Explanation Interfaces for the Semantic Web: Issues and Models

Deborah L. McGuinness[1], Li Ding[1], Alyssa Glass[1], Cynthia Chang[1], Honglei Zeng[1], Vasco Furtado[2]

[1] Stanford University, Stanford, CA 94305
{dlm | ding | glass | csc | hlzeng}@ksl.stanford.edu
[2] University of Fortaleza, Washigton Soares 1321, Fortaleza, CE, Brazil
vasco@unifor.br

Abstract. As the Semantic Web has enabled new application capabilities, new interaction modes arise and grow in importance. Applications can now not only retrieve results but also use term meanings to derive new results. Thus, explaining results has become an important new interaction mode for Semantic Web applications. The explanation interaction mode needs to provide transparency and accountability to application results. We have developed an explanation infrastructure that can provide Semantic Web consumers (humans and agents) with explanations for results, such as where results came from and how they were derived. We have addressed explanation requirements for applications that range from intelligent analyst assistants that leverage text analytics to transparent and accountable reasoning systems that protect user privacy. In this paper, we will describe some Semantic Web user interaction requirements and paradigms that are important for Semantic Web applications.

Keywords: explanation, knowledge provenance

1 Introduction

The vision of the Semantic Web includes a world where semantics-powered applications are knowledgeable assistants for end users. The web paradigm shifts from one where users are browsing for information that is largely static to a paradigm where users are also asking questions and expecting answers from applications. The applications producing the results may still use information retrieval techniques to locate answers, but they may also use additional semantics such as formal definitions of terms to perform additional kinds of information access (such as targeted database queries or knowledge base queries) along with information manipulations (such as reasoning using theorem provers or other kinds of inductive or deductive methods). In this new world where Semantic Web applications are potentially doing a combination of information lookup, integration, manipulation, and inference, explanation services become much more important.

Some, including Tim Berners-Lee, have asked for web interfaces to include an "Oh yeah?" button[1] that one presses when one wants to ask "How do I know I can trust this information?" There are many kinds of answers, types of content, and styles of presentation that could be used to support this functionality. The goal of our line of research is to provide an explanation infrastructure for Semantic Web applications. Our Inference Web effort [1] includes a proof markup language (PML) [2] that offers representational constructs for capturing information about where information came from (provenance), how it was manipulated (justifications), and corresponding trustworthiness. PML has an OWL encoding and can be used as a proof and justification Interlingua, and it (along with Inference Web tools) is being used as such in several government-sponsored projects. Inference Web additionally includes a number of tools and services for manipulating the markup – including tools for browsing, filtering, summarizing, searching, and validating the explanations.

We believe explanation and knowledge provenance research has a strong user interaction component. Explanations may be viewed as a special kind of Semantic Web data. A user may need to be in "explanation mode" when they are deciding whether to believe an answer. When they are in this mode, they may require special kinds of browsing and visualization tools. While the previous generation browsers and search interfaces provide starting points for interaction paradigms, they do not appear adequate without additional enhancement geared for explanations.

In the rest of this paper, we will introduce requirements for explanations in a Semantic Web setting. We will describe the types of information that need to be captured along with describing (some) interaction modes that become important. when interacting with the information types. We will focus on browsing, trace (with abstraction and follow-up capabilities), and trust views. We also introduce some supporting infrastructure concerning publishing and accessing explanations. We will conclude with a discussion highlighting evolving interaction issues.

2 Types and Requirements of Explanation

Explanation can be viewed as Semantic Web metadata about how results were obtained. Our experience designing and implementing a wide variety of explanation facilities for Semantic Web applications reveals that explanation metadata generates requirements concerning representation, manipulation, and presentation.

First and foremost, in distributed settings such as the Web, representation interoperability is paramount. We endorse an Interlingua for use in sharing explanation metadata.

Second, transparency is required so that users may be able to find the lineage that often appears hidden in the complex network of the Semantic Web. Note that explanations should not be viewed as a single "flat" annotation, but instead as a web of interconnected objects recording source information, intermediate results, and final results. Support for navigating through this web is required.

[1] http://www.w3.org/DesignIssues/UI.html

Third, a variety of "user friendly" rendering and delivery modes are required in order to present to different types of users in varying contexts [3]. Explanations may need to be delivered to experts or novice (human) users, in addition to machine agents. This variety of uses requires a representation that is flexible, manageable, extensible, and interoperable. Additionally, corresponding presentation modes need to be customizable and context-dependent, and need to provide options for abstract summaries, detailed views, and interactive follow-up support.

