generated the conclusion, $\mathcal{T}(r)$. These two axes of trust are independent of each other, but must be considered together in order to draw a meaningful idea of the trust that may be placed in any conclusions.

If we consider rules to be black boxes, we must necessarily separate trust in rules from trust in data in this way. Any inferences generated by a rule may be generated locally or by a third party in exactly the same way. Because inferences may be generated by an unknown third party, it is difficult to offer strong guarantees about the trust of any inferences made. RDF and rule systems do not innately have any semantics pertaining to trust, and, as trust is subjective in any case, it is difficult to offer any universal guarantees in how input data might be used or how output data might be generated in order to determine the level of trust that might apply to a particular inferred fact. Thus, as in certain modal logics of trust, we must ground the degree of trust in the output data in the degree of trust we have in the particular rule which generated that particular data [18].

Similarly, provenance-based approaches to trust evaluation necessarily consider the "paths" by which data came to be (i.e. from whom data came from, as well as the processes which generated them, which may include rule systems) [9]. Thus, when calculating the trust in a derived fact, we must necessarily consider the trust in the rule which generated the conclusion, $\mathcal{T}(r)$ in any meta-modeling system capable of assessing trust.

[3] Throughout this paper, we use Turtle syntax for RDF (`http://www.w3.org/TeamSubmission/turtle/`), which is a subset of the Notation 3 syntax compatible with the RDF abstract syntax.

Furthermore, as rules may depend on external knowledge to create conclusions, our trust in these conclusions must, necessarily, be no greater than our trust in this input knowledge. This differs from our trust in rules in much the same way that a quantitative process or algorithm may generate precise results without necessarily being accurate (perhaps due to some bias in the input data). Thus, while a trusted rule may generate reliable output data (i.e. it is precise), the output data may only be reliable for other purposes to the extent that the input data is reliable (i.e. its results may not be accurate).

Although both axes should be considered when determining whether or not to trust a conclusion, the trust model used for drawing such conclusions may vary from one "invocation" of a rule to the next. The trust model used for one axis may differ from that used for the other, but both generally must consist of a synthesis of two different categories of trust: **content-based trust**, and **metadata-based trust** [10,5].

In content-based trust, we derive our trust values from the contents of the facts asserted. Thus, any metric $\mathcal{T}$ would be defined in terms of one or more facts, $f \in \mathcal{F}$, the set of all facts known. For example, one trust metric might determine trust in external statements about the actors in certain movies if the source happens to agree with certain statements about the directors of the movies known a priori ($\mathcal{T} \sim |\mathcal{F}_{\text{known}} \cap \mathcal{F}_{\text{source}}|$). Similarly, one might place a higher trust value on statements which use a well-defined and well-used ontology rather than an ill-defined one ($\mathcal{T} \sim \forall_{f \in \mathcal{F}_{\text{source}}} predicate(f)$).

Metadata-based trust is more helpful when calculating trust based on circumstantial facts about the data, rather than facts in the data itself. Each fact $f \in \mathcal{F}$ may be considered to have a vector of corresponding "metadata" facts $M(f)$ which describe additional information regarding the fact, such as authorship, creation times, data sources, and other such derivative data. Metadata-based trust subsumes all data that might describe a given fact, including the provenance of the fact and any user-generated trust values or ratings of the source from which the fact is derived. Any of these metadata-facts may then be used to calculate $\mathcal{T}$.

For example, one may have a higher trust value in statements made by one's friends than those made by arbitrary people ($\mathcal{T} \sim M_{\text{source}}(f)$). Such a trust value depends not on the data being said, but rather on the *source* of the data. Similarly, if, during a science experiment, a bad sensor is identified and replaced, different trust values may be assigned to the data recorded at different times in the experiment ($\mathcal{T} \sim time - M_{\text{time}}(f)$). In this case, the trust value depends on the *time* data was collected, $M_{\text{time}}(f)$, rather than in the data itself (which may have no information indicating its accuracy or lack thereof).

