Trusting Answers on the Web

Deborah L. McGuinness and Paulo Pinheiro da Silva

Knowledge Systems Laboratory
Stanford University

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. The Web also lacks support for explaining reasoning paths used to derive or retrieve answers. The Inference Web (IW) aims to take opaque query answers and make the answers more transparent by providing explanations. The explanations include information concerning where answers came from and how they were derived (or retrieved). This chapter describes the IW support for explanations on question answering (QA) environments. A characterization of explanation strategies on QA environment is presented. One usage of the Inference Web infrastructure, and in particular of the browser, explainer, and portable proof specification for supporting multiple explanation strategies in the KSL Wine Agent is described in the chapter. Also, a generic architecture for a QA environment incorporating the Inference Web for explaining answers is presented.

1 Introduction

Inference Web (IW) aims to enable applications that can generate portable and distributed explanations for any of their answers. IW addresses needs that arise with systems performing reasoning and retrieval tasks in heterogeneous environments such as the Web. Users (humans and computer agents) need to decide when to trust answers from varied sources. Our belief is that the key to trust is understanding and explanations of information provenance and derivation history can be used to provide that understanding. In the simplest case, users may retrieve information from individual or multiple sources and they may need provenance information (e.g., source, recency, authoritativeness, etc.) before they may decide to trust an answer. Users may also obtain information from systems that manipulate data and derive information that was implicit rather than explicit. Users may need to inspect the deductive proof trace that was used to derive implicit information before they trust the system answer. Many times proof traces are long and complex so users may also need the proof transformed (or abstracted) into a more understandable explanation of the proof. Users may also obtain information from hybrid and distributed systems and they may need help integrating answers and solutions. As Web usage grows, a broader and more distributed array of information services becomes available for use and the needs for explanations that are portable, sharable, and reusable grows.

In this paper, we present the Inference Web as an answer to the problem of trusting information from question answering (QA) systems on the Web. We introduce the notion of knowledge provenance and show how Inference Web is used to provide provenance information increasing users trust in answers. We describe the Inference Web by use of a running example - the KSL Wine Agent. Through this example, we describe explanation needs in typical web agents. We provide examples of how users can get and use explanations in a QA environment. We describe the main inference web components that support the task including the registry, the explainer and the browser[1] and include a block diagram of the Inference Web architecture as viewed by a user. Finally we present our contributions and future work and point to related work.

2 Question Answering with Explanations

An answer explanation is any augmentation that can improve the understanding of an answer. Answers typically will have multiple explanations, each one possibly exposing different aspects of the answer. In this section we present some possible explanation strategies for answers in QA environments.

[1]http://onto.stanford.edu:8080/iwbrowser/

2.1 Knowledge Provenance for Answers

Knowledge provenance includes *source metadata 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 distinguish between the source metadata information and the knowledge process information because users and applications vary with respect to their needs concerning the two topics. We exploit the distinction between source meta information and knowledge processing in our explanation strategies described below.

We have used the phrase knowledge provenance instead of data provenance intentionally. Data provenance [7, 10] 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 it includes proof-like information about the process by which knowledge arrives in the knowledge base. The typical process by which knowledge arrives in a knowledge base may include extensive reasoning such as complicated theorem proving to generate deductive closure information while the typical process by which data arrives in a database may not include as much extensive reasoning. Thus, there may be different challenges and a different emphasis for knowledge provenance work in comparison to data provenance work.

Our past experience with explanation systems for knowledge representation systems has shown that source metadata information for answers is the most important explanation feature for many users and, in many cases, it meets all or most of their explanation needs. For example, users may need to know the author names, authoritativeness, currency, etc. of sources in order to trust answers depending upon them. When they are using applications that use simple look-up techniques to retrieve answers *or* when they are using reasoning components that they view as totally trustworthy, then they may only need source metadata information in order to trust answers. Even if they are using applications that are only using authoritative sources of information, they still may be interested in the source of the information before believing the answer. On the Web, it is rarely the case that all source information and their surrogates, if any, would be considered authoritative and current [16]. Thus it is typically the case that users will need access to source metadata information in order for them to understand and trust web answers. Furthermore, if users are using QA systems that have manipulated information (by reasoners, extractors, heterogeneous agents, etc.), then it would be expected that users would need some kind of understanding of the manipulations and their rationale before they could trust the answers.

2.2 Explanation Strategies

Information retrieval techniques for the Web, such as document ranking, are widely used in QA environments. The presentation of some source metadata along with the answers as performed by many search engines can be viewed as an explanation strategy. However, source metadata may not be enough for explaining retrieved answers. Users may be interested, for example, in understanding the process of selecting answers. In this case, some description of how the answer selection was made, as described in JAVELIN [24], may be a useful way of presenting knowledge process information.

