Registry-Based Support for Information Integration

Deborah L. McGuinness and Paulo Pinheiro da Silva
Knowledge Systems Laboratory
Stanford University

Abstract

In order for agents and humans to leverage the growing wealth of heterogeneous information and services on the web, increasingly, they need to understand the information that is delivered to them. In the simplest case, an agent or human is retrieving "look-up" information and would benefit from having access to provenance information concerning recency, source authoritativeness, etc. In more complicated situations where information is manipulated before it is returned as an answer, agents and humans would benefit from understanding the derivations and assumptions used. When services are involved, users and agents also would benefit from understanding what actions could be or were executed on the user's behalf. In this paper, we introduce a strategy for registering information sources and question answering systems providing support for implementing distributed and cooperative web services. In this paper, we describe the inference web infrastructure that supports explanations in distributed environments such as the web and describe the elements of its registry.

1 Introduction

If the web's promise of highly leveraged, interoperable, distributed information services is to be realized, consumers of the service need to understand the information and how to use it. This means that the consumer needs to make decisions about when to trust service results. Thus, consumers need to know why an application provided them with information, why the information should be believed, and if and why any actions were executed on the consumer's behalf. In short, the consumer needs to have access to explanations for information and actions.

The goal of our work is to facilitate trust and interoperability by providing proofs and explanations in a distributed and combinable manner. Our work is partially motivated by the explanation needs gathered from a few government sponsored research projects aimed at generating, evolving, and leveraging large knowledge sources, in particular DARPA's Agent Markup Language[1] Project and its Rapid Knowledge Formation[2] project and ARDA's Novel Intelligence for Massive Data[3] and its Advanced Question and Answering for Intelligence[4] programs. Our work is also motivated by experience supporting long-lived applications (e.g. PROSE [McGuinness and Wright, 1999]) where we found that evolution (e.g., Chimaera [McGuinness, et al., 2000]) and [McGuinness, 2000]) and explanation environments (e.g., [McGuinness, 1996], [Borgida, et al., 1999]) were critical for the longevity of the deployment. Some of the needs that we want to address are to support users who need to know:

- information source, recency, and pedigree
- how conclusions were derived
- what if any assumptions were used
- what terms mean and what their inter-relationships are

In the rest of the paper, we will briefly describe the Inference Web focusing on its registry content and infrastructure. We also describe how the registry can be used to realize the Inference Web, which is our approach for handling proofs and their explanations in distributed setting such as the web. These explanations provide the foundation for allowing consumers (agents and humans) to decide when and how much to trust information and results.

2 Inference Web and the Registry

Inference Web [McGuinness and Pinheiro da Silva, 2003] provides an infrastructure for proofs [Pinheiro da Silva and

[1] http://www.daml.org
[2] http://reliant.teknowledge.com/RKF/
[3] http://www.ic-arda.org/Novel_Intelligence/
[4] http://www.ic-arda.org/InfoExploit/aquaint/

McGuinness, 2003] and it uses the registry of information. The Inference Web framework is composed of a specification of proofs and proof elements, tools for handling proofs (viz., proof browsers, parsers, etc.), a registry of information supporting the explanation of proofs and a registrar to handle the registry.

2.1 The Registry

The Inference Web registry is a repository of information relevant for explaining answers provided by a wide variety of retrieval tools such as database management systems, web search engines, and inference engines. Each entry in the registry is stored as a small and self-contained file. Entries are pieces of information ready to be used by users and agents to facilitate the composition of queries, answers, proofs of answers, explanations of proofs, etc. The registry emerged as a necessary component of the IW infrastructure for proofs and their explanations. However, the registry can play a more fundamental, general role on query-answering systems since it can also provide infrastructure for a variety of querying/reasoning tasks other than explanations. For instance, the registry can support the collaboration of multiple agents towards the composition of complex web services by providing metadata describing agent capabilities.

