Knowledge Provenance Infrastructure

Paulo Pinheiro da Silva   Deborah L. McGuinness   Rob McCool
Knowledge Systems Laboratory,
Stanford University, Stanford CA 94305
{pp,dlm,robm}@ksl.stanford.edu

Abstract

The web lacks support for explaining information provenance. When web applications return answers, many users do not know what information sources were used, when they were updated, how reliable the source was, or what information was looked up versus derived. Support for information provenance is expected to be a harder problem in the Semantic Web where more answers result from some manipulation of information (instead of simple retrieval of information). Manipulation includes, among other things, retrieving, matching, aggregating, filtering, and deriving information possibly from multiple sources. This article defines a broad notion of information provenance called knowledge provenance that includes proof-like information on how a question answering system arrived at its answer(s). The article also describes an approach for a knowledge provenance infrastructure supporting the extraction, maintenance and usage of knowledge provenance related to answers of web applications and services.

1 Introduction

People who have become effective at using information obtained from the web have become proficient at investigating and evaluating sources of information. If a person believes that, for example, the CNN television station or the New York Times newspaper are reliable sources, then she may be willing to believe the information those organizations publish on the web. If the reputation of the information source is unknown, the person may want more information about the source that may reveal biases, agendas, affiliations, etc. before believing the information. If the information about the source is unavailable or the information source itself is unknown, the person may have a reason to disbelieve the data.

When the user of information is an agent, the task of source investigation and evaluation can not rely on common-sense knowledge such as the reputations of CNN or the New York Times. The agent must have access to the source of information *and* it must have some information about the source in order to be able to evaluate its credibility as a publisher of web information.

If users are going to trust answers obtained from the Semantic Web, then users (humans and agents) need access to knowledge provenance. In order for the Semantic Web to provide knowledge provenance, it needs underlying standards and tools for storing, maintaining and using knowledge provenance. Moreover, this infrastructure must be comprehensive enough to allow the tracking of knowledge provenance from answers to their sources. Therefore, such infrastructure should deal with at least the following kinds of systems:

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

- Information extraction tools used for building knowledge sources such as ontologies, databases, knowledge bases, taxonomies and thesauri. These tools should be able to register meta information about the sources of assertions in knowledge sources.
- Search engines used for producing answers from web documents. These tools should be able to register and present meta information about retrieved documents.
- Inference engines used for deriving answers from knowledge sources. These tools should be able to register meta information about the knowledge sources used in the process of deriving answers. Also, they should be able to dump their proof traces in a sharable, portable format that can be used to explain how answers are derived.
- Web applications and services using answers derived by inference engines and documents retrieved by search engines. These systems should be able to present provenance information in response to user requests. These systems need to understand knowledge provenance registered by other systems.

This article describes a knowledge provenance infrastructure that facilitates the integration of web applications requiring provenance information. Systems can use the infrastructure for keeping knowledge provenance during the process of extracting knowledge and building knowledge sources. Also, knowledge provenance is considered at a level of granularity appropriate for assertions within knowledge sources. Our previous work on knowledge provenance [8, 9] describes a more conservative approach where knowledge provenance is aimed at the level of granularity appropriate for knowledge sources as a whole.

Moreover, the infrastructure supports Semantic Web functionalities beyond the identification of sources of answers. Our infrastructure provides a specification of a portable proof that can be used to capture information manipulation descriptions. This information is then used by the Inference Web browser to display interactive proof displays for debugging and abstractions of the proofs into more understandable explanations for end users. This information may also be used by truth maintenance systems as well as hybrid reasoning environments.