Standard information retrieval techniques may be augmented for question answering by using some stores of background knowledge. Simple query expansion techniques such as using Verity's Topic Sets may be used to retrieve additional related answers to questions. Terms in the query are expanded by related terms in the topic set and thus additional related answers may be retrieved that would have been missed otherwise. Explanations that show why answers were retrieved, including the terms or phrases that were used in the query expansion provide an additional source of explanations. QA systems that use background ontologies, such as FindUR [18] use equivalent terms or subclass relationships to do query expansion. Explanations that expose the subclass relationships (or equivalent terms used) may be used to support those answers.

As QA environments are supported by additional structured information, such as ontologies and annotated documents, they may be able to provide additional answers or more precise answers for questions typically answered by search engines. Since the Semantic Web [1] provides such structured information (along with some notion of the meaning of the structure) it provides the foundation for improved QA systems. In the case of the Semantic Web, the process of explaining derived answers may become much more complex than explaining retrieved answers. In fact, any particular reasoning path to an answer may have many reasoning steps, each one applying one of the available rules implemented by the inference engine used to derive the answers.

As more information is used to answer queries, source metadata information may become more complex. For example, ontologies and ontology sources are new sources to be considered in the process of gathering and annotating source information. Additionally, it is clear that proofs, which have been used for a long time for explaining answers for expert users of theorem provers, may be too complex for users in the Web. Thus, many types of explanations may need to be offered for users on the Web if users need to understand answers. Also, explanations should initially be as simple as possible and users should have the option of asking follow-up questions concerning portions of the answer that they are interested in knowing more about. Both the initial question and the follow-up questions generated because of a desire for understanding may involve information about the source metadata information or knowledge process information or a combination of both.

3 KSL Wine Agent: A Case Study

Every system that provides answers to users is a potential context where Inference Web may add benefit. We will use the KSL Wine Agent[2] as both motivation and as a pedagogical tool for explaining how Inference Web can be used in practice. The KSL Wine Agent is a prototype implementation of the wine agent described in [27]. It is also an extension of the description logic tutorial [19, 6]. It allows a user to choose a meal course (such as pasta with red sauce) and ask for either a description of the suggested wine to drink (such as a dry, red, medium bodied, moderate-flavored wine, a suggested variety such as zinfandel, or a specific suggestion such as Marietta Zinfandel). In its web setting, it can also connect to a few wine web sites and check for matching wines available for sale on those sites. Before a user chooses to buy the wine suggested, they may also ask for information such as why the agent suggested a red wine (because of the red sauce) or where the information came from (such as the DAML wines ontology) or where the specific wine recommendation came from (the wines ontology or a match from the online web site).

Thus, a user might use a search engine interface or a query language such as DQL[3] for retrieving information such as "zinfandels from Napa Valley" or "wine recommended for serving with a spicy red meat meal" (as exemplified in the KSL wine agent example in the OWL guide document [27]). A user might ask for an explanation of why the particular wines were recommended

[2]http://onto.stanford.edu:8080/wino/
[3]http://www.daml.org/2002/08/dql/

as well as why any particular property of the wine was recommended (like flavor, body, color, etc.). The user may also want information concerning whose recommendations these were (a wine store trying to move its inventory, a wine writer, etc.). In order for this scenario to be operationalized, we need to have the following:

- A way for applications (reasoners, retrieval engines, etc.) to dump justifications for their answers in a format that others can understand. To solve this problem we introduce a portable and sharable proof specification.

- A place for receiving, storing, manipulating, annotating, comparing, and returning meta information used to enrich proofs and proof fragments. To address this requirement, we introduce the Inference Web registry for storing the meta information and the Inference Web registrar web application for handling the registry. This addresses the issues related to data provenance.

- A way to present justifications to the user. Our solution to this has multiple components. First the IW explainer is capable of using rewrite rules (or tactics) to abstract proofs in order to provide more understandable explanations. Additionally, the IW browser is capable of navigating through explanations provided in the portable proof format. It can display explanations in multiple styles and using multiple sentence formats including English and KIF[4]. This addresses the issues related to reasoning, explanations, and presentation.

In the next section we demonstrate the KSL Wine Agent use of the Inference Web for explaining answers.

4 Inference Web Support for Question Answering

In this section we describe how portable proofs along with proof-based tools and proof metadata can support explanations in QA environments. In fact, the portable proof specification is the main component of the Inference Web. The proof specification is written in the web markup language DAML+OIL[5] [9] and proofs dumped in this format become a portion of the Inference Web.