Content-based trust and metadata-based trust may be synthesized in any number of ways to create a trust metric. For example, we may trust statements made about the actors of the movie to a different degree from statements about the creators of the movie. In constrast, metadata-based trust allows us to assign different trust values to statements made by the creators of the movie, Avatar, separate from statements made by the actors of the movie. Thus content-based trust is associated with the statements themselves whereas metadata-based trust may be based on facts related to the statements such as their authors.

As mentioned previously, these two trust categories apply not only to the facts that caused the deduction of new facts, but also to the rules themselves. For example, we may have metadata-based trust in a rule and be able to trust rules that deduce information about a movie's rating that have been written by a movie critic, but not necessarily the same rules written by a director interested in promoting his own movies.

Any language seeking to be used for the purpose of calculating trust in rules or using trust levels to make deductions must be able to model trust metrics not only based on the rules and facts, but it should be able to synthesize trust values based on both metadata about the data, and the data itself. We believe that the AIR rule language is capable of doing so, and we will illustrate this in the following sections.

3.3 Possible Trust Representations in RDF

Given any trust measure, there are numerous ways trust assignments may be made using RDF. One of the simplest ways to declare binary trust is by defining two classes of trust such as *Trusted* and *UnTrusted* and declare URIs, sources, rules, or any resource, to be one or the other. We demonstrate such a model in example (a) in Figure 1, where Isabel trusts two resources, `:RobertEbert` and `:Karl`, but does not trust the resource identified by the URI, <http://example.org/critix.n3>.

The above trust assignment is simple and only allows users to classify resources as trusted or not. In order to have finer grained trust values, it is possible to define a property such as *trustvalue* and use it to assign values (either quantitative or qualitative) to resources. This property can also be used to model fuzzy RDF where triples are annotated with a degree of truth in [0, 1] as defined by Straccia et al. [24]. For example, 'Rome is a big city to degree 0.8' can be represented in Notation 3 (Please refer to Section 4.1 for more information about Notation 3) as **{:Rome :a :BigCity} :trustvalue 0.8**. In example (b) from Figure 1, Isabel defines trust as her confidence in the accuracy of the data/resource and associates trust values with `:RogerEbert`, <http://example.org/critix.n3>, a rule, `:KarlWatchRule`, and a statement about CitizenKane.

Instead of using a quantitative approach as above, an alternate approach would be to create properties for every discrete type of trust possible such as the model for social networks proposed by Golbeck et al. [12]. Golbeck defines individual properties for different trust relationship between users — distrusts absolutely, distrusts highly, distrusts moderately, distrusts slightly, trusts neutrally, trusts slightly, trusts moderately, trusts highly, and trusts absolutely. In example (c) from Figure 1, Isabel highly trusts `:RogerEbert` but trusts `:Karl` moderately.

Additionally, it is possible to trust certain sources or documents with certain information but not all information they contain. For example, a hospital may be trusted with information about a potential virus outbreak but may not be

```
# Example (a)

# Isabel's trust declaration
:RogerEbert a :Trusted .
<http://example.org/critix.n3> rdf:type :Untrusted .
:Karl a :Trusted .

# Example (b)

# Isabel's trust declaration
:RogerEbert :trustvalue 5 .
<http://example.org/critix.n3> :trustvalue 9 .
:KarlWatchRule :trustvalue 7 .
{:CitizenKane :rating :ThumbsUp} :trustvalue 8 .

# Example (c)

# Isabel's trust declaration
:Isabel :trustsHighly :RogerEbert .
:Isabel :trustsModerately :Karl .

# Example (d)

# Isabel's trust declarations
<http://example.org/critix.n3> :istrustedwith :some-t.
:some-t rdf:type :TrustInfo;
    :tval 95;
    :tpattern  { :RogerEbert ?p ?o } .
```

