Fox, M.S., and Huang, J., (2003), "Knowledge Provenance: An Approach to Modeling and Maintaining the Evolution and Validity of Knowledge", EIL Technical Report, University of Toronto.

Knowledge Provenance: An Approach to Modeling and Maintaining The Evolution and Validity of Knowledge

Mark S. Fox and Jingwei Huang

Enterprise Integration Laboratory, University of Toronto
40 St. George Street, Toronto, ON M5S 3G8 Canada
msf@eil.utoronto.ca
www.eil.utoronto.ca
22 May 2003

Abstract. This paper addresses the problem of how to determine the validity and origin of information/knowledge on the web. Knowledge Provenance is proposed to address this problem. Four levels of Knowledge Provenance are introduced: Static, where the validity of knowledge does not change over time; Dynamic, where validity may change over time; Uncertain, where the validity of knowledge is uncertain; and Judgmental: where the societal processes for determining certainty of knowledge are defined. . An ontology, semantics and implementation using RDFS is provided for Static Knowledge Provenance.

1 Introduction

This paper addresses the problem of how to determine the validity of information/knowledge on the web. The problem arises from many directions: information may no longer be relevant (e.g., discontinued products or old operating procedures), may contain incorrect information (e.g., news stories), and may even be outright lies. For example, in 1999, two men posted fraudulent corporate information on electronic bulletin boards, which caused the stock price of a company (NEI) to soar from $0.13 to $15, resulting in their making a profit of more than $350,000 [Mayorkas, 2000]. Currently, anyone can publish information on the web; the information may be true or false, uncertain or dated, but no tool exists to discern the differences.

In this paper, Knowledge Provenance (KP) is proposed to address this problem by introducing standards and processes for how to model and maintain the evolution and validity of web information/knowledge. The major questions that need to be answered in KP are: For any piece of web information, where does it come from? Who created it? Can it be believed?

Philosophically, we believe the web will always be a morass of uncertain and incomplete information. But we also believe that it is possible to annotate web content to create islands of certainty. Towards this end, Knowledge Provenance introduces 4 levels of Provenance that range from strong provenance (corresponding to high certainty) to weak provenance (corresponding to high uncertainty). Level 1 (Static KP) focuses on provenance of static and certain information; Level 2 (Dynamic KP) considers how the validity of information may change over time; Level 3 (Uncertain KP) considers information whose validity is inherently uncertain; Level 4 (Judgment-based KP) focuses on social processes necessary to support provenance. This paper focuses on Static KP.

The focus of this paper is on Level 1: Static Knowledge Provenance. Static Knowledge Provenance provides the fundamental building blocks for determining validity, i.e., truth, traceability, and trust, on which higher levels of Knowledge Provenance are constructed.

The content of this paper is organized as follows: Section 2 provides a brief discussion of related research; section 3 provides motivating scenarios for Knowledge Provenance. Section 4 introduces an Ontology for Static KP (Level 1). Section 5 illustrates how web authors can annotate web documents to enable reasoning about Knowledge Provenance. Finally, we provide a summary and a view on future work in section 6.

2 Related Research

Information validity in the context of the web has been a popular concern from the outset. Interest in addressing the issue has appeared under the umbrella of the "Web of Trust" [Khare & Rifkin 97]. (Similar concerns exist in the Library and Information Sciences world [Bearman & Trant 98].) Currently the major concerns in "Web of Trust" are digital signature and digital certification [W3C XMLSig]. However, they only provide an approach to validate author identification and information integrity. In the context of knowledge provenance, they can only

be used to determine who is the information creator and if the information is the same as originally defined, but they do not indicate if the information is true and what it depends on. Though no direct solution for knowledge provenance has been proposed, a number of relevant technologies exist.

Our approach to truth and traceability is based upon earlier work in Artificial Intelligence, called Truth Maintenance Systems (TMS). TMS provides an approach to representing the truth value of a proposition and what the proposition's truth value depends on [deKleer et al., 1989].

Finally, our implementation of validity is based on RDF (Resource Description Framework) which provides an embeddable means for representing Web resource metadata.

3 What is Static Knowledge Provenance?

The basic unit of web information to be considered in KP is a "proposition". A proposition, as defined in First Order Logic, is a declarative sentence that is either true or false. A proposition is the smallest piece of information to which provenance-related attributes may be ascribed.