[4]http://logic.stanford.edu/kif/kif.html

[5]An OWL [23] version is coming soon.

First we present a brief description on how engines can dump portable proofs followed by a characterization of some strategies used in the KSL Wine agent for explaining answers. Finally, we describe a possible process for QA environments using the Inference Web for explaining answers.

4.1 Generating Proofs

In order for Inference Web to present a proof or explanation of an answer, the retrieval or deductive engine needs to generate raw material that Inference Web can process. The generation of proofs is a straightforward task assuming that the engine data structures storing proof elements can be identified as IW components. To facilitate the generation of proofs, the Inference Web provides a web service that builds portable proofs from IW components and uploads IW components from portable proofs. A description of this web service is available at http://www.ksl.stanford.edu/software/IW/faq/registering/. QA environments can use the IW for explaining answers once their embedded inference/search engines are ready for generating proofs. A few deductive engines, including Stanford's JTP and SRI's SNARK, are currently producing proofs in portable proof format and the web services using Inference Web use those systems for providing answers.

The details of IW inference engine registration for dumping IW proofs are beyond the scope of this chapter although we encourage interested readers to look at the Inference Web infrastructure description in [22] and the portable proof specification description in [25]. Also, the DAML+OIL proof specification is available at http://www.ksl.stanford.edu/software/IW/spec/iw.daml.

4.2 Explaining Answers

The Inference Web aims to present explanations that help users to identify whether they should trust answers in a limited number of interactions with the IW browser. Depending on their requirements, users can move from one explanation to another.

4.2.1 A Default Explanation

Default initial explanations in the IW are very short and are presented in a simple English format as in Figure 1. In this case, one can imagine a scenario where a user is interested in knowing why the Wine Agent suggested a white wine to go along with the meal that is called "Tonys Specialty". We learn that Tonys Specialty is a crab dish which is also a seafood dish. Later if we asked more follow-up questions, we would discover that the wine ontology suggests white wines with seafood dishes.

The IW browser is used to render all the explanations presented in this chapter. When interacting with the browser, users may select from a number of styles and sentence formats for displaying explanations. Initially, the browser includes the "English", "Proof", and "Dag" styles and the restricted "English" and "KIF" sentence formats[6] for rendering explanations (and their proofs). We also expect that some applications may implement their own displays using the IW API. The browser implements a lens metaphor responsible for rendering a fixed number of levels of inference steps depending on the lens magnitude setting. This is used to limit the information presented and expects interested users to ask for follow-up questions about the portions of explanations they are most interested in.

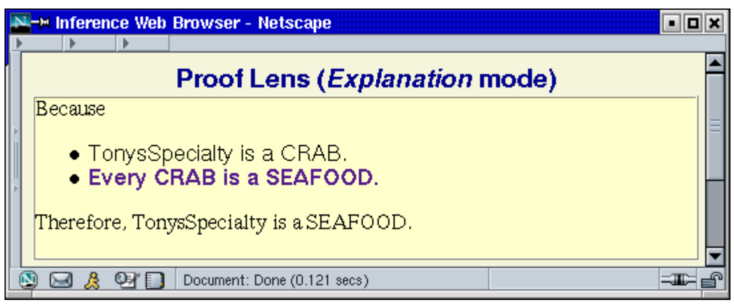

Figure 1: An explanation in English.

The explanation in Figure 1 presents a small part of the knowledge provenance since it shows just the last inference step of what is already a simplification of a proof dumped by an inference engine. We believe that one of the keys to presentation of justifications is breaking proofs into separable pieces.

4.2.2 A Source-Oriented Explanation

If users rely on embedded inference/search engines but do not trust answers, they may ask the browser for the sources metadata information of the answer. Figure 2 presents the source metadata information for the `SEAFOOD` answer (or the answer that `TonysSpecialty` is a SEAFOOD dish). There, the user can learn the three sentences used to derive the answer and that all of the sentences are axioms from the WINE-TEST ontology. Moreover, the user can learn that the KSL team is the author of this particular ontology.

[6]Current investigations are underway for N3 format as well.