Fig. 1. Trust Representations

trusted with respect to its economic predictions. This could be modeled in Notation 3 (Please refer to Section 4.1 for more information about Notation 3) in several ways, one of which is to use quoted formulae [4]. A property, *istrustedwith* is defined whose object is of type *TrustInfo* and is associated with data sources. The *TrustInfo* class has two properties *tpattern* and *tval.* The *tpattern* is a graph pattern that describes the facts that are trusted from that particular source. The *tval* is similar to the *trustvalue* property and is a trust value associated with all those facts that match the (graph) pattern. The example (d) in Figure 1 shows how these properties are defined and Isabel's trust declaration states that she assigns a quantitative trust value of 95 to all statements made about `:RogerEbert` described in document <http://example.org/critix.n3>.

4 AIR Web Rule Language

AIR is made up of a set of built-in functions and two independent ontologies — the first is for the specification of AIR rules, and the second deals with describing justifications for the inferences made by AIR rules [14]. The built-in functions allow rules to access (Semantic) Web resources, query remote RDF databases, as well as to perform basic math, string and cryptographic operations. We describe the syntax and functionality of the AIR language and the Rule ontology next. The AIR justification ontology is derived from PML [20] and customized for AIR reasoning, and is described in detail at [15]. However, we will use only the PML vocabulary in the paper, as the specifics of AIR reasoning are not important.

```
@prefix s: <http://s.example.org/ontology#> .
@prefix a: <http://a.example.org/instance#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix : <http://src1.example.org/JohnAnalyst#> .

:Jim rdf:type foaf:Person.
:Matt rdf:type foaf:Person.

a:StarWars s:rec s:Watch.
a:TheRoom s:rec s:DontWatch.
a:TheGraduate s:rec s:Watch.

:RogerEbert :said {a:CitizenKane :rating :ThumbsUp}.
```

Fig. 2. Content of an RDF Data Source

4.1 Syntax

AIR extends Notation 3 (N3) [3], a syntax based on the RDF abstract syntax. N3 makes use of a number of basic concepts from RDF, including the concept of triples. N3 extends RDF's abstract syntax by adding formula quoting, which allows for RDF Graphs to be treated as subjects or objects, and variable quantification. Figure 2 provides some examples of N3 statements. Please refer to `http://www.w3.org/2000/10/swap/Primer` for an overview of N3.

AIR uses N3's formula quoting and variable quantification to describe graph patterns, which are similar to the Basic Graph Pattern (BGP) of SPARQL queries[4], that are to be matched before rules may fire.

4.2 Rule Ontology

The AIR rule vocabulary consists of several key classes and properties. ***Belief-rule*** is a class of resources representing the set of all rules. These rules may then have the properties ***if***, ***then***, and ***else*** associated with them to represent the N3 pattern to be matched, and the actions to take if the pattern matches, or does not match, respectively. ***then*** and ***else*** actions may be described in terms of the facts they assert (using the ***assert*** property) or the rules they cause to match next (using the ***rule*** property).

If the graph pattern or condition matches the current state of the world, defined as the facts known or inferred to be true so far, then all the actions under ***then*** are fired, otherwise all the actions under ***else*** are fired. The condition matches the current state if there is a subgraph of known facts that matches the graph pattern.

Figure 3 demonstrates how the AIR rule vocabulary can be used to define the recommendation rule Isabel uses. The rule suggests that Isabel only watch movies, ***air:assert { :Isabel s:shouldWatch :MOVIE}***, if there is a five-star rating, ***:MOVIE s:starRating 5*** . The same rule is illustrated in a tabular representation in Figure 4. For clarity we use this tabular representation of AIR rules for the remainder of this paper.