2 Knowledge Provenance

Knowledge provenance includes *source meta-information*, which is a description of the origin of a piece of knowledge, and *knowledge process information*, which is a description of the reasoning process used to generate the answer. We have used the phrase knowledge provenance instead of data provenance intentionally. Data provenance [1, 3] may be viewed as the analog to knowledge provenance aimed at the database community. That community's definition typically includes both a description of the origin of the information and the process by which it arrived in the database. Knowledge provenance is essentially the same except that it includes proof-like information about the process by which knowledge arrives in the knowledge base. This process may include extensive reasoning used to generate deductive closure information. In this sense, knowledge provenance broadens the notion of data derivation that can be performed before data is inserted into a database or after data is retrieved from a database. Nevertheless, data provenance and knowledge provenance have the same concerns and motivations.

The use of reasoning is not a requirement for using a knowledge provenance infrastructure. For instance, many components of the Inference Web [8] such as the IWBase (a registry containing information about objects useful for proofs and explanations) and portable proofs (a proof interlingua) are used in the ARDA Aquaint project[1], which has a main thrust of question answering using information retrieval techniques. Also, the infrastructure can be used to provide simple source justification for answers that are simply retrieved or for answers

[1] http://www.ic-arda.org/InfoExploit/aquaint/

that have been obtained using complex reasoning and, more typically, it can be used when the answers are derived using a combination of both. A typical scenario includes using knowledge sources where information is available in a format appropriate for machine processing e.g., RDF [6], DAML+OIL [2], OWL [10], etc. If a knowledge base was built using a particular source, for example CNN, then Inference Web would store CNN as the original source of the knowledge. Additional information may be stored about knowledge sources such as the source's authoritativeness, URL, contributors, date of input and update, etc. If some of the information in a knowledge base is from another source, for example the AP news wire, then Inference Web may be used to store that certain assertions came from another source. This information may be attributed at the knowledge base level or at the assertion level.

3 The Stanford Knowledge Provenance Infrastructure

The Stanford Knowledge Provenance Infrastructure (KPI) is an integration approach for TAP [4] with Inference Web's IWBase [9] and Inference Web's portable proofs [12]. TAP is a tool for extracting information and building knowledge sources. It can store provenance information at the level of assertions. IWBase provides infrastructure for provenance originally aimed at the granularity of knowledge bases. In KPI, IWBase will support provenance at the assertion level. Portable proofs support explanations of inferred information along with provenance-based explanations of retrieved answers. These systems are in use for funded projects and are supporting different levels of knowledge provenance. Currently, TAP is not integrated with IWBase and portable proofs. This paper describes the integration route underway and shows how putting the two systems together provides much more than TAP or the Inference Web can do alone. The integrated system provides a scalable solution to knowledge provenance that is aimed at the needs of knowledge bases generated as the result of automated programs (e.g., wrapper and extraction software, e.g., Fetch [5]) as well as knowledge bases generated by humans using tools such as OWL ontology editors (e.g., Protégé [11]).

3.1 IWBase: Infrastructure for Meta-Information Annotation

IWBase [9] (formerly known as the IW Registry) is an interconnection of distributed repositories of meta-information relevant to proofs and explanations, including knowledge provenance information. Every entry in these repositories is an instance of an IWBase concept. For example, *Knowledge Source* is an IWBase concept that is useful for entries such as ontologies, knowledge bases, thesauri, etc. The knowledge source entry for an ontology describes stores of assertions about the ontology such as its original creator(s), date of creation, data of last update, version, URL (for browsing), description in English, etc. IWBase's provenance information is expanding on an as-needed basis driven by application demands.

Every entry has an URI and is stored both as a file written in DAML+OIL/OWL and as a set of tuples in a database. IWBase files are mainly used by portable proofs (see Section 3.3) to annotate their content. IWBase databases are mainly used for evolving meta-information and for supporting web services querying the IWBase.

IWBase can keep provenance at the level of documents. For example, consider the case where the data came from "Joe's Tom Hanks Fan Information Collection", and Joe does not make information available about the source of each piece of data. In KPI, IWBase will also be able to keep provenance at the assertion level. For instance, suppose we have an entry about "Tom Hanks", who is an "Actor". This entry may come from a RDF document where one triple says that there is an entity with a `rdfs:label` of "Tom Hanks", and another triple says that his `rdf:type` is Actor. Some IWBase users, however, simply do not want to keep provenance information at the triple level. In this case, IWBase can store the fact that all of the elements in that particular file or knowledge base, i.e., triples, came from Joe. Thus, when users ask where the particular elements come from, an application using the IWBase can dynamically attach the provenance to the elements and return it. The inclusion of provenance is possible whether provenance is kept at the level of documents, assertions or both documents and assertions.