Static Knowledge Provenance focuses on the simplest yet strongest form of provenance. Basically, any proposition has a truth value of: True, False or Unknown. The default truth value is "Unknown". -- A Static proposition's truth value does not change over time. (Dynamic truth values are considered in Dynamic KP.) ;

In the following, the underlying concepts of Static Knowledge Provenance are explored in the context of two case studies.

Case 1: Asserted Information

Consider the proposition found on a web page that "perennial sea ice in the Arctic is melting faster than previously thought at a rate of 9 percent per decade." From a provenance perspective, there are three questions that have to be answered: 1) What is the truth value of this proposition? 2) Who asserted this proposition? 3) Should we believe the person or organization that asserted it? In this example, a further examination of the text of the web page provides the answers (www.gsfe.nasa.gov/topstory/2002/1122seaice.html): It is a true proposition, asserted by NASA, who most people believe is an authority on the subject. Question is, how can this provenance information be represented directly without having to resort to Natural Language Processing of the page?

Other examples of asserted information include assertions made by persons or organizations, statistical data and observation data such as stock quotes and weather readings issued by organizations. In addition, commonly recognized knowledge, such as scientific laws, are regarded as "asserted information". Inaddition, the derivation of scientific laws have been validated in the past, and needn't to be validated again, even though scientific laws are usually regarded as "derived information".

Case 2: Dependent Information

Consider the following proposition found in another web page: "The accelerated rate of reduction of perennial sea ice in the Arctic will lead to the extinction of polar bears within 100 years." This is actually two propositions composed of a premise, "The accelerated rate of reduction of perennial sea ice in the Arctic" and a conclusion, "the extinction of polar bears within 100 years." Just as in the previous case, there are three questions that need to be answered: 1) What is the truth value of these propositions? 2) Who asserted them? 3) Should we believe the person or organization that asserted them? What makes this case more interesting is that answering these question is dependent upon propositions found in other web pages. There are two types of dependency occurring. First the truth of the premise is dependent on the truth of the proposition found in another web page. Secondly, the truth of the conclusion depends on the truth of the premise and upon some hidden reasoning that led to the deduction. These types of propositions are called "dependent propositions" in KP.

It is common to find information in one document reproduced in another. The reproduction of a proposition in a second document leads to an equivalence relation between the two propositions, i.e., the truth value of the

two propositions are equivalent. But the relationship is also asymmetric; one proposition is a copy of the other. The copy of one proposition is classified as **"equivalent information"** Furthermore, a proposition can be derived using logical deduction. Hence, the truth value of the derived proposition depends on the truth values of its antecedent propositions. This type of derived proposition is classified as **"derived information"**.

Returning to the example, determining the provenance of the premise requires that we link, in some way, the premise to the proposition in the other web page from which it is copied. That link will also require some type of certification so that we know who created it and whether it is to be trusted. The same is true of the conclusion. Minimally, we should link it to its premise, maximally we should link it to the axioms that justify its derivation. This link would also need to be certified in a similar manner.

In practice, a proposition may be derived by applying different axioms. For example, according to the demerit point system of Ontario's Ministry of Transportation, a person may get 3 points for the following reasons: Failing to yield the right-of-way; Failing to obey a stop sign, traffic light or railway crossing signal; Going the wrong way on a one-way road. Each may be a possible reason for a loss of points.

Derived propositions may also be dependent upon disjunctions, conjunctions and/or negations of other propositions.

From these two cases, a number of concepts required for reasoning about provenance emerge:

Text is divided into propositions. Once so designated, they are assumed to be indivisible.

An asserted proposition must have a digital signature.

If the assertion is to be believed, then the person or organization that signed the assertion must be acceptable to the user of the information.

As propositions are reused across the web, a link between where it is used and where it came from must be maintained. These links, or dependencies, must also be signed.

Dependencies can be simple copies, or can be the result of a reasoning process. If the latter, then axioms used in the reasoning should also be identified and signed by an acceptable organization.

Finally, throughout the above points, the notion of acceptable signing authorities is basic to the analysis of provenance. Consequently, Knowledge Provenance is context sensitive, where the context is defined by a set of signing authorities acceptable to the person requesting provenance.