We designed and built Inference Web to provide explanation infrastructure to address the above requirements. We designed the Proof Markup Language (PML) to serve as an Interlingua for explanations on the Web. It includes critical basic concepts for representing information about trust, justifications, and provenance. In order to address manipulation and presentation we have built a series of online and standalone tools for publishing, searching, rendering and computing various explanations. Figure 1 illustrates the general architecture of Inference Web. Our design incorporates requirements from explaining web service discovery [4] (with OWL-S [5] and BPEL [6]), policy engines (with N3 [7]), hybrid first order logic theorem provers (with KIF [8]), task execution engines (with SPARK [9]), and text analytic components [10]. The representation language evolved to address needs in this wide range of question answering systems. The toolkit includes registry components for automatic source registration, search capabilities to find justifications meeting particular restrictions, browsing components supporting interactive debugging modes, abstraction components for rewriting justifications and presenting justifications in various views, and trust components for computing, combining, and presenting trust information.

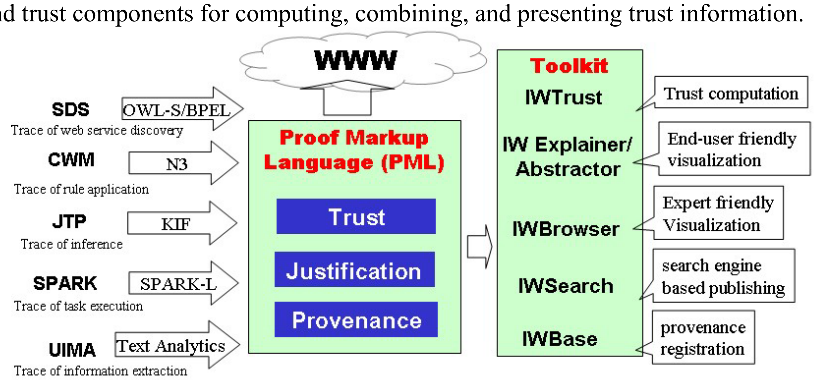

Fig. 1. Architecture of Inference Web Explanation Infrastructure

2.1 Provenance Metadata

Provenance metadata contains annotations concerning information sources, (e.g., when, from where, and by whom the data was obtained). It connects statements in a knowledge base to raw sources, such as web pages and publications, with annotations about data collection or extraction methods, e.g., text analytic components such as those in the Unstructured Information Management Architecture (UIMA) [10]. In order to effectively represent and expose provenance to the end users, we need:

- terms for representing references to (i) asserted statements in a knowledge base; and (ii) information fragments from a source
- a manageable set of core concepts to be used as the basis for annotating the usage of information sources, and to allow further extension
- a machine-processable representation to support rendering, combination, and information filtering

In Inference Web, the provenance portion of PML[2] addresses these requirements. Within the PML-P namespace, we define: (i) a concept *Information* used to refer to individual pieces of information, whose content is referenced either by literal string or by URL; (ii) a class hierarchy rooted at *Source* for annotating types of information sources, such as documents, websites, and people; (iii) concepts related to inference, such as *Axiom* and *InferenceRule*; (iv) supporting concepts for annotating languages, formats, and encoding of information content; and (v) a concept *SourceUsage* for annotating individual accesses of information sources.

Figure 2 shows an example screen shot of provenance metadata. The conclusion is presented as a literal string which is annotated with clickable boxes linking to provenance metadata. Its source metadata is presented in the bottom left window: the source is a publication available at a particular URL. The original source with the highlighted text fragment is shown on the bottom right.

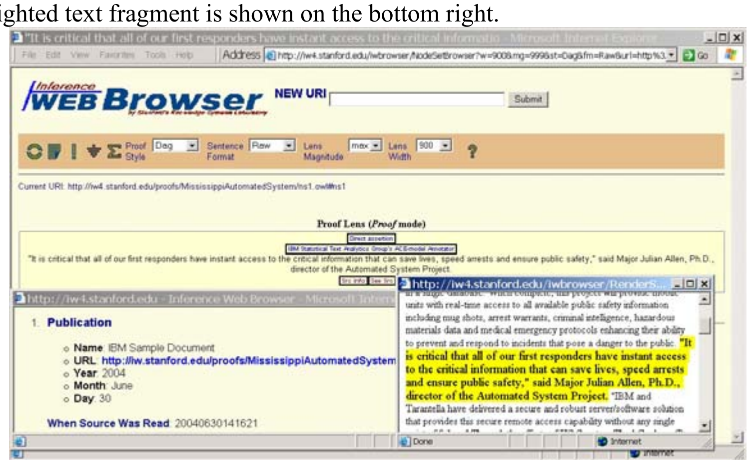

Fig. 2. Example provenance metadata for a piece of information.

2.2 Information Manipulation Traces

Results from Semantic Web applications may be derived from a series of information manipulation steps, each of which applies a primitive information manipulation

[2] Representational constructs for provenance notions. (iw.stanford.edu/2006/06/pml-p.owl).

operation on some antecedents and produces a conclusion. A “trace” is essentially a transaction log for information manipulation steps. When a user requests a detailed explanation of what has been done or what services have been called, it is important to be able to present an explanation based on this trace. In order to effectively represent and expose the trace to the end users, we need:

- appropriate data structures for annotating individual information manipulation steps and their dependencies and
- user friendly rendering mechanisms for browsing the trace.