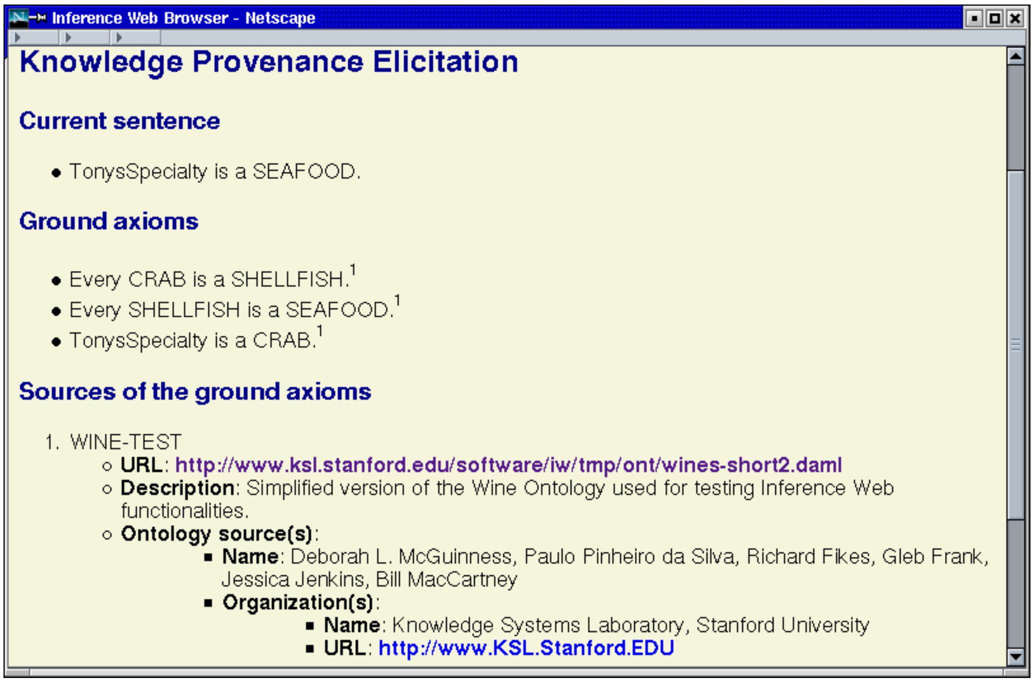

Figure 2: Presenting source metadata information.

In practical scenarios such as those used on the ARDA AQUAINT[7] and NIMD[8] projects, presentation of the source metadata information can be a large list of axioms and sources since a typical scenario can include hundreds of axioms coming from multiple ontologies. Moreover, these ontologies are usually the outcome of an effort of analyzing, summarizing, and compiling information described in multiple documents. Thus, all the original documents and the analysts and subject matter experts involved on the process of compiling these documents also become sources of ontologies.

4.2.3 A Process-Oriented Explanation

A complex explanation may be required if users do not trust the embedded engines. As more hybrid reasoning environments emerge in implemented applications, we expect this kind of explanation to be more important. If these new environments have been built by "plugging" in a number of special pur-

[7] http://www.ic-arda.org/InfoExploit/aquaint/
[8] http://www.ic-arda.org/Novel_Intelligence/

pose reasoners, end users may need to know the inference rules used along with the sentences they were used on. The order of application of these rules may also be of interest. Figure 3 presents an approach to explaining knowledge process information. There, for example, the `Class Transitivity` inference rule was used to derive that `Every CRAB is a SEAFOOD` dish from the premises `Every CRAB is a SHELLFISH` and `Every SHELLFISH is a SEAFOOD`.

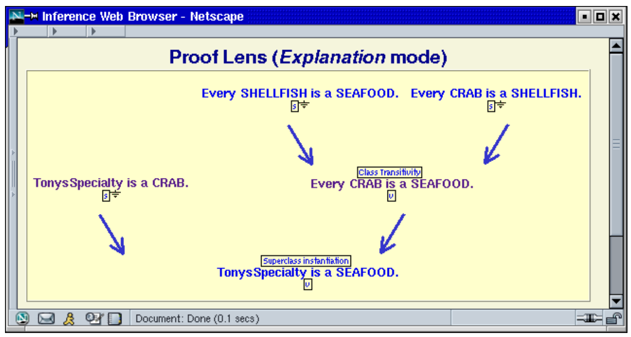

Figure 3: A graphical explanation.

Since the explanations in Figures 1 and 3 are collections of explanation fragments, automatic support for follow-up questions is a critical function of the IW browser. Every element in the viewing lens can trigger a browser action. The selection of an antecedent re-focuses the lens on an antecedent's inference step.Context-appropriate follow-up questions are generated for the user in case they are interested in more information. For other lens elements, associated actions present registry meta-information.

4.3 Inference Web Support

The explanations presented so far are abstractions of actual proofs dumped by engines. In this section we describe how a QA environment can use the IW to generate and display explanations such as the explanations presented in Section 4.2.