[4] `http://www.w3.org/TR/rdf-sparql-query/#BasicGraphPatterns`

```
@forAll :MOVIE.
:IsabelWatchRule a air:Belief-rule;
   air:if { :MOVIE s:starRating 5  };
   air:then [ air:assert { :Isabel s:shouldWatch :MOVIE } ].
```

Fig. 3. Example AIR Syntax

```
     :IsabelWatchRule a air:Belief-rule .

     @forAll :MOVIE .

          # Implicitly trust all 5-star ratings.
     IF
          :MOVIE s:starRating 5 .

     THEN assert :Isabel s:shouldWatch :MOVIE .
```

Fig. 4. Example AIR Rule: `IsabelWatchRule` implicitly encodes **content-based trust** into the rule by explicitly trusting all stated 5-star ratings to the full degree without regard to how the ratings were generated or who stated them. Although there is no reliance on trust values of the statement or its pattern itself, it is the same as if complete trust was placed in all such statements.

5 Trust Assessment Framework

Our trust assessment framework is not restricted to any specific trust measure such as reputation, degree of truth, or completeness. As long as the trust assignment can be captured in RDF, it can be used by our framework. However, it is up to the user developing the trust assessment model to ensure that the semantics associated with different trust assignments is maintained when combining different trust values.

In this section we show how our framework can be used to evaluate trust in a movie recommendation scenario. Assume that a movie streaming service, WebCinema, offers several different methods of movie recommendations to its clients. Isabel, a member of WebCinema, uses one of WebCinema's metrics, which recommends movies that have at least one five-star rating. WebCinema permits users to select the critics giving the five-star ratings, so as to ignore reviews from critics users disagree with, and Isabel has chosen to make use of this feature. Some of WebCinema's customers may create their own rules which merge the results of several of WebCinema's built-in rules. Karl uses WebCinema's rating-based rules as part of his decision making process, but does not entirely trust them. Karl may wish to only partially trust WebCinema's ratings-based rules, depending additionally on the trust he places on the facts used by the rules to make recommendations.

For this scenario, we assume that trust declarations are made using the *istrustedwith* property, where we trust a source with respect to certain data, and the *trustvalue* property, where we associate a trust value with a resource, as defined in Section 3.3. The rule in Figure 5, `:IsabelWatchRule`, uses metadata-based trust and assigns trust to sources of data with respect to watch recommendations and creators of data. The rule looks for patterns referring to watch recommendations in sources whom she trusts more than 75 with the specified pattern

```
:IsabelWatchRule a air:Belief-rule .

@forAll :CRITICREVIEWS, :VARIABLE, :MOVIE, :STRUST,
:CREATOR, :CTRUST, :VAL, :TRUST .

     :CRITICREVIEWS
          t:istrustedwith [ t:pattern { :VARIABLE s:rec s:Watch . } ;
                   t:value :STRUST ] .
     :STRUST math:notLessThan 75 .
     :CRITICREVIEWS log:includes { :MOVIE s:rec s:Watch . }.
IF   :CRITICREVIEWS foaf:maker :CREATOR .
     :CREATOR t:trustvalue :CTRUST .
     (:STRUST 100) math:integerQuotient :VAL .
     (:CTRUST :VAL) math:product :TRUST .
     :TRUST math:notLessThan 80 .

THEN assert :Isabel s:shouldWatch :MOVIE .
```

Fig. 5. Compute trust in data using metadata-based trust: `IsabelWatchRule` uses the trust ontology described in section 3.3 to assign trust to critics with respect to watch recommendations. The source is only searched if the trust in the source is greater than 75. The trust in the new information is calculated from the trust in the source website and creator of the information.

{ :VARIABLE s:rec s:Watch . }. If the source contains this information, the rule calculates trust for this new information based on her trust in the source and the creator of the data and recommends that she watch it if the trust is greater than 80.

As AIR is a general purpose reasoner, AIR rules can be written to consume these trust declarations and combine them in different ways in order to compute trust values for data or inferences of interest. `IsabelWatchRule` as defined in Figure 4 uses metadata-based trust and assigns trust to critics with respect to watch recommendations. `:IsabelWatchRule` recommends that Isabel watch a movie if there is a watch recommendation made by a critic with trust value greater than 7.