4 Static Knowledge Provenance Ontology

In order to give a formal and explicit specification for Static KP and to make it available on the web, a Static KP ontology is defined in this section. Following the ontology development methodology of Gruninger & Fox [1995], we specify Static KP ontology in 4 steps: (i) provide a motivating scenario; (ii) define informal competency questions for which the ontology must be able to derive answers; (iii) define the terminology (i.e., predicates); (iv) define the axioms (i.e., semantics). We already discussed motivating scenarios in the earlier section. This section presents informal competency questions, terminology, and axioms.

4.1 Informal Competency Questions

Competency questions define the scope of an ontology. In other words, assuming some type of deductive reasoning system, an application built using the ontology must be able to deduce answers to the competency questions. The key concepts the ontology has to support regarding static validity are: what is the truth value, what does it depend on, who created it, and do we trust them? The following questions define the competence of the Static KP's ontology.

Is this proposition true, false, or unknown?

Who created this proposition?

Does the truth of this proposition depend on any other propositions? If so, what?

If this proposition is to believed, who must I trust?

What is the digital signature verification status of the proposition? Of the dependency?

Which knowledge fields does this proposition belong to?
In these fields, can the information creator be trusted?

4.2 Terminology

There are five main classes in the ontology: Propositions (KP-Props), Documents, Information Sources (InfoSource), Trust Relations (TrustRelation) and Signature Status (SigStatus). Figure 1 depicts the taxonomy of concepts in the Static KP Ontology.

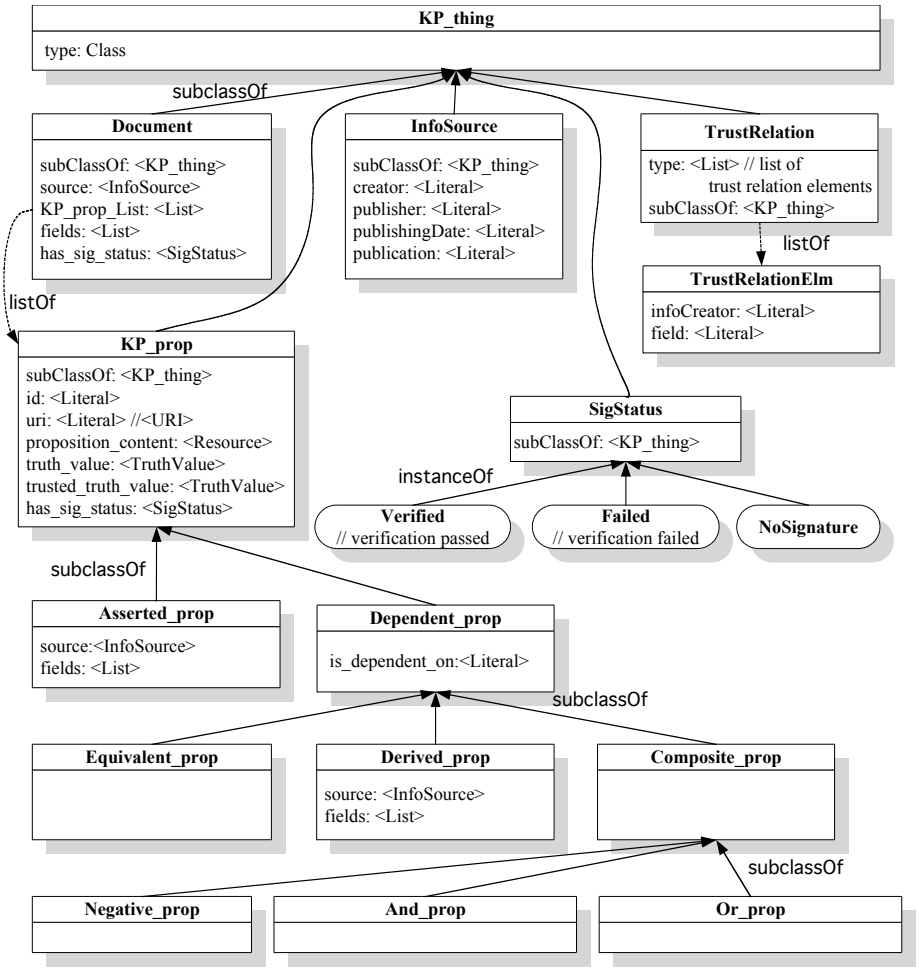

Figure 1. Major KP1 Class Definitions and Taxonomy

Propositions

KP-Prop is the most general concept used to represent propositions in a document. The following table defines the predicates for depicting a KP proposition and its attributes:

Predicate	Description
type(x, KP-prop):	x is defined to be a proposition, signified by being of type KP-prop.
proposition_content(x,s):	*s* is the content of the proposition *x*. In html files, the content of a proposition usually is a string; in xml files, the content of a proposition can be a xml element.
truth_value(x,v):	Proposition *x* has a truth value *v* specified by proposition creator. *v* may be one of "True" or "False".
trusted_truth_value(a,x,v)	Agent a trusts that proposition x has a truth value v. *v* may be one of "True"

	"False", or “Unknown”.
type(x, asserted_prop):	x is an assertion and not dependent upon any other proposition. Its truth value is defined by the truth_value predicate.
type(x, dependent_prop):	x is a proposition whose truth value is dependent upon another proposition. Dependent-prop class is further divided into 3 subclasses: equivalent-prop, derived-prop, and composite-prop.
type(x, "equivalent_prop"):	An equivalent-prop is a copy of and its truth value is the same as the proposition it depends on.
type(x, "composite_prop"):	Composite-prop’s is defined to be the logical combination of its constituent propositions. A composite-prop is divided into 3 subclasses: negative-prop, and-prop, and or-prop.
type(x, "derived_prop"):	A derived-prop indicates that the proposition’s truth value is entailed by the proposition linked to it by its is_dependent_on attribute. For example, derived-prop B has dependency-link pointing to composite-prop A, which means that A entails B, i.e., A -> B
is_dependent_on(x, y)	Proposition x is dependent on proposition y.

Documents

To facilitate the determination of the provenance of a proposition, properties of the document in which it appears may need to be considered. For example, knowing who created the document may be important in determining the validity of a proposition within. A document can be any type of file. For the purposes of this paper, we restrict our attention to standard web files such as: html files, xml files, and xhtml files. Following are document related KP predicates:

Predicate	Definition
type(x, "Document"):	x is defined to be a KP document.
in_document(y,d):	Proposition y is contained in document d.

Information Source and Signature

For any document and proposition its creator can be defined. Along with it can be defined a digital signature and the verification status of the signature. Assume that digital signature validation software provides the result of signature verification.

Predicate	Description
has_infoCreator(x,c):	KP-prop or Document x has infoCreator c. Here, infoCreator may be either creator or publisher.
has_signature(x, s):	The proposition or document x has a signature s, where s is the result of a signature process not defined here.
has_sig_status(x, v):	The digital signature verification status of x is v, where v may be one of three status: "Verified"--- the signature is verified successfully; "Failed"--- the signature verification is failed; and "NoSignature"--- do not have digital signature.
in_document(x,d):	Proposition x is contained in document d.

Trust Relations

In section 3 we stated that Knowledge Provenance is context sensitive, where the context is defined by a set of signing authorities acceptable to the person requesting provenance information. Provenance is dependent on who the requester trusts. Trust in Knowledge Provenance is defined as a set of triples {(a, y, z)} where the information receiver a "trusts" information creator y in a topic or a specific knowledge field z, here, "trust" means that x believes any proposition created by y in the field z to be true. The following defines the trust related predicates:

Predicate	Description
trusted_in(a, c, f):	Provenance requester a trusts information creator c in knowledge field f.
trusted(x, a):	Proposition *x* is trusted by agent a. That means its information creator is trusted by a in one of the fields which proposition x belongs to.
in_fields(x,f):	Proposition *x* is a member of knowledge field *f*.
subfieldOf(x,y):	Knowledge field *x* is a sub-field of knowledge field *y*

4.3 Axioms

In the following, a set of axioms is defined to specify truth conditions of KP-props. Basically, the truth value of an asserted proposition depends on if the proposition is "trusted"; the truth value of an equivalent proposition depends on the truth value of its dependency KP-prop that the equivalent proposition points to by its dependency-link; the truth value of a derived proposition depends on if the proposition is "trusted" and if its dependency KP-prop is true. In addition, a KP-prop is "trusted", if the creator or publisher of the proposition is trusted in one of the fields of the proposition, and the digital signature verification status is "Verified".