4.3.1 Metadata Support

The IW registry is a hierarchical interconnection of repositories of information relevant to explanations and their proofs and is a main component of the process of generating explanations. Entries in the registry contain the information linked to in the proofs. Every entry in the registry is a file

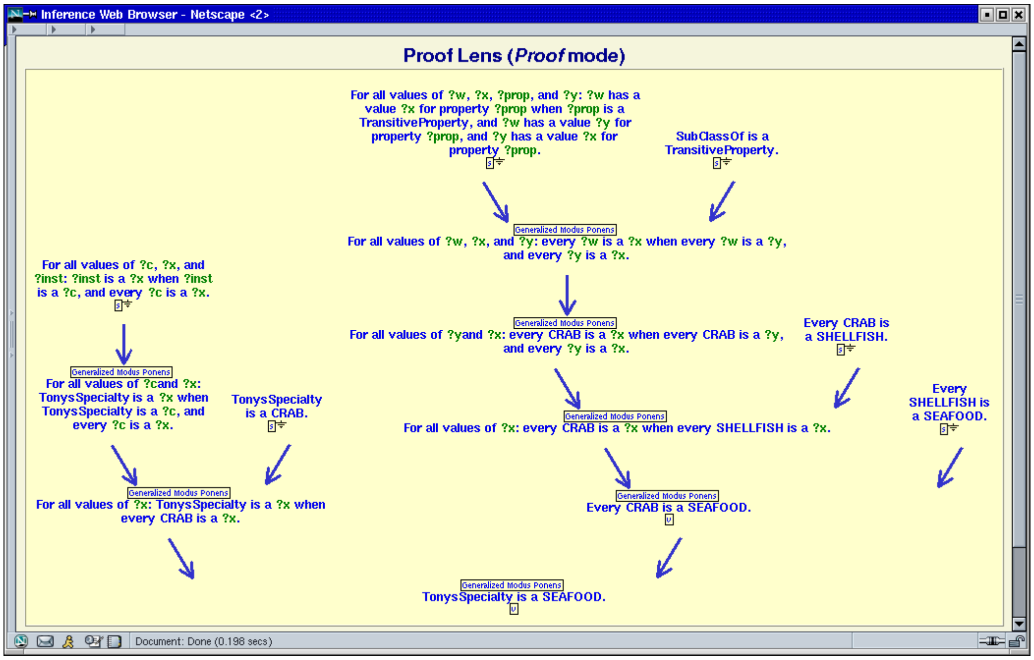

Figure 4: Browsing a proof.

written in DAML+OIL. Also, every entry is an instance of a registry concept. *InferenceEngine*, *Language* and *Source* are the core concepts in the registry. Other concepts in the registry are related to one of these core concepts. For example, *InferenceRule* and *DerivedRule* are concepts related to *InferenceEngine*, *LanguageAxiomSet* and *Axiom* are concepts related to *Language*, and *Ontology*, *Team*, *Person* and *Publication* are concepts related to *Source*. Inference Web tools include a registrar for interacting with the registry. Users must be registered in the Inference Web for maintaining entries in the registry (users can only update their own entries). In the current version of the registrar, any user can browse the registry entries using the registrar at http://onto.stanford.edu:8080/iwregistrar/. Today's implementation of the registry is centralized although we anticipate a distributed registry infrastructure in the future.

4.3.2 Explanation Generation Support

Although essential for automated reasoning, inference rules typically used by theorem provers are often inappropriate for "explaining" reasoning tasks because they are at the wrong level of granularity. The inference rules were generated because they were good for computer programs to reason with but were not typically designed for human understanding. Moreover, syntactic manipulations of proofs based on atomic inference rules may also be insufficient for abstracting machine-generated proofs into some more understandable proofs [15]. Proofs, however, can be abstracted when they are rewritten using rules derived from axioms of upper level ontologies such as the SUMO and the DAML [14] axiomatic set. Axioms in rewriting rules are the elements responsible for aggregating some semantics in order to make the rules more understandable. Entries of *DerivedRule* are the natural candidates for storing specialized sets of rewriting rules. In the IW, tactics are rules associated with axioms.

The explainer algorithm generates explanations in a systematic way using the derived rules in the registry. Figure 4 includes the proof information for the same answer as we presented with the explanation information in Figure 3. The explainer has abstracted away both a number of proof steps and details that most users will not want to see. The user may always ask follow-up questions and still obtain the detail, however the default explanation provides abstracted explanations. The general result is to hide the core reasoner rules and expose higher-level derived rules.