In Inference Web, PML is used to encode traces using a "node set" concept to annotate critical components of an information manipulation step including the conclusion, the information manipulation step, and its antecedents. Inference Web also provides rich presentation options for browsing the justification traces, including a directed acyclic graph (DAG) view that shows the global justification structure, a collection of hyperlinked web pages that allows step-by-step navigation, a filtered view that displays only certain parts of the trace, an abstracted view, and a discourse view (in either list form or dialogue form) that answers follow-up questions.

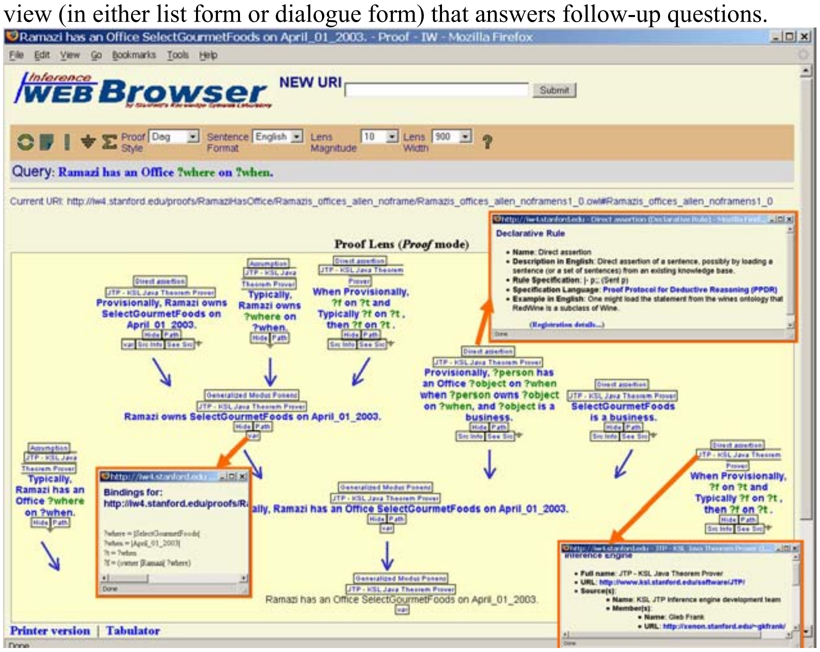

Fig. 3. Trace-Oriented Explanation with Several Follow-up Question Panes

Global View. Figure 3 depicts a screen shot of the IW browser in which the *Dag* proof style has been selected to show the global structure of the reasoning process. The format of the sentences can be displayed in (limited) English or in the reasoner’s native language, and the depth and width of the tree can be restricted using the lens magnitude and lens width options, respectively. The user may ask for additional information by clicking hot links. The three small panes show the results of asking for follow-up information about an inference rule, an inference engine, and the variable bindings for a rule application.

Focused View. Merely providing tools to browse an execution trace is not adequate for most users. It is necessary to provide tools for visualizing the explanations at different levels of granularity and focus. In Figure 4a, our explainer interface includes an option to focus on one step of the trace and display it using an English template style for presentation. The follow-up action pull down menu then helps the user to ask a number of context-dependent follow-up questions.

Filtered View. Alternative options may also be chosen such as seeing only the assertions (ground facts) upon which this result depended, only the sources used for ground assertions, or only the assumptions upon which the result depended. Figure 4b is the result of the user asking to see the sources. As one interesting note, we have found that one popular filtered view is the source collections. Some users are willing to assume that the reasoning is correct and as long as only reliable and recent knowledge sources are used, they are willing to believe an answer. Initially, they do not want to view all the details of the information manipulations (but they do want the option of asking follow-up questions when necessary).