Principles guiding the design of the registry include:

- Interoperability: Every entry is a file written in DAML [Connolly, et al, 2001]. Thus, information in each entry has a precise interpretation since they are specified using the DAML vocabulary[5]. Specifications of information on entries are also based on the InferenceWeb vocabulary[6] derived from the DAML vocabulary. Logical sentences used in entries are based on the KIF Interchange Format (KIF[7]).
- Distributability: The registry is a hierarchical interconnection of repositories of information. Each repository specifies a namespace used to identify its entries in a unique way. Registry repositories can be made available in the Web. Entries in one repository can be based on entries of another repository.
- Scalability: Entries are typically small files with an average size of 2 Kilobytes. Moreover, retrieval of entries can be restricted to the ones relevant for a particular context since entries can be directly referred and retrieved.

Name resolution uses the base addresses of registry instances. Currently, the registry relies on W3C's URLs for specifying the base addresses of their instances. For instance, the base address of the registry at KSL is http://www.ksl.stanford.edu/software/IW/registry/. More sophisticated approaches for name resolution such as Persistent URL[8] and the Handle System [Sun and Lannom, 2002] will be considered in a near future.

2.2 The Registrar

The Inference Web registrar is a web agent in charge of administering the registry. The registrar may do things such as granting update or access privileges for updating some categories of information to selected users and they can define and implement policies for accessing the registry.

Name resolution within a registry instance, rendering of entries for human presentation, and browsing of entries are examples of registrar's functionalities. These functionalities are available on the KSL Inference Web registrar at http://www.ksl.stanford.edu/software/iw/

In this paper we focus on the description of the categories of entries of the IW registry.

3 Registry Concepts

Registry concepts are terms of the Inference Web vocabulary. In this section we describe how concepts are realized by registry entries. Further, we describe how these concepts are interconnected.

3.1 Concepts and Entries

Registry entries are instances of concepts related to tools retrieving and presenting information. Moreover, these tools may use diversified sources to gather and manipulate information presented to users or agents.

It may be challenging to disclose the information most relevant to results of any particular retrieval process. For instance, in the context of database systems, users may want to know the original sources of query answers as a default feature of the systems if they are going to trust the results [Buneman, et al., 2001]. Moreover, in the context of deductive inference engines, users may also want to understand the process of deriving information if they are going to trust the deductive engine results.

Trust disclosure is a major concern in the design of the registry. For instance, each entry is a *RegistryElement*[9] as presented in Figure 1. There we can see that a *RegistryElement* has a name, a description in English, a

[5] http://www.daml.org/2001/03/daml+oil
[6] http://www.ksl.stanford.edu/software/IW/spec/iw.daml
[7] http://logic.stanford.edu/kif/kif.html
[8] http://purl.oclc.org/
[9] Our convention is to italicize registry terms.

URI, and a URL. The URI is the entry's absolute address. The URL is an optional attribute referring to a resource in the Web further describing the information in the entry. Users authorized to update the registry are submitters who should also be registered in the registry as *Sources*. The registry has a default *Source* template entry for registrar administrators used for bootstrapping the registration of users. Along with a submitter, the registrar also records the first and last submission date of each entry.

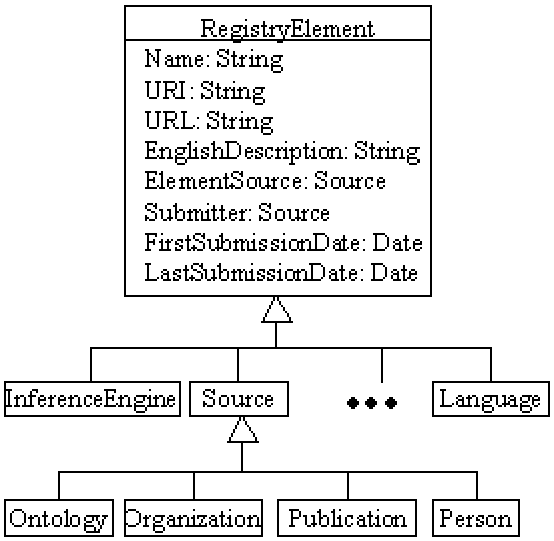

Figure 1. Registry Elements

The identification of the submitter of each entry is an Inference Web mechanism for creating a "web of trust".