4.3.3 Using the Inference Web

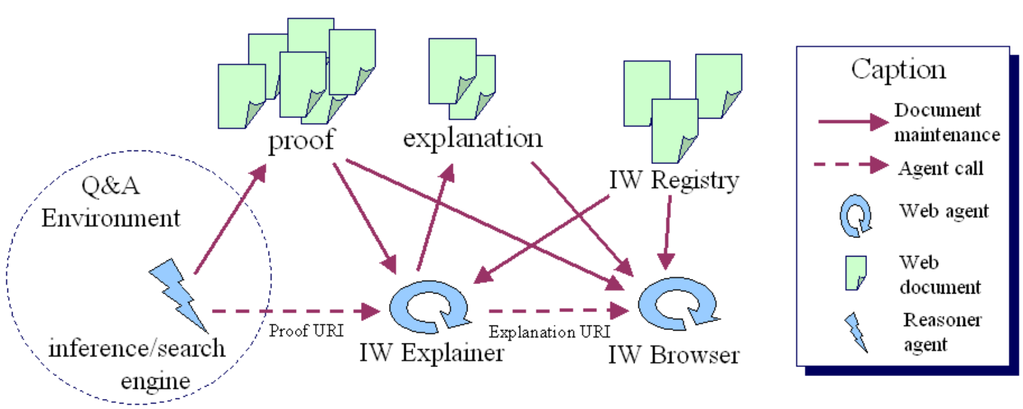

Figure 5: Using the Inference Web.

Figure 5 presents a diagrammatic representation on how a QA environment can use the IW. There, the dashed circle represents a generic QA environment. The lightning symbol represents an inference or search engine embedded in the QA environment used for deriving/retrieving answers. To use the Inference Web, the engine should be able to generate proofs in the IW format. Once the engine generates a proof, it can call the IW explainer passing the URI of the proof fragment at the bottom of the proof (assuming that answers are derived downward). Using registered tactics in the registry the explainer abstracts the proof into an explanation. It then calls the browser and passes in the URI of the fragment at the bottom of the explanation. The browser can then present explanations such as in Figures 1, 2 and 3 or proofs such as in Figure 4. In addition to its support to follow-up questions related to the explanation, the browser can also be asked to present the original proof when browsing explanations. The proof when presented along with its metadata is the most detailed and complex explanation of an answer that the Inference Web can present.

5 Contributions and Future Work

The KSL Wine Agent and the the DAML Query Language Front-End[9] are two current Semantic Web agents supported by the Inference Web. These agents are based on Stanford's JTP theorem prover that produces portable proofs. The IW registry is populated with JTP information: one *InferenceEngine* entry for the reasoner itself, nine entries for its primitive inference rules, one entry for its set of DAML axioms, and 56 entries for the axioms.

If either of these applications wanted to use another reasoner (other than the ones already registered in Inference Web), the inference engine entry would need to be made for the new reasoner along with its core inference rule entries. If the new reasoner was similar to a reasoner already registered, one might find that it was using many of the same core inference rule entries and thus those could be reused. (We found this for example when we registered SNARK after having registered the published inference rules for OTTER.)

Beyond just explaining a single system, Inference Web attempts to incorporate best in class explanations and provide a way of combining and presenting proofs that are available. It does not take one stance on the form of the explanation since it allows engines to dump single or multiple explanations of any information manipulation producing answers. It provides the user with flexibility in viewing fragments of single or multiple explanations in multiple formats. IW simply requires inference rule registration and portable

[9]http://onto.stanford.edu:8080/dql/servlet/DQLFrontEnd

proof format.

We can identify the following contributions:

- *Support for source metadata information* is provided by: the portable proof specification that allows node sets to be associated with sources; and the registry that supports meta information for annotating sources.

- *Support for knowledge process information* is provided by: the proof specification that supports a comprehensive representation of proof trees; and the registry that supports meta information for annotating inference engines along with their primitive inference rules. Also, the proof specification provides support for alternative justifications by allowing multiple inference steps per node set and the proof browser supports navigation of the information.

- *Support for explanation generation* is provided by the registry that supports both formal and informal information about languages, axioms, axiom sets, and derived rules. The proof support for alternative justifications allows derivations to be performed by performance reasoners with explanations being generated by alternative reasoners aimed at human consumption.

- *Support for portable and distributed proofs* is provided by the IW architecture. Portable proofs are specified in the emerging web standard DAML+OIL so as to leverage XML-, RDF-, and DAML-based information services. Proof fragments as well as entire proofs may be combined and interchanged.

- *Support for explanation and proof presentation* is provided by a lightweight proof browsing using the lens-based IW browser. The browser can present either pruned justifications or guided viewing of a complete reasoning path.

- *Support for multiple explanation strategies* is provided by the combination of the Inference Web support for portable and distributed proofs and for explanation and proof presentation.

We are currently extending SRI's SNARK[10] theorem prover to produce portable proofs and simultaneously populating the IW registry with SNARK

[10] http://www.ai.sri.com/~stickel/snark.html

information. Also, we are in the process of discussing the registration of the W3C's CWM[11] theorem prover in the Inference Web.