3.2 TAP: Infrastructure for Knowledge Source Construction

TAP is a system for knitting together data fragments from disparate XML/SOAP based web services, and from disparate HTML based web sites, into a single schematically unified global knowledge base. The TAP system consists of a number of data servers which communicate with each other and with applications using an XML/SOAP based protocol we call *GetData*. In this case, sites wanting to provide data to the Semantic Web about a particular subject must first agree on the schema for describing that subject. New sites may be added to the system without any modifications to the applications that use the data.

The TAP system has two types of knowledge source tracking. The first method is implicit: the TAP server knows it is interacting, for example, with CNN, therefore it knows that any data it receives in this session is from CNN. The second method is explicit: if the data being retrieved was somehow manipulated, then the TAP object identifiers for each entity being discussed can be examined to discover the page from which they originated. A small match code appended to the page URL also indicates the approximate region of the page from which the entity was found. In either case, the TAP system includes information about each site, and a way to query which site a particular source page comes from. Knowledge provenance information extracted by TAP can be added into the IWBase in an automatic way. Thus, users can see the knowledge provenance later when asking for the sources of answers derived from TAP generated knowledge sources.

TAP provides a bootstrapping system which uses HTML parsers to transform web sites intended for humans into Semantic Web sites intended for agents. To date, TAP has scanned and aggregated data from over 110,000 URLs spread across 35 sites into a knowledge base consisting of over 860,000 logical sentences about over 500,000 individuals. Individual types include corporations, nations, politicians, locations, celebrities, movies, music albums, weapons systems, and many others.

3.3 Portable Proofs: Infrastructure for Answer Derivation Representation

The portable proof specification[2] is a DAML+OIL (migrating to OWL) representation of proofs produced by reasoners during the process of deriving answers. There, a *node* in a deduction tree is labeled by one formula and one inference rule used to conclude the labeling formula. Labeling formulas are formula occurrences. Conceptually one can think of a node in a deduction tree as a representation of one step in a deductive information manipulation process. It is the result of a single rule application applied to some previous information deriving a single formula. A *node set* is a set of one or more nodes where all the nodes are labeled by the same formula. Conceptually one can think of a node set as a set of applications of inference rules used to derive the identical formula in a single step. Node sets capture information concerning *all* of the ways one or more question answering systems came to believe a single statement. A node captures *one* way one or more question answering systems came to believe a single statement. Node sets are a critical building block of the Inference Web since they are the key to proof combination and multiple explanations.

Node sets are used to support knowledge provenance since they are used to track how conclusions are derived from antecedents. The premises of an inference step are the formulas labeling node sets associated with the inference step as antecedents. An answer is derived by the last inference step in a proof.

Knowledge source information may be stored at the level of an entire knowledge base or at an individual assertion level. Either way, Inference Web can take any particular answer and trace back through the inference steps used, looking at their antecedents and determining all of the sources used to arrive at an answer. This process allows Inference Web to provide a summary collection of all sources used to obtain an answer and also allows it to provide the sources used for any particular statement. The identification of all sources used is an important strategy to determine whether or not we should trust the data. For example, we may not trust

[2] http://www.ksl.stanford.edu/software/IW/spec/iw.daml.

information coming from "Joe's Tom Hanks Collection" but we may trust information coming from "The Rita Wilson Fan Club". Thus, we will probably be more likely to believe the data if it is associated with both sources rather than just the "Joe's Tom Hanks Collection". Moreover, suppose that we know that "Joe's Tom Hanks Collection" is known for publishing unreliable information. Then we may be inclined to disbelieve the data even if it is also associated with "The Rita Wilson Fan Club".