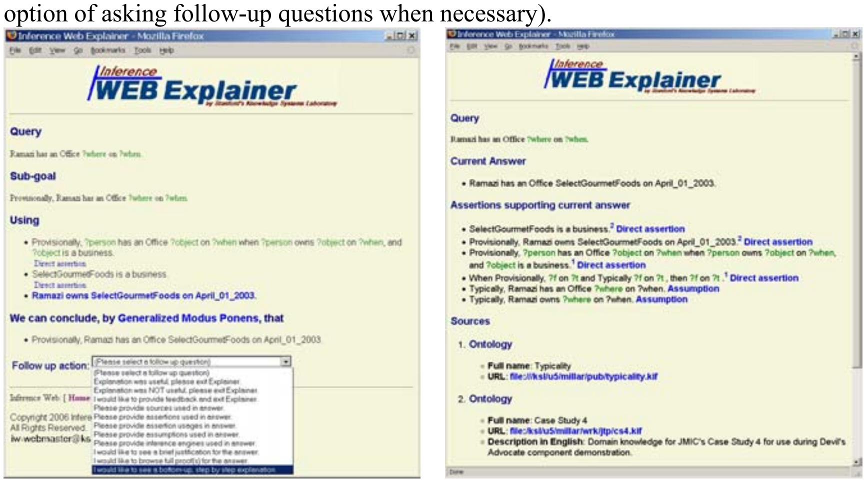

Fig. 4. (a) step-by-step view focusing on one step using an English template, and list of follow-up actions; (b) filtered view displaying supporting assertions and sources

Abstraction View. Machine-generated proofs are typically characterized by their complexity and richness in details that may not be relevant or interesting to all users. Inference Web approaches this issue with two strategies:

- Filter explanation information and only provide one type of information (such as what sources were used). This strategy just hides portions of the explanation and keeps the trace intact.
- Transform the explanation into another form. The IW abstractor component helps users to generate matching patterns to be used to rewrite proof segments

producing an abstraction. Using these patterns, IW may provide an initial abstracted view of an explanation and then provide context appropriate follow-up question support.

The IW abstractor consists of an editor that allows users to define patterns that are to be matched against PML proofs. A matching pattern is associated with a rewriting strategy so that when a pattern is matched, the abstractor may use the rewriting strategy to transform the proof (hopefully into something more understandable). An example of how a proof can be abstracted with the use of a generic abstraction pattern is shown in Figure 5. In this case, the reasoner used a number of steps to derive that crab was a subclass of seafood. This portion of the proof is displayed in the *Dag* style and is outlined in blue. The application author believed that this was more detailed than end users would care to see and so he/she wrote an abstraction pattern that provided a template for matching instances of proofs containing the more complicated proof. With this pattern, the abstractor algorithm can then produce the simpler proof fragment on the left side of Figure 5 with many fewer steps, just showing class transitivity and the classes involved.

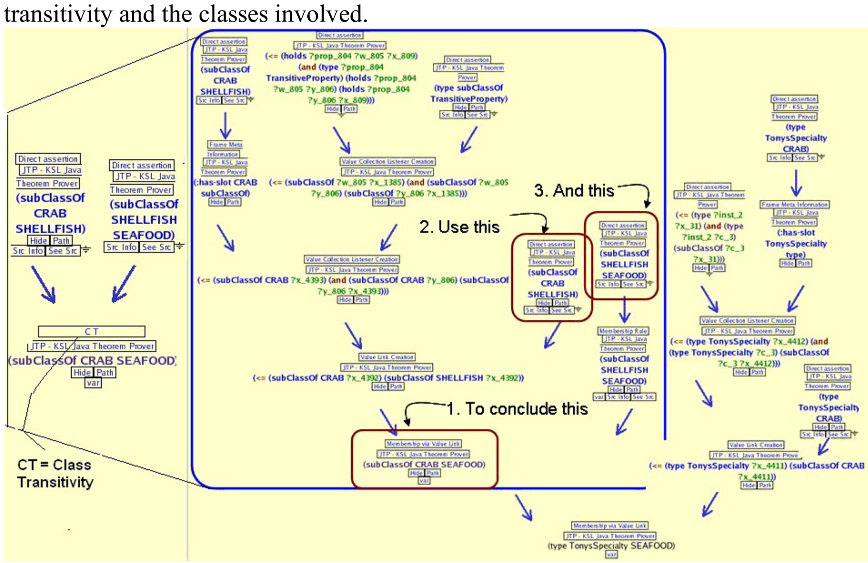

Fig. 5. Example of an abstraction of a piece of a proof

We are building up abstraction patterns for domain independent use, e.g. class transitivity as well as for domain-dependent use. It is an ongoing line of research to consider how best to build up a library of abstraction patterns and how to apply them in an efficient manner.

Discourse View. For some types of information manipulation traces, particular aspects or portions of the trace are predictably more relevant than others. Additionally, the context of the explanation request and a model of the user can often be used to select and combine these portions of the trace, along with suggestions of which aspects may be important for follow-up queries. Particularly for these types of traces, we provide a *discourse view*, which selects trace portions and presents them in simple natural language sentences. As with the various examples above, a full PML justification is generated and stored by the system. In this interaction mode, however, the full details of the inference rules and node structure are kept hidden from the user. Individual nodes, provenance information, and metadata associated with those nodes, are used as input for various explanation strategies, which select just the information relevant to the user's request and provide context-sensitive templates for displaying that information in dialogue form. This same information is also used to generate suggested follow-up queries for the user, including requests for additional detail, clarifying questions about the explanation that has been provided, and questions essentially requesting that an alternate explanation strategy be used.