Future work includes the registration of more inference engines. We also intend to provide specialized support for why-not questions expanding upon [8] and [17]. We are also looking at additional support for proof browsing and pruning. We have also initiated conversations with the verification community in order to provide a portable proof and registry format that meets their needs as well as meeting the needs of the applications that require explanation.

6 Related Work

Recognition of the importance of explanation components for reasoning systems has existed in a number of fields for many years. For example, from the early days in expert systems and MYCIN [26] in particular, expert systems researchers understood the need for systems that understood their reasoning processes and could generate explanations in a language understandable to its users. Inference Web attempts to stand on the shoulders of past work in expert systems, such as MYCIN and the explainable expert system [28] on generating explanations using both their leanings on how to generate explanations and interoperating with next generation systems that generate explanations.

IW also builds on the learnings of explanation in description logics (e.g., [2, 3, 17, 20]) that attempt to provide a logical infrastructure for separating pieces of logical proofs and automatically generating follow-questions based on the logical format. IW also attempts to integrate learnings from the theorem proving community on proof presentation(e.g., [5, 12]) and explanation (e.g., [15]). IW attempts to learn from this and push the explanation component started in Huang's work and also add the emphasis on provenance and distributed environments.

The work in this paper also builds on experience experience designing query components for frame-like systems [4, 13, 17] to generate requirements. The foundational work in those areas typically focus on answers and only secondarily on information supporting the understanding of the answers. In our requirements gathering effort, we also obtained requirements input from contractors in DARPA-sponsored programs concerning knowledge-based appli-

[11]http://www.w3.org/2000/10/swap/doc/cwm.html

cations (the High Performance Knowledge Base program[12], Rapid Knowledge Formation Program[13], and the DARPA Agent Markup Language Program[14] and more recently, the ARDA AQUAINT and NIMD programs). We also gathered requirements from work on the usability of knowledge representation systems (e.g., [21]) and ontology environments (e.g., [11]). We have also gathered needs from the World Wide Web Consortium (W3C) efforts on CWM[15] and the related reasoner effort on Euler[16]. Finally, we gathered knowledge provenance requirements from the programs above and from previous work on data provenance from the database community(e.g., [7]).

7 Conclusion

QA environments can use the Inference Web for explaining their answers. We identified the support for provenance information, reasoning information, explanation generation, distributed proofs, and proof presentation as requirements for explanations in the Web. We described the major components of IW - the portable proof specification based on the emerging web language-DAML (soon to be updated to OWL) supporting proofs and their explanations, the registry, the explainer, and the browser. We described how Inference Web features provide infrastructure for the identified requirements for web explanations. We facilitated use in a distributed environment by providing IW tools for registering and manipulating proofs, proof fragments, inference engines, ontologies, and source information. We also facilitated interoperability by specifying the portable proof format and providing tools for manipulating proofs and fragments. We have implemented the IW approach for two web agents based on JTP and are in discussions with additional reasoner authors to include more reasoning engines. We have presented the work at government sponsored program meetings(RKF, DAML, AQUAINT, and NIMD) to gather input from other reasoner authors/users and have obtained feedback and interest. Current registration work includes SRI's SNARK and W3C's CWM.

Acknowledgments Many people have provided valuable input to our work. Thanks in particular go to colleagues at KSL including Richard Fikes, Jes-

[12] http://reliant.teknowledge.com/HPKB/
[13] http://reliant.teknowledge.com/RKF/
[14] http://www.daml.org
[15] http://www.w3.org/2000/10/swap/doc/cwm.html
[16] http://www.agfa.com/w3c/euler/

sica Jenkins, Gleb Frank, Eric Hsu, Sheila McIlraith, Rob McCool, and Yulin Li for input on JTP, our specification or applications. Also thanks go to a number of colleagues in some government programs who provided input including Hans Chalupsky, Peter Clark, Ken Forbus, Ken Murray, and Steve Reed. All errors, of course are our responsibility.

References

[1] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, May 2001.

[2] Alex Borgida, Enrico Franconi, and Ian Horrocks. Explaining *ACL* Subsumption. In *Proc. of the 14th European Conf. on Artificial Intelligence (ECAI2000)*, pages 209–213. IOS Press, 2000.

[3] Alex Borgida, Enrico Franconi, Ian Horrocks, Deborah L. McGuinness, and Peter F. Patel-Schneider. Explaining *ALC* Subsumption. In *Proc. of the International Workshop on Description Logics (DL'99)*, pages 33–36, Linköping, Sweden, July 1999.