Asserted Propositions

An asserted-prop has its truth value as specified, if the asserted-prop is trusted by the agent making the provenance request.

Axiom 1:

for-all (a,x,v) ((type(x, "asserted_prop") ^ trusted(x, a) ^ truth_value(x, v))
->trusted_truth_value(a, x, v)).

A KP-prop is "trusted", if the creator or publisher of the proposition is "trusted" in one of the fields of the proposition, and the digital signature verification status is "Verified".

Axiom 2:

for-all (a,x,fl,z,c,w)
((type(x, "KP-prop") ^ has_sig_status(x, "Verified") ^ has_infoCreator(x, c)
^ in_fields(x, fl) ^ contained_in(z, fl) ^ trusted_in(a, c, w) ^ subfield_of(z, w))
->trusted(x, a)).

For an asserted or derived KP-prop that has no creator specified, the creator of the document is the default creator of the KP-prop.

Axiom 3:

for-all (x, d, c)
(((type(x, "asserted-prop") or type(x, "derived-prop")
or type(x, "equivalent_prop"))
^ (not(exist (c2) has_creator(x, c2)))
^ in_document(x, d) ^ has_creator(d, c))
-> has_creator(x, c)).

If a proposition does not have a creator, then the digital signature verification status of the KP-prop is determined by the digital signature verification status of the document.

Axiom 4:

for-all (x, d, c, v)

(((type(x, "asserted-prop") or type(x, "derived-prop")
or type(x, "equivalent_prop"))
^ (not (exist (c2) has_creator(x, c2)))
^ in_document(x, d) ^ has_creator(d, c) ^ has_sig_status(d, v))
-> has_sig_status(x, v)).

Equivalent Propositions

The trusted truth value of an equivalent-prop is the same as the trusted truth value of the proposition it depends on. Note that we have to trust whomever created the equivalent proposition and the proposition it depends on.

Axiom 5:

for-all (a, x,y,v) ((type(x, "equivalent_prop")
^ trusted(x, a)
^ is_dependent_on(x, y) ^ trusted_truth_value(a, y, v))
->trusted_truth_value(a, x, v)).

Composite Propositions

The trusted truth value of a negative-prop is the negation of the trusted truth value of the KP-prop it is dependent on.

Axiom 6

for-all (a, x,y)
((type(x, "negative_prop")
^ is_dependent_on(x, y) ^ trusted_truth_value(a, y, "True"))
-> trusted_truth_value(a, x, "False")).

Axiom 7:

for-all (a, x,y)
((type(x, "negative_prop")
^ is_dependent_on(x, y) ^ trusted_truth_value(a, y, "False"))
-> trusted_truth_value(a, x, "True")).

Axiom 8:

for-all (a, x, y)
((type(x, "negative_prop")
^ is_dependent_on(x, y) ^ trusted_truth_value(a, y, "Unknown"))
-> truth_value(a, x, "Unknown")).

The truth value of an And-prop is "True" if all its dependency KP-props are "True"; The truth value of an And-prop is "False" if at least one of its dependency KP-props is "False"; and the truth value of an And-prop is "Unknown" if at least one of its dependency KP-props is "Unknown" and none of them is "False".

Axiom 9:

for-all(a, x)
((type(x, "and_prop")
^ for-all (y) (is_dependent_on(x, y) -> trusted_truth_value(a, y, "True")))
->trusted_truth_value(a, x, "True")).

Axiom 10:

for-all(a, x)
((type(x, "and_prop")
^(exist(y) (is_dependent_on(x, y) ^ trusted_truth_value(a, n, "False"))))
->trusted_truth_value(a, x, "False")).

Axiom 11:
for-all(a, x,)
((type(x, "and_prop")
^ (exist(y) (is_dependent_on(x, y) ^ ^ trusted_truth_value(a, y, "Unknown")))
->trusted_truth_value(a, x, "Unknown")).

The truth value of an Or-prop is "True" if at least one of its dependency KP-props is "True"; The truth value of an Or-prop is "False" if all its dependency KP-props are "False"; and the truth value of an Or-prop is "Unknown" if at least one of its dependency KP-props is "Unknown" and none of them is "True".

Axiom 12:
for-all(a, x)
((type(x, "or_prop")
^ (exist (y) (is_dependent_on(x, y) ^ trusted_truth_value(a, y, "True"))))
->trusted_truth_value(a, x, "True")).