As rules themselves could have trust values associated with them, as described in Section 3.3, it is possible to reason about the trustworthiness of *rules* and use them to deduce trust in the inferences made by them. AIR's support for the ***air:justifies*** property allows for the execution of other rules which we may be able to query for trust. The rule `:KarlWatchRule` in Figure 6 encapsulates such a rule (which synthesizes rule-trust with data-trust); it recommends that Karl watch a movie only if both the rule and data used to justify the recommendation are trusted by Karl with an average trust value greater than or equal to 7.

In `:KarlWatchRule`, ***air:justifies*** is used to run the rules at the URL <http://webcinema.example.com/rules> against some trusted data. The result of this reasoning is stored in the output variable `:RULEJUST`, which may be used with other built-in functions, like ***log:includes***, to determine not only which facts are asserted by the rules, but also the justifications for such. These justifications may then be used to determine the rules which caused some conclusion to be found to be true and their trust values.

```
:KarlWatchRule a air:Belief-rule .

@forAll :RULESET, :RULEJUST, :MOVIE, :RULEAPPEVENT,
     :EXTRACTIONEVENT, :SOURCE, :RULE, :RULETRUST, :SOURCETRUST,
     :TRUSTSUM, :TRUSTAVG .

     <http://webcinema.example.com/rules> log:semantics :RULESET .
     ((:RULESET) (:DATA)) air:justifies :RULEJUST .
     :RULEJUST log:includes {
          @forSome :RULEAPPEVENT .
          :Karl s:shouldWatch :MOVIE .
          :RULEAPPEVENT pmlj:outputdata { :Karl s:shouldWatch :MOVIE . } .
          :RULEAPPEVENT pmll:operation :RULE .
IF        :RULEAPPEVENT pmll:antecedent :EXTRACTIONEVENT .
          :EXTRACTIONEVENT pmlp:source :SOURCE .
     } .
     :RULE t:trustvalue :RULETRUST .
     :SOURCE t:trustvalue :SOURCETRUST .
     (:RULETRUST :SOURCETRUST) math:sum :TRUSTSUM .
     (:TRUSTSUM 2) math:quotient :TRUSTAVG .
     :TRUSTAVG math:notLessThan 7 .
THEN assert :Karl s:shouldWatch :MOVIE .
```

Fig. 6. Compute trust in data using metadata-based trust in rules: `KarlWatchRule` uses the trust ontology described in section 3.3 to assign trust to the rules used by WebCinema to generate recommendations. If the average trust in the rule used to generate a watch recommendation and the data used by the rule has a trust value greater than or equal to 7, then the rule recommends that Karl should watch that movie.

We could use a similar rule to more generally judge the output of rules based on where the output came from. While the above rule checks trust values in the rule and source individually, rules could also be written to expressly generate trust based on properties of the rule or data, as well as the trust measurements themselves (for example, we could check the "author" of a rule to implicitly trust all rules authored by WebCinema, giving their results a high trust value.)

Provenance models such as the Open Provenance Model (OPM) [21] or Provenir [23] are also supported by our framework. OPM and Provenir are high-level, general-purpose provenance models that may be encoded in RDF and be queried using the AIR language in a manner similar to that shown above. As long as it is possible to identify the URIs of rules used in a particular fact's derivation, one need only express an appropriate pattern which may be used to find and bind the rule's identifier to search for an appropriate trust value for the rule.

For example, content-based trust models may be used together with languages like OPM and Provenir to identify and determine trust in particular products of generic scientific processes. The faulty scientific sensor example discussed in Section 3.2 may be easily implemented in our framework when the provenance is encoded in OPM as can be observed in the sample rule in Figure 7.

```
:BadSensorRule a air:Belief-rule .

@forAll :DATA, :PROCESS, :TIME, :UNIXTIME .

     :DATA a opm:Artifact ;
          opm:wasGeneratedBy :PROCESS ;
IF        opm:wasGeneratedAt :TIME .
     :PROCESS owl:sameAs :BadSensorProcess .
     :TIME time:inSeconds :UNIXTIME .

THEN activate-rule :BadSensorTimeRule
-----------------------------------------------------
:BadSensorTimeRule a air:Belief-rule .

IF   :UNIXTIME math:greaterThan :BadSensorFixedTime .

THEN assert :DATA t:trustvalue 7.

ELSE assert :DATA t:trustvalue 3.
```