Figure 6 shows an example of such a discourse, providing an explanation of a trace of a task execution example generated by SPARK as implemented by our CALO explanation component (built using Inference Web). In this dialogue, the user has requested an explanation of the motivation for executing a particular high-level task. An explanation is provided, along with three suggested follow-up questions for the user to choose from. Both the explanations and the suggested follow-up queries are generated in English using simple templates with well-defined parameters filled in by parsing the full justification. For instance, in the example explanation shown in Figure 6, the justification involves the modification of a procedure by adding a conditional test before executing a subtask. The system has chosen a strategy based on a template of the form "You asked me to <current_task>, and instructed me to <current_subtask> under certain conditions: <condition_instruction>." Our current focus for the discourse view has been on justifications of task executions, as exemplified in this figure.

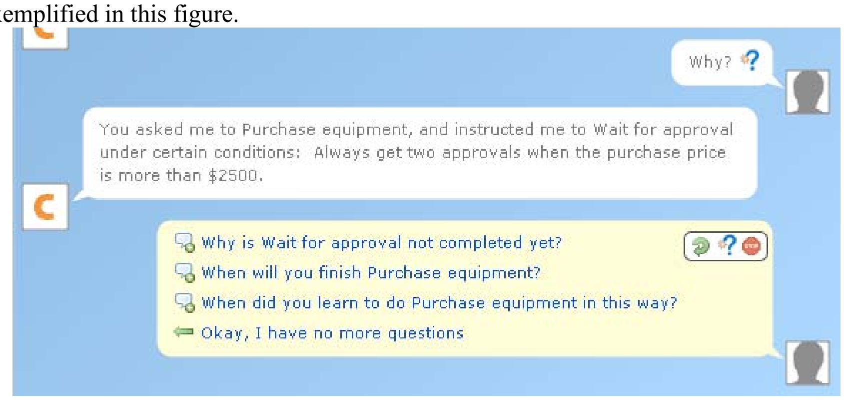

Fig. 6. English discourse view of explanation in CALO project.

2.3 Trust

Trust in the Semantic Web has been a subject of growing interest. A recent survey on the subject can be found in [11]. Inference Web provides a general trust infrastructure, IWTrust [12], to integrate various trust representations and computation services. This general infrastructure is particularly useful for increasing user trust by providing explanations concerning integrated information and filtering out unreliable information. In [13], we augmented Wikipedia with (i) computation modules that derive trust values for each article fragment, (ii) a trust extension of PML, and (iii) a "trust view" that renders article fragments according to their trust annotations. Figure 7 shows Wikipedia augmented with our "trust view" tab. When a user clicks the trust view tab, the fragments of the Wikipedia article that the user is viewing are rendered in different colors based on their trustworthiness, allowing users to gain insight into relative trust from just glancing at the presentation of an article.

The benefits of encoding, computing, and propagating trust information extend far beyond the trust view in Wikipedia. Trust representation, computation, combination, presentation, and visualization present issues of increasing importance for Semantic Web applications, particularly in settings that include large decentralized communities such as online social networks.

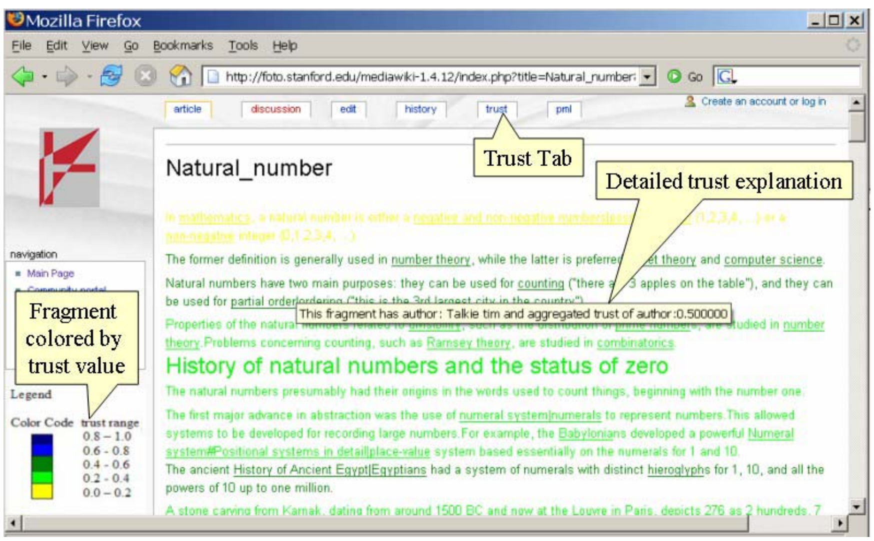

Fig. 7. A snapshot of the Wikipedia article *Natural Number* rendered with trust explanation

3 Publishing and Accessing Online Explanations

In order to share explanations on the Web, effective mechanisms are needed to facilitate publishing and accessing online explanations. Publishing explanations encoded in PML on the Web makes them publicly available; however, end users may not be aware of the presence and address of such explanations. To address this problem, we have investigated two methods that help users locate online explanations. The first is a centralized registry and the second is a web search service.