Axiom 13:
for-all(a, x)
((type(x, "or_prop")
^ (for-all (y) (is_dependent_on(x, y) ^ trusted_truth_value(a, y, "False"))))
->trusted_truth_value(a, x, "False")).

Axiom 14:
for-all(a, x)
((type(x, "or_prop")
^ (not ((exist y) (is_dependent_on(x, y) ^ trusted_truth_value(a, y, "True"))))
^ ((exist y) (is_dependent_on(x, y) ^ trusted_truth_value(a, y, "Unknown"))))
->trusted_truth_value(a, x, "Unknown")).

Derived Propositions

The truth value of a derived proposition is "True" or "False" as specified, if it is "trusted" and its dependency KP-prop (condition) is "True". Note that the axiom used to derive the truth value does not have to be included as part of the dependency.

Axiom 15:
for-all (a, x, y, v)
((type(x, "derived_prop")
^ trusted(x, a) ^ truth_value(x, v)
^ is_dependent_on(x, y) ^ trusted_truth_value(a, y, "True"))
->trusted_truth_value(a, x, v)).

5. RDFS Implementation and Example

This section illustrates how to embed Knowledge Provenance information into web documents and how the axioms are used to determine the provenance of propositions. We define KP metadata using RDFS, but rather than maintain provenance in separate "meta" documents, KP metadata is embedded directly in web document containing the propositions, making it easier to read and maintain. In the following example, the content that is marked with a pair of tags <kp:proposition> … </kp:proposition> is one proposition in a document.

Document1: http://www.example.com/polar_bears030109.html
<HTML xmlns="http://www.w3.org/1999/xhtml"

```
        dsig =  "http://www.w3.org/2000/09/xmldsig#"
        kp = "http://www.eil.utoronto.ca/kp#"
        xml:lang="en" lang="en">
<HEAD>
  <kp:Document rdf:about=" http://www.example.com/polar_bears030109.html"/>
</HEAD>
<BODY>

  <kp:derived_prop rdf:id="EndangeredPolarBears">
    <kp:proposition_content> the extinction of polar bears within 100 years
        </kp:proposition_content>
    <kp:truth_value>"True"</kp:truth_value>
    <kp:is_dependent_on>"MeltingArcticSeaIce"</kp:is_dependent_on>
    <kp:infoSource>
      <kp:creator>"Andrew Derocher, Scientist in U. of Alberta"</kp:creator>
    </kp:infosource>
    <kp:in_fields>"Polar Bears"</kp:in_fields>
    </rdf:List>
  </kp:Derived_prop>

  <kp:equivalent_prop rdf:id="MeltingArcticSeaIce">
    <kp:proposition_content>
      The accelerated rate of reduction of perennial sea ice in the Arctic
    </kp:proposition_content>
     <kp:infoSource>
      <kp:creator>"Andrew Derocher, Scientist in U. of Alberta"</kp:creator>
     </kp:infosource>
    <kp:is_dependent_on>
      http://www.unep.org/.../rpt2002.html#MeltingArcticSeaIce
    </kp:is_dependent_on>
  </kp:equivalent_prop>

   <Signature ID="Derocher--polarBears">
     <SignedInfo>
        <CanonicalizationMethod
  Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
       <SignatureMethod
  Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
       <Reference URI="#EndangeredPolarBears">
         <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
         <DigestValue>j6hm43k9j3u5903h4775si83...=</DigestValue>
       </Reference>
       <Reference URI="#MeltingArcticSeaIce">
         <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
         <DigestValue>g79lk20rjf023rr032kr93kjr...=</DigestValue>
       </Reference>
     </SignedInfo>
     <SignatureValue>M459ng9784t...</SignatureValue>
     <KeyInfo>
       <X509Data>
         <X509SubjectName>...</X509SubjectName>
         <X509Certificate>MIID5jCCA0+gA...lVN</X509Certificate>
       </X509Data>
     <KeyInfo>
```

```
    </Signature>

</BODY>
</HTML>
```

In this sample document, there are two KP-props: (1) a “derived-prop” with id "EndangeredPolarBears", entailed by KP-prop "MeltingArcticSeaIce", (2) an “Equivalent-prop” with id "MeltingArcticSeaIce". The trusted truth value of the drived proposition is determined using axiom 15. Let’s assume that the digital signature (in XML-Signature syntax) of these two propositions is verified successfully, and Andrew Derocher is trusted in the field of Polar Bears by the person requesting knowledge provenance. Therefore, according to Axiom 2, the derived-prop "EndangeredPolarBears" will be trusted. But to determine whether its truth value is to be trusted, we have to determine whether the proposition it depends on has a trusted_truth_value of true. The proposition "MeltingArcticSeaIce" is an equivalent proposition, its truth value, as defined by axiom 5, depends on "http://www.unep.org/.../rpt2002.html#MeltingArcticSeaIce".