Fig. 7. Deriving trust from OPM provenance: `BadSensorRule` uses provenance metadata about some datum encoded in OPM to assign trust to the datum depending on the time the data was generated by the faulty sensor

6 Contributions

As demonstrated above, our framework has several unique features that make it useful for trust assessment and modeling thanks to its foundation in the Semantic Web. First, its use of the RDF data model and ability to uniquely identify and specify rules allows for the augmentation of existing rulesets with trust data. Trust values and metrics may be defined separate from the data for which trust is being assigned. This also allows third-party representations of trust, which may not be possible in all trust model implementations. Furthermore, its rule-based nature allows for more nuanced trust-assessment on the web than a simple binary trust model, although binary trust models are also supported.

Although the examples in this paper make use of specific terms for trust and PML and OPM ontologies for provenance, the general principles employed in the motivating use case implementations can be used with **any other trust and provenance vocabularies** that have an RDF representation. Our framework may serve to construct and evaluate trust metrics in general, regardless of the vocabularies used to encode provenance and trust.

Second, as a generic Semantic Web rule language, AIR is capable of reasoning over several different semantic representations of trust metrics, including straightforward numerical values like that in Figure 7 and trust values assigned on a per-pattern basis, such as in Figure 5. Thus, existing trust metrics may be integrated with framework needing only a suitable mapping into the RDF data model, and users may thus choose a trust model that best captures their requirements.

Finally, our framework identifies two trust categories, namely content-based and metadata-based, for assigning trust and recognizes that trust in rules is as important as trust in data on the Semantic Web as rules are used frequently to make inferences over Web data. So far as we can tell, there exists no literature regarding these categories or axes of trust within the context of rule systems.

7 Summary and Future Work

In this paper, we discussed the importance of trust on the Semantic Web and identified two trust axes namely data and rules and two trust categories namely content-based and metadata-based that are useful for trust declarations associated with Semantic Web data. Furthermore, we outlined a meta-modeling trust framework and demonstrated how an implementation of this framework in the AIR Web rule language could be used to develop different trust assessments models.

Though this work demonstrates the usefulness of AIR, it relies on user-generated rules for handling trust. As part of our future work, we will work on general rules that will handle trust transparently such that users do not need to explicitly know about or handle trust in their systems but will be able to customize these rules to do it for them. We also intend to evaluate the addition of trust as a first-class entity within AIR, potentially adding explicit support for trust in built-in functions. Although the benefits to adding trust as an implicit part of a rules language may be great, it is possible that the flexibility of trust models is preferable to forcing one particular trust model on the language.

Acknowledgements

This paper is based upon work supported by the National Science Foundation under Award No. CNS-0831442 and by the Air Force Office of Scientific Research under Award No. FA9550-09-1-0152.

References

1. Artz, D., Gil, Y.: A Survey of Trust in Computer Science and the Semantic Web. Web Semantics 5, 58–71 (2007)
2. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference, W3C Recommendation (February 10, 2004), `http://www.w3.org/TR/owl-ref/`
3. Berners-Lee, T.: Primer: Getting into RDF and Semantic Web using N3 (2005), `http://www.w3.org/2000/10/swap/Primer`
4. Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y., Hendler, J.: N3Logic: A Logical Framework For the World Wide Web. Journal of Theory and Practice of Logic Programming (2007)
5. Bizer, C., Cyganiak, R.: Quality-driven information filtering using the WIQA policy framework. Journal of Web Semantics 7, 1–10 (2009)
6. Blaze, M., Feigenbaum, J., Ioannidis, J., Keromytis, A.: The KeyNote Trust Management System Version. Internet RFC 2704 (September 1999)
7. Brickley, D., Guha, R.: RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation (February 2002), `http://www.w3.org/TR/rdf-schema`