IW Base [14] offers a traditional registry-based solution for publishing and finding information. Content publishers can log in to a registry website (manually or using IW services) and populate metadata about information sources and supporting information (e.g. languages, inference rules, inference engines). The registry then exposes the metadata in separate PML documents and provides a browsing interface organized by class hierarchy (see Figure 8).

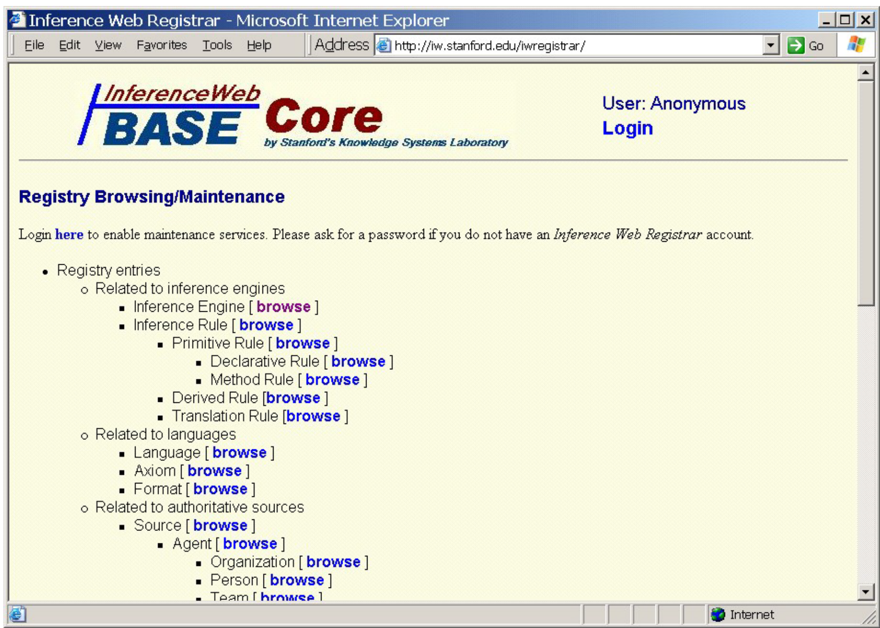

Fig. 8. The browse interface of IW Base Registry

IW Search offers a generic solution that searches for PML documents on the Web. IW Search uses the Swoogle Semantic search engine [15] to generate and maintain an up-to-date inventory of online PML documents. It then allows end users to search for explanations using the following search interfaces:

- any combination of three constraints: the literal descriptions used in explanation (e.g. find all explanations referencing a particular term), the type of explanation (e.g. find a publication or a justification step), and whether the instance is top level (i.e. not referenced by any other explanations, thus it is considered "told"). Figure 9 shows one example IW Search result pane where a user has searched for justifications that contain the term "wine".
- the relations between instances. IW Search can enumerate explanation instances linking-to or linked-by a given explanation instance.

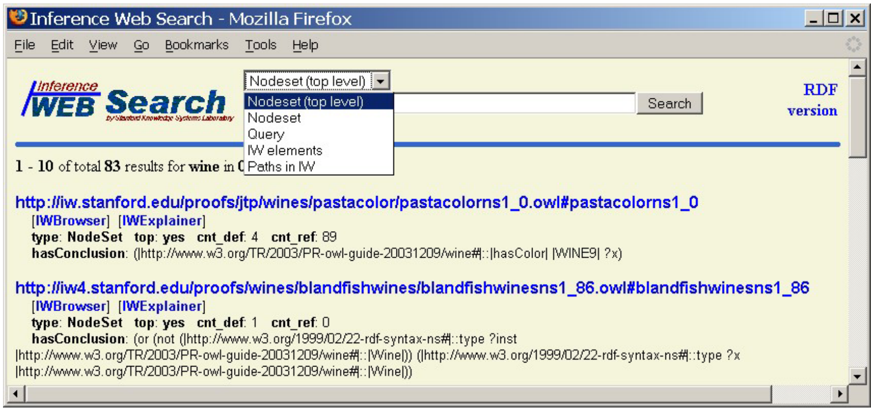

Fig. 9. An example search results page generated by IW Search

4 Discussion and Related Work

While explanations have been a research area for artificial intelligence for many years, interest in and importance of explanation is growing as Semantic Web applications proliferate. Our renewed interest in explanation over the last few years has led us to investigate representation and manipulation requirements for various types of explanation-oriented information. A few recurring themes emerge from our work. In some sense, they may all be viewed as a variation on the theme of user- and context-based customizations of explanations.