Document2: http://www.unep.org/.../rpt2002.html

```
<HTML xmlns="http://www.w3.org/1999/xhtml"
          dsig =  "http://www.w3.org/2000/09/xmldsig#"
          kp = "http://www.eil.utoronto.ca/kp#"
          xml:lang="en" lang="en">
<HEAD>
   <kp:document rdf:about="http://www.unep.org/.../rpt2002.html">
      <kp:infoSource>
          <kp:creator>"UNEP Arctic Monitoring Team"</kp:creator>
          <kp:publishingDate>"Jan. 3, 2003"</kp:publishingDate>
      </kp:infoSource>
   </kp:Document>
</HEAD>
<BODY>
   <kp:asserted_prop rdf:id="MeltingArcticSeaIce">
     <kp:proposition_content>
      In 2002, a satellite-based survey found Arctic sea ice coverage fell from around 6.5 million square kilometres to around 5.5 million square kilometres in one year.
      </kp:proposition_content>
      <kp:truth_value>"True"</kp:truth_value>
      <kp:in_fields>"Arctic Environment"</kp:in_fields>
   </kp:asserted_prop>

     <Signature ID="unep--meltingArctic">
       <SignedInfo>
          <CanonicalizationMethod
    Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
         <SignatureMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
         <Reference URI="http://www.unep.org/.../rpt2002.html">
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>f44k5ky375...=</DigestValue>
         </Reference>
       </SignedInfo>
       <SignatureValue>M5h9f4...</SignatureValue>
       <KeyInfo>
         <X509Data>
           <X509SubjectName>...</X509SubjectName>
           <X509Certificate>f89r3...k4u</X509Certificate>
```

```
        </X509Data>
      <KeyInfo>
    </Signature>
</BODY>
</HTML>
```

In document 2, "MeltingArcticSeaIce is an asserted-prop. Its truth value, as defined by axiom 1, depends on whether it is trusted. The creator of the document is "UNEP Arctic Monitoring Team" (United Nations Environment Program). Assume the digital signature is verified successfully, and the requestor trusts UNEP in the field of "Arctic Environment". Then, MeltingArcticSeaIce has a trusted_truth_value of true. Consequently, the equivalent proposition "MeltingArcticSeaIce" in document 1 also has a trusted truth value of true, and finally, the derived proposition "EndangeredPolarBears" is has a trusted truth value of true.

6. Implementation

The Static Knowledge Provenance model has been implemented in RDFS-Prolog. The system reasons about RDFS data. In the system, all RDFS data are represented as triples in the form of rdf_triple(S, P, O) where S denotes “Subject”, P denotes “Predicate”, and O denotes “Object”. Furthermore, the semantics of RDFS and axioms of Static KP are represented with Prolog rules. In this way, the system can infer the truth of any KP-prop.

A Static Knowledge Provenance analyzer (based on earlier version of KP1 model) has been implemented in JAVA as a service available over the web at http://www.eil.utoronto.ca/kp1/ (see figure 2). Given a URL, the analyzer extracts KP-props and their descriptions, and follows paths through the web to accumulate provenance information. The KP analysis result is then displayed in the web browser. Test data files (html file with kp tags) can be found at: (Note: to run the system, Java 1.4.1 is required.)

http://www.eil.utoronto.ca/kp1/data1.html
http://www.eil.utoronto.ca/kp1/stock0.html
http://www.eil.utoronto.ca/kp1/sw.html