Portable proofs may also be used to tackle some problems related to knowledge provenance redundancy. In the simple case, if everything in one knowledge base came from one source, a single statement may be used to capture the source of every statement in the knowledge base. If the knowledge base is created as a view or aggregation of the content available from multiple sources, IWBase can be used to store source information at the statement granularity or it can store that the information in this knowledge base used multiple sources and not distinguish which assertions came from which source.

4 Knowledge Provenance Usage

The infrastructure provides support for provenance information whenever it is possible to identify some document or document element to which we can associate provenance information. Also, it provides a systematic way for generating documents that are relevant for the "semantic part" of the web. Three approaches are considered for an application to use provenance information: it incorporates source meta-information into documents; it incorporates knowledge process information into documents; and it interacts with a data server which is performing multi-site aggregation of data and provenance information.

4.1 Incorporating Source Meta-Information

Applications using our infrastructure do not need to store and manipulate data and its corresponding provenance information in any particular format: provenance information is kept separately in the IWBase, and then reassembled upon request. In fact, our approach has been either to avoid transporting provenance data where we can (and use services to access it later), or to simply accept the cost of storing and maintaining provenance information as a necessary one in order to support trustworthiness of data.

When provenance information is needed, it can be added on a per-file basis. Thus, an application can use a KPI service to retrieve provenance information and it can apply its preferred way of incorporating the information including reification, appending new XML elements, or using quads [7]. For example, for RDF, DAML, and OWL files, the application can use the same approach that we use with TAP documents where TAP can ask IWBase for the URI of provenance information of a given piece of information, e.g., a RDF triple, and apply reification.

The identification of a specific piece of information within a document may be a problem for some document formats such as XML. However, we expect that new standards for XML such as XPath will provide a solution for this problem for XML files.

4.2 Incorporating Knowledge Process Information

When an application computes an answer, the Inference Web infrastructure allows it to dump a portable proof format of the computation process. It also allows an interaction mode that would ask the application for a regeneration of the answer with the portable proof support on demand if that interaction style makes more sense. In either case, the agent (or user) through use of Inference Web can peruse the portable proof to find ground facts supporting the derived answer. If the application dumps limited granularity in the proof such as if it used told information (from a particular source) or used told information from a source and then applied complicated reasoning, the end user could at least have access to the sources used and if the application manipulated the information. We encourage granularity in the portable proof dumps that support demand-based explanations giving access to the deduction path but we do not require it. This allows us to interact with question answering systems that can not or do not want to provide details of their information manipulation path but still can provide access to the source of the original information.

It is important to note that a portable proof is a forest of proof trees rather than a single tree. This structure is required so that Inference Web can support infrastructures where multiple question answering systems contribute pieces of an answer and also can support hybrid reasoning environments where query managers may break up questions into components that different agents will answer. This is also used to support situations where the same answer can be obtained from multiple paths. This forest feature is one potential reason why Inference Web and the knowledge provenance work may be well suited (and potentially better suited than a data provenance approach) to supporting explanation of the Semantic Web.

4.3 Querying Provenance Information

Each node of the IWBase is a repository of DAML/OWL files mirrored in a relational database. This means that documents can refer to IWBase entries as typical DAML/OWL documents without needing to know about details of database management systems. It also means that queries are expected to be performed in an effective way over the database. In fact, the metamodel for storing provenance meta-information (see the class diagrams in http://www.ksl.stanford.edu/software/iw/spec) is a typical database schema using conventional indexing techniques. Thus, queries over the structured database are expected to have a better performance than over a RDF file storing all the triples for provenance information.

TAP can generate the RDF triples from any particular site on demand and pass the provenance information to IWBase. The set of source URLs and sites contributed to any aggregated data block can then be recorded on IWBase. Any receiving application can then query the IWBase for the source(s) of any triple.