Provenance. Centrality and criticality of provenance is evident. Our work began focused on explaining information manipulation traces, but a focus on provenance soon took over as we learned that

- a broad cross section of users demand detailed provenance information before they will believe answers.
- sometimes provenance information (such as source and recency) is the only view that users routinely need. In one of our applications built for intelligence analysts, we were asked much more for source and citation views than any other explanation views [16].

The result for us has been increased emphasis on the provenance component of PML and additional views geared towards presentation of provenance.

Multiplicity of Presentation Styles. Providing a variety of presentation strategies and views is critical to broad acceptance. As we interviewed users both in user studies (e.g., [16]) and in ad hoc requirements gathering, it was consistently true that broad user communities require focus on different types of explanation information in different formats. For any segment that prefers a detailed trace-based view, there is typically a balancing segment that requires an extensively filtered view. This finding results in the design and development of the trace-based browser, the explainer with inference step focus, multiple filtered follow-up views, and the discourse-style presentation component.

Multiplicity of Languages. While it remains clear that an Interlingua is desired for internal representation, it is also clear that users require presentation of information in multiple languages. Many communities require natural language presentations, while others find logical languages (e.g., KIF) to be more useful. In two significant explanation efforts, we have encoded extensive explanation information in a logical language (in these cases, KIF and N3), and our explanation interfaces needed to present English descriptions for the explanations. Our tools include a KIF-English translator, and ongoing work (with N3 colleagues) will provide an N3 to English translator. Our basic infrastructure includes the ability to use a translator if it is available, as well as to use natural language strings for presentation if available.

Positive and Negative Explanation Support. While the typical explanation question concerns why something was deduced or recommended, additional support for why something is not true, or has not terminated, is also important. In our work on one project with a “devil’s advocate” component, this negative support became even more critical. We have additional work underway on explaining a broader range of questions, e.g. supporting cognitive assistants to explain "why-not" questions [17].

Domain-Independent and Domain-Dependent Presentation. In an effort aimed at building a transparent accountable data mining system for lawyers and judges [18], it has become important to explain why information usage is justified or unjustified. In this system, we are exploring (i) legally-literate techniques for presenting explanations of legal interpretations for lawyers; and (ii) domain independent techniques for checking justifications against policies such as privacy and security. Moreover, the system may also be used to help lawyers find an argument FOR and AGAINST a particular conclusion to help support the legal argumentation style.

Abstraction: Inference Web currently supports abstraction with editor and abstractor components. We currently adopt KIF as a *lingua franca* for representing the axioms of the abstraction pattern. The algorithm to process abstractions has been tested in the context of the Knowledge Associates for Novel Intelligence (KANI) project, which focuses on supporting intelligence analysis tasks. A repository of eleven patterns was used to match against proofs of up to ten depth levels, with an average branch factor of four. We are investigating efficient application strategies to improve the performance of the algorithm with larger proofs.

Trust. Representation and presentation of trust are critical to explanation, especially in supporting decision-making. Existing work in this area (for instance, the TRELLIS system [19]) has studied argumentation-like explanation using Semantic Web-based vocabulary and tools, and recently TriQL.P browser [20] studied detailed explanations focusing on application of trust policy and derivation of trust metrics. We are investigating presentation and computation issues in collaborative information integration contexts that use a wide array of question answering components and information sources.

We advocate a view of explanations as a special test case for handling Web data, because it provides additional information for why and how other Web data has been obtained. Any well-designed generic Semantic Web browser would necessarily need to provide users with a mechanism for finding and presenting justifications for the data being browsed. Moreover, a usable Semantic Web would need to offer users a flexible interface for browsing the overall structure and relevant detailed portions of provenance- and trace-based explanations. We have designed and implemented some semantic web browsing capabilities aimed at explanations that exploit semantic technologies informed by OWL ontologies and using web services. In addition to browsers, facilitating services are also needed for assisting browsers in accessing and manipulating explanations (for example, search/registry based tools for finding relevant explanations).

Our work concerning the development of explanation interfaces has been driven by end users' needs. In particular, we have customized our explanation tools for a range of projects and users as a result of extensive requirements gathering for such efforts as CALO (for explaining task planning and execution), TAMI (for explaining policy-based data usage), NIMD (for explaining intelligence analyst tools), and UIMA (for explaining information extraction processes),.

5 Conclusion

In this paper, we have identified a few types of explanation information that are of particular interest in Semantic Web applications, including provenance metadata, information manipulation traces, and trust information visualizations. All of them generate representation and interaction requirements. In order to address these issues, we developed Inference Web, which provides a representation Interlingua, PML, and a set of tools and services for manipulating and presenting those representations. We are actively exploring new models and interaction styles for evolving explanation interfaces for the Semantic Web.