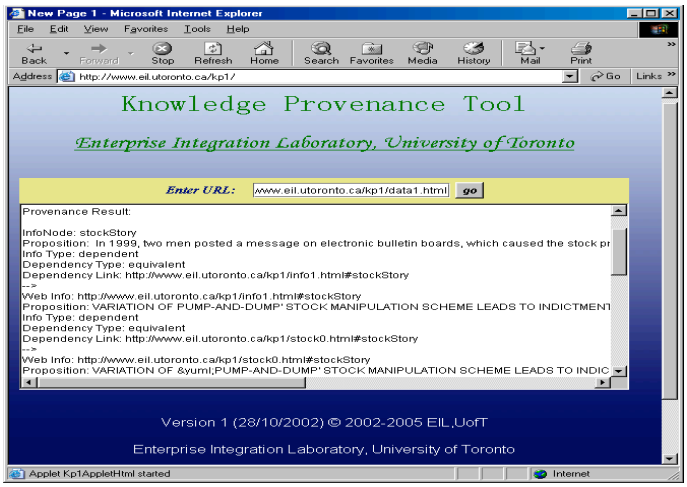

Figure 2. Web-based KP System

7. Summary and Future Work

In this paper, the problem of how to determine the validity of information/knowledge on the web is addressed. Knowledge provenance is proposed to address the problem by modeling and maintaining the evolution and validity of knowledge. We introduced 4 levels of Provenance:

12 Figure 2. Web-based KP System

Level 1: Static KP model is designed and formally specified as an ontology, and a semantic web implementation is provided for static knowledge provenance.

Level 2: Dynamic KP, extends Level 1 to deal with dynamic changes of truth over time. This level incorporates notions of time intervals, non-monotonic reasoning and version management.

Level 3: Uncertain KP, extends Level 2 to account for uncertainty in assertions, inferences, dependencies, etc. Notions of combining and propagating uncertainty are incorporated at this level.

Level 4: Judgment-Based KP, extends Level 3 by providing a process to infer the truth value of knowledge based on human judgment. The goal is to develop a formal "social" process incorporating Certification, Recommendation, Review, Authority, etc. [Alexander & Tate 99]

In this paper, we have focused on Level 1: Static Knowledge Provenance. Static KP introduces concepts and standards that are fundamental to all four levels, namely: truth, traceability and trust. We have provided a level 1 ontology, its specification in RDFS and demonstrated its use within a web browser.

As stated earlier in the paper: "we believe the web will always be a morass of uncertain and incomplete information". The challenge therefore is to create models and processes that will enable the validation of as much information as possible.

References

1. Alexander, J. E., and Tate, M.A., (1999), “Web Wisdom: how to evaluate and create information quality on the web”, Lawrence Erlbaum Associates Publishers.
2. Bearman, D., and Trant, J., (1998), “Authenticity of Digital Resources”, *D-Lib Magazine*, June Issue. http://www.dlib.org/dlib/june98/06bearman.html
3. Bhatnager, R.K., and Kanal, R., (1986), “Handling Uncertain Information: A Review of Numeric and Non-numeric Methods”, in *Uncertainty in Artificial Intelligence*, edited by L. Kanal and J. F. Lemmer, Elsevier Science Publishers.
4. Berners-Lee, T., Semamtic Web Road Map, http://www.w3.org/DesignIssues/Semantic.html
5. Berners-Lee, T., Hendler, J., and Lassila, O., (2001), "The Semantic Web", Scientific American, May 2001.
6. de Kleer, J., Forbus, K., McAllester, D., (1989), “Truth Maintenance Systems (Tutorial SA5)”, IJCAI-89, SA5-182~225.
7. Gruninger, M., and Fox, M.S., (1995), "Methodology for the Design and Evaluation of Ontologies", Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal.
8. Khare, R., andRifkin, A., (1997), "Weaving and Web of Trust", World Wide Web Journal, Vol. 2, No. 3, pp. 77-112.
9. Mayorkas, A. N., http://www.usdoj.gov/usao/cac/pr/pr2000/003.htm
10. Kaijun T., (2002), “Building Your Appropriate Certificate-based Trust Mechanism for Secure Communications”, White Paper, March, 2002. http://www.rainbow.com/library/8/BuildingYourAppCert_based.pdf
11. Lassila, O., Swick, R.R., et al., (1999), “Resource Description Framework (RDF) Model and Syntax Specification”, W3C Recommendation 22 February 1999. http://www.w3.org/RDF/
12. W3C XML Signature, http://www.w3.org/Signature/