[4] Alex Borgida and Deborah L. McGuinness. Asking Queries about Frames. In *Proceedings of Fifth International Conference on the Principles of Knowledge Representation and Reasoning*, Cambridge, Massachusetts, November 1996. Morgan Kaufmann.

[5] Robert Boyer, Matt Kaufmann, and J. Moore. The Boyer-Moore Theorem Prover and Its Interactive Enhancements. *Computers and Mathematics with Applications*, 29(2):27–62, 1995.

[6] Ronald J. Brachman, Alex Borgida, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lori Alperin Resnick. Living with CLASSIC: When and How to Use a KL-ONE-Like Language. In John Sowa, editor, *Principles of Semantic Networks: Explorations in the representation of knowledge*, pages 401–456, San Mateo, CA, USA, 1991. Morgan Kaufmann.

[7] 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.

[8] Hans Chalupsky and Tom Russ. WhyNot: Debugging Failed Queries in Large Knowledge Bases. In *Proc. of the 14th Innovative Applications of Artificial Intelligence Conference (IAAI-02)*, pages 870–877, 2002.

[9] 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.

[10] Yingwei Cui, Jennifer Widow, 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.

[11] Aseem Das, Wei Wu, and Deborah L. McGuinness. *The Emerging Semantic Web*, chapter Industrial Strength Ontology Management. IOS Press, 2002.

[12] Amy Felty and Dale Miller. Proof Explanation and Revision. Technical Report MSCIS8817, University of Pennsylvania, 1987.

[13] Richard Fikes, Pat Hayes, and Ian Horrocks. DAML Query Language (DQL) Abstract Specification. Technical report, W3C, 2002.

[14] Richard Fikes and Deborah L. McGuinness. An Axiomatic Semantics for RDF, RDF-S, and DAML+OIL (March 2001). Technical Report Note 18, W3C, December 2001.

[15] Xiaorong Huang. Reconstructing Proofs at the Assertion Level. In *Proceedings of CADE-94*, LNAI-814, pages 738–752. Springer, 1994.

[16] Clifford A. Lynch. When Documents Deceive: Trust and Provenance as New Factors for Information Retrieval in a Tangled Web. *Journal of the American Society for Information Science and Technology*, 52(1):12–17, 2001.

[17] Deborah L. McGuinness. *Explaining Reasoning in Description Logics.* PhD thesis, Rutgers University, 1996.

[18] Deborah L. McGuinness. Ontological Issues for Knowledge-Enhanced Search. In N. Guarino, editor, *Proc. of the 1st International Conference on Formal Ontology in Information Systems(FOIS'98)*, pages 302–316, Trento, Italy, June 1998. IOS Press.

[19] Deborah L. McGuinness, Merryll K. Abrahams, Lori Alperin Resnick, Peter F. Patel-Schneider, Rich Thomason, Violetta Cavalli-Sforza, and Cristina Conati. CLASSIC Knowledge Representation System Tutorial. Technical report, Artificial Intelligence Principles Research Department, AT&T Labs Research, 1994.

[20] Deborah L. McGuinness and Alex Borgida. Explaining Subsumption in Description Logics. In *Proc. of the 14th International Joint Conference on Artificial Intelligence*, pages 816–821, Montreal, Canada, August 1995. Morgan Kaufmann.

[21] Deborah L. McGuinness and Peter Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter From Description Logic Provers to Knowledge Representation Systems, pages 265–281. Cambridge University Press, 2003.

[22] Deborah L. McGuinness and Paulo Pinheiro da Silva. Infrastructure for Web Explanations. In *Proceedings of 2nd International Semantic Web Conference (ISWC2003)*, Sanibel, FL, USA, October 2003. Springer. To appear.

[23] Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. Technical report, World Wide Web Consortium (W3C), 2003. Working Draft.

[24] E. Nyberg, T. Mitamura, J. Carbonell, J. Callan, K. Collins-Thompson, K. Czuba, M. Duggan, L. Hiyakumoto, N. Hu, Y. Huang, J. Ko, L. Lita, S. Murtagh, V. Pedro, and D. Svoboda. The JAVELIN Question-Answering System at TREC 2002. unpublished manuscript, November 2002.

[25] 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.

[26] Edward Hance Shortliffe. *Computer-Based Medical Consultations: MYCIN*. Elsevier/North Holland, New York, NY, USA, 1976.

[27] Michael Smith, Deborah L. McGuinness, Raphael Volz, and Chris Welty. Web Ontology Language (OWL) Guide Version 1.0. Technical Report Working Draft, World Wide Web Committee (W3C), 2003.

[28] W. Swartout, C. Paris, and J. Moore. Explanations in Knowledge Systems: Design for Explainable Expert Systems. *IEEE Intelligent Systems*, June 1991.