5 Conclusions

The Semantic Web will need infrastructure for knowledge provenance if users are going to trust answers produced by Semantic Web applications and services. In this article we described an infrastructure that can provide comprehensive answer-to-source knowledge provenance for the Semantic Web. This solution integrates the Inference Web infrastructure for explaining answers from web applications and the TAP system for extraction and semantic search.

We also described how provenance information supported by the infrastructure could be used on demand in association with web documents. The Wine Agent[3], the DAML Query Service[4], and the OWL Query Service[5] are Semantic Web agents supported by the Inference Web that present knowledge provenance at the granularity of knowledge sources. These agents are based on the Stanford's JTP hybrid reasoner that produces portable proofs. Also, in the context of the CALO project[6], we are creating a new agent that provides answers along with knowledge provenance information supported by KPI to handle a distributed, hybrid question answering system using a number of reasoning systems.

We are currently extending SRI's SNARK theorem prover[7] to produce portable proofs and integrating with ISI's query planner as well as pursuing discussions with designers of other reasoning systems including W3C's CWM[8] and UT's KM[9]. We also presented a solution that provides provenance information at a granularity aimed

[3]http://www.ksl.stanford.edu/people/dlm/webont/wineAgent/
[4]http://www.ksl.stanford.edu/projects/owl-ql/
[5]http://www.ksl.stanford.edu/projects/dql/
[6]http://www.ai.sri.com/project/CALO
[7]http://www.ai.sri.com/˜stickel/snark.html
[8]http://www.w3.org/2000/10/swap/doc/cwm.html
[9]http://www.cs.utexas.edu/users/mfkb/km.html

at facts. Our work provides insight into how to obtain, manipulate, and use meta information using the Inference Web and TAP tools to improve trust on the Semantic Web.

Acknowledgment The authors would like to thank Kevin Wilkinson for his many valuable comments during the preparation of this article.

References

[1] Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Why and Where: A Characterization of Data Provenance. In *Proceedings of 8th International Conference on Database Theory*, pages 316–330, January 2001.

[2] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. DAML+OIL (March 2001) Reference Description. Technical Report Note 18, World Wide Web Committee (W3C), December 2001.

[3] Yingwei Cui, Jennifer Widom, and Janet L. Wiener. Tracing the Lineage of View Data in a Warehousing Environment. *ACM Trans. on Database Systems*, 25(2):179–227, June 2000.

[4] Ramanathan V. Guha, Rob McCool, and Eric Miller. Semantic search. In *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)*, pages 700–709, Budapest, Hungary, May 2003. ACM Press.

[5] Fetch Technologies Inc. http://www.fetch.com/products.asp?sub=prod-agentplatform.

[6] Ora Lassila and Ralph Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 22 February 1999.

[7] Robert MacGregor and In-Young Ko. Representing Contextualized Data using Semantic Web Tools. In R. Volz, S. Decker, and I. Cruz, editors, *Proceedings of the First International Workshop on Practical and Scalable Semantic Systems*, Sanibel Island, FL, USA, 2003.

[8] Deborah L. McGuinness and Paulo Pinheiro da Silva. Infrastructure for Web Explanations. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *Proceedings of 2nd International Semantic Web Conference (ISWC2003)*, LNCS-2870, pages 113–129, Sanibel, FL, USA, October 2003. Springer.

[9] Deborah L. McGuinness and Paulo Pinheiro da Silva. Registry-Based Support for Information Integration. In *Proceedings of IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, pages 117–122, Acapulco, Mexico, August 2003.

[10] Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. Technical report, World Wide Web Consortium (W3C), August 2003. Candidate Recommendation.

[11] Natalya F. Noy, Michael Sintek, Stefan Decker, Monica Crubézy, Ray W. Fergerson, and Mark A. Musen. Creating Semantic Web Contents with Protégé-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.

[12] Paulo Pinheiro da Silva and Deborah L. McGuinness. Combinable Proof Fragments for the Web. Technical Report KSL-03-04, Knowledge Systems Laboratory, Stanford University, Stanford, CA, USA, January 2003.