Acknowledgments. We gratefully acknowledge funding support for this effort from NSF contract #5710001895, DARPA contracts #F30602-00-2-0579, #HR0011-05-0019, #55-300000680 to-2 R2, and DTO contract #2003*H278000*000.

References

1. Deborah L. McGuinness and Paulo Pinheiro da Silva. Explaining Answers from the Semantic Web: The Inference Web Approach. Journal of Web Semantics 1(4). (2004)
2. Paulo Pinheiro da Silva, Deborah L. McGuinness and Richard Fikes. A Proof Markup Language for Semantic Web Services. Information Systems 31(4-5). (2006)

3. Gabriella Cortellessa and Amedeo Cesta. Evaluating Mixed-Initiative Systems: An Experimental Approach. In: ICAPS'06. (2006)
4. Deborah L. McGuinness, Dan Mandell, Sheila Mcllraith, Paulo Pinheiro da Silva. Explainable Semantic Discovery Services. Stanford Networking Research Center Project Review, February 17, 2005, Stanford, CA
5. David Martin, Massimo Paolucci, Sheila McIlraith, Mark Burstein, Drew McDermott, Deborah McGuinness, Bijan Parsia, Terry Payne, Marta Sabou, Monika Solanki, Naveen Srinivasan, Katia Sycara, "Bringing Semantics to Web Services: The OWL-S Approach", In: the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC'04). (2004)
6. Business Process Execution Language for Web Services (BPEL), Version 1.1. (last modified in Feb 2005). (2003) http://www.ibm.com/developerworks/library/ws-bpel
7. Tim Berners-Lee. Notation 3 – Ideas about Web architecture. (last modified in Mar 2006) (1998) http://www.w3.org/DesignIssues/Notation3.html
8. Michael R. Genesereth and Richard E. Fikes, Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Computer Science Department, Stanford University. (1992) http://logic.stanford.edu/kif/Hypertext/kif-manual.html
9. David Morley and Karen Myers. The SPARK Agent Framework. In: the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'04). (2004) http://www.ai.sri.com/~spark/
10. Dave Ferrucci and A. Lally, UIMA by Example, IBM Systems Journal 43(3). (2004)
11. Donovan Artz and Yolanda Gil, "A Survey of Trust in Computer Science and the Semantic Web". submitted for publication. (2006)
12. Ilya Zaihrayeu, Paulo Pinheiro da Silva and Deborah L. McGuinness. IWTrust: Improving User Trust in Answers from the Web. In: the 3rd International Conference on Trust Management (iTrust'05). (2005)
13. Deborah L. McGuinness, Honglei Zeng, Paulo Pinheiro da Silva, Li Ding, Dhyanesh Narayanan and Mayukh Bhaowal. Investigations into Trust for Collaborative Information Repositories: A Wikipedia Case Study. In: the Workshop on the Models of Trust for the Web (MTW'06). (2006)
14. Deborah L. McGuinness, Paulo Pinheiro da Silva and Cynthia Chang. IWBase: Provenance Metadata Infrastructure for Explaining and Trusting Answers from the Web. Technical Report KSL-04-07, Knowledge Systems Laboratory, Stanford University, USA. (2004)
15. Li Ding, Rong Pan, Tim Finin, Anupam Joshi, Yun Peng and Pranam Kolari, Finding and Ranking Knowledge on the Semantic Web, In: the 4th International Semantic Web Conference (ISWC'05). (2005)
16. Andrew J. Cowell, Deborah L. McGuinness, Carrie F. Varley, and David A. Thurman. Knowledge-Worker Requirements for Next Generation Query Answering and Explanation Systems. In: the Workshop on Intelligent User Interfaces for Intelligence Analysis, International Conference on Intelligent User Interfaces (IUI'06). (2006)
17. Deborah L. McGuinness, Paulo Pinheiro da Silva and Michael Wolverton. Plan for Explaining Task Execution in CALO. Stanford KSL Technical Report. KSL-05-11. (2005)
18. Daniel J. Weitzner, Hal Abelson, Tim Berners-Lee, Chris P. Hanson, Jim Hendler, Lalana Kagal, Deborah L. McGuinness, Gerald J. Sussman, K. Krasnow Waterman. Transparent Accountable Inferencing for Privacy Risk Management. In: AAAI Spring Symposium on The Semantic Web meets eGovernment. AAAI Press, Stanford University, USA. (2006)
19. Yolanda Gil and Varun Ratnakar, Trusting Information Sources One Citizen at a Time, In: the 1st International Semantic Web Conference (ISWC'02), 2002
20. Christian Bizer, Richard Cyganiak, Tobias Gauss, and Oliver Maresch, The TriQL.P Browser: Filtering Information using Context-, Content- and Rating-Based Trust Policies, Semantic Web and Policy Workshop, (2005).