Although essential for trusting an entry, the identification of the submitter of the entry may not add value to explaining its content. Each entry can also be associated with an original source for the information called the *ElementSource*. For example, as described in the next section, an *InferenceEngine* is a registry concept. The *Organization* developing the inference engine may be the *ElementSource* of the engine entry. For example, Oracle Corporation could be an element source for Oracle 9i RDBMS although someone interested in populating the IW registry may be the submitter of the information on Oracle 9i.

The registry specifies four basic classes of sources: *Person*, *Publication*, *Ontology*, and *Organization*. Moreover, *Sources* may be *ElementSources* of other *Sources*.

We are currently expanding the description of the relationship between sources and query answers and will expand the specification of (authoritative) sources as required. We are starting with a minimal source description specification for the inference web. Attributes of sources are those mainly inherited from the *RegistryElement*. However, the long-term plan of the project is to refine the specification of these sources by specifying additional attributes and subclasses for *Source*. For example, we can refine the *Publication* concept by adding attributes already available in popular tools for handling publication references, e.g., Bibtex and EndNote, and use vocabulary terms from common sources such as the Dublin Core[10]. We also anticipate a scheme such as that used by UNSPSC[11] for making additions to the vocabulary specification.

3.2 Core Concepts

InferenceEngines and *Language*s are, along with *Sources*, the three core concepts in the registry.

InferenceEngines include all tools capable of retrieving information either by straight "look-up" or by any kind of inferential process. Inference Engines are characterized by their *InferenceRules.* An inference rule for the Inference Web is defined by a set of sentence patterns for the premises, a sentence pattern for the conclusion, and optional side conditions. All patterns and conditions are currently specified in KIF format. In addition to these specifications in terms of KIF sentences, *InferenceRules* can also have a textual description and example in English. Inference rules are typically those implemented in deductive reasoners such as resolution, generalized modus pones, and demodulation.

A query answer is the conclusion of the last application of an inference rule (or last inference step) in a proof tree. Thus, if a tool does not perform any inference such as a non-deductive database management system, a query answer may be considered to be the conclusion of the application of a Told inference rule, which just retrieves the told information. In this case, the proof tree consists of this single inference step. If a tool is a deductive engine, and some inference other than lookup is used, the Told inference rule may be used to associate assumptions with query answers. Proof trees stop when all branches end in a Told inference rule application. The source of the told information connects the ontology containing the told information to the query answer.

Inference engines may use specialized language axioms to support a language such as DAML, OWL, or RDF. Axiom sets such as the one specified in [Fikes-McGuinness, 2001] may be used as a source and specialized rewrites of those axioms may be used by a particular theorem prover to reason efficiently. Thus proofs may be dependent upon these language specific axiom sets called *LanguageAxiomSets* in the Inference Web. It is worth noting that one language may have a number of

[10] http://dublincore.org/
[11] http://www.unspsc.org

LanguageAxiomSets as different reasoners may find different sets of axioms to be more useful. Also, individual axioms may be included in multiple *LanguageAxiomSets*. The content attribute of axiom entries contains the axiom stated in KIF.