8. Chu, Y.-H., Feigenbaum, J., LaMacchia, B., Resnick, P., Strauss, M.: REFEREE: Trust management for Web Applications. Computer Networks and ISDN Systems 29(8-13), 953–964 (1997)
9. Dai, C., Lin, D., Hwang, J., Kantarcioglu, M.: An Approach to Evaluate Data Trustworthiness Based on Data Provenance. In: Jonker, W., Petković, M. (eds.) SDM 2008. LNCS, vol. 5159, pp. 82–98. Springer, Heidelberg (2008), doi:10.1007/978-3-540-85259-9_6
10. Gao, Q., Houben, G.-J.: A Framework for Trust Establishment and Assessment on the Web of Data. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 1097–1098. ACM, New York (2010)
11. Gil, Y., Ratnakar, V.: Trusting Information Sources One Citizen at a Time. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 162–176. Springer, Heidelberg (2002)
12. Golbeck, J., Parsia, B., Hendler, J.: Trust Networks on the Semantic Web. In: Proceedings of Cooperative Intelligent Agents, pp. 238–249 (2003)
13. Hogan, A., Harth, A., Polleres, A.: SAOR: Authoritative Reasoning for the Web. In: Domingue, J., Anutariya, C. (eds.) ASWC 2008. LNCS, vol. 5367, pp. 76–90. Springer, Heidelberg (2008)
14. Kagal, L., Hanson, C., Weitzner, D.: Using Dependency Tracking to Provide Explanations for Policy Management. In: IEEE Policy 2008 (2008)
15. Kagal, L., Jacobi, I., Khandelwal, A.: Gasping for AIR: Why we need linked rules and justifications on the Semantic Web. Technical Report MIT-CSAIL-TR-2011-023, Massachusetts Institute of Technology (April 2011)
16. Kuter, U., Golbeck, J.: SUNNY: a new algorithm for trust inference in social networks using probabilistic confidence models. In: Proceedings of the 22nd National Conference on Artificial Intelligence, pp. 1377–1382. AAAI Press, Menlo Park (2007)
17. Li, N., Grosof, B.N., Feigenbaum, J.: Delegation Logic: A Logic-based Approach to Distributed Authorization. ACM Transactions on Information Systems Security (TISSEC) 6(1) (February 2003)
18. Liau, C.-J.: Belief, information acquisition, and trust in multi-agent systems–A modal logic formulation. Artificial Intelligence 149, 31–60 (2003)
19. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized Trust Management. In: Proceedings of IEEE Conference on Privacy and Security (1996)
20. McGuinness, D.L., Ding, L., da Silva, P.P., Chang, C.: PML 2: A Modular Explanation Interlingua. In: AAAI 2007 Workshop on Explanation-aware Computing (2007)
21. Moreau, L., Plale, B., Miles, S., Goble, C., Missier, P., Barga, R., Simmhan, Y., Futrelle, J., Mcgrath, R.E., Myers, J., Paulson, P., Bowers, S., Ludaescher, B., Kwasnikowska, N., Bussche, J.V.D., Ellkvist, T., Freire, J., Groth, P. (eds.): The Open Provenance Model (v1.01) (2008), `http://eprints.ecs.soton.ac.uk/16148/1/opm-v1.01.pdf`
22. Richardson, M., Agrawal, R., Domingos, P.: Trust Management for the Semantic Web. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 351–368. Springer, Heidelberg (2003)
23. Sahoo, S.S., Sheth, A.: Provenir ontology: Towards a Framework for eScience Provenance Management. In: Microsoft eScience Workshop (October 2009)
24. Straccia, U., Lopes, N., Lukacsy, G., Polleres, A.: A General Framework for Representing and Reasoning with Annotated Semantic Web Data. In: AAAI (2010)
25. W3C OWL Working Group. OWL 2 Web Ontology Language, W3C Recommendation (October 27, 2009), `http://www.w3.org/TR/owl2-overview/`