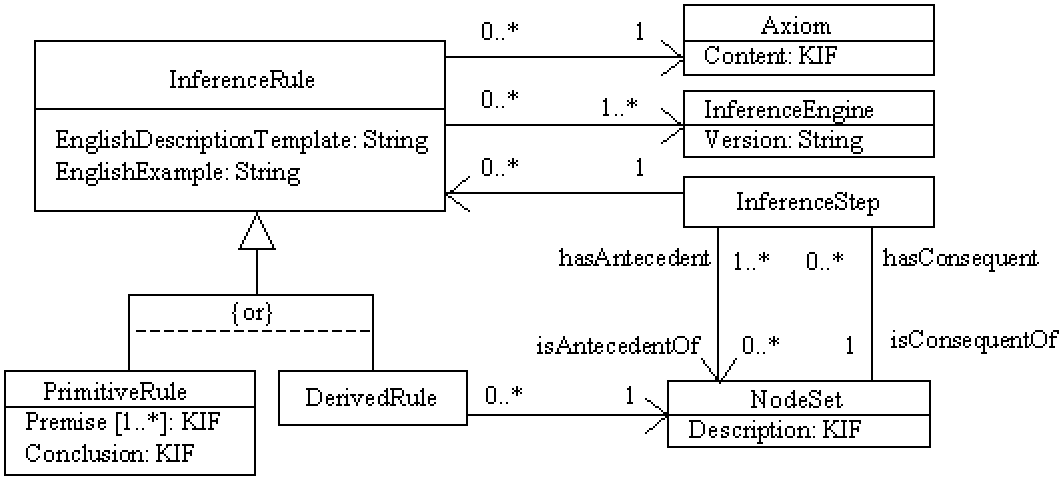

Figure 2: Inference Web concept relationships in UML.

3.3 Concept Relationships

In this section we describe the inter-relationships between the concepts used in the Inference Web registry. While some explanations may not expose much of this information, it is important to have a thorough logical foundation on which to rest the infrastructure. The logical infrastructure provides a precise semantics that facilitates interoperability and extensibility. It also provides the framework that allows the system to automatically generate the appropriate follow-up question based on the structure of the inference application. Thus perusal through the proof or explanation is dependent upon the foundation. It also provides the foundation for the rewrite rules or tactics that allow the system to transform more complicated proofs into more understandable (more abstracted) explanations.

Every *RegistryElement*, including *InferenceEngines* and *Languages*, may be associated with a *Source*. More complex however is the relationship between *Sources* that are *Ontologies* and *InferenceEngines*. The UML class diagram in Figure 2 graphically shows that these two concepts are related through proofs composed of instances of *NodeSet* and *InferenceStep*[1]. An *Ontology* is typically associated with a *NodeSet* when an instance of the *InferenceStep* deriving a NodeSet is an application of a Told inference rule.

A query answer can have multiple justifications since one NodeSet can be justified by many InferenceSteps. Consequently, the answer can have multiple sources for data. For instance, this may be the case of an event reported by several people where each one of them may have distinct interpretations of the event. This situation may require a model of data provenance such as the one supported by the IW. In fact, the deterministic model [Buneman, et al, 1999] for data provenance used on scientific databases may be, for example, unable to represent the situation above due to its restriction of having a unique path to the location of any piece of data.

One other complex relationship concerns *InferenceEngines* and *Languages*. Figure 2 presents the indirect relationship between *InferenceEngine* and *Axiom*, which is used to specify *Language*. Further, an *InferenceEngine* may be associated with many *InferenceRules*. Rules directly implemented by engines are called *PrimitiveRules*. These rules are primitive with respect to their associated *InferenceEngines*, even if they can be derived from other rules unrelated to their associated inference engines.

InferenceRules can also be derived from other *InferenceRules*. *DerivedRules* are defined in terms of proof fragments that are combinations of *NodeSets* and *InferenceSteps*, as further described in [Pinheiro da Silva and McGuinness, 2003]. They are derived since each *InferenceStep* is the application of one *InferenceRule*, as represented by the association between the two concepts in Figure 2. The arrow in the association means that InferenceStep has visibility of InferenceRules.

Tactics are *InferenceRules* that have the restriction that one of its premises is also an *Axiom* of a registered *Language*. Thus, a *Tactic* is the main concept making the connection between *InferenceEngines* and *Languages*. In the context of

[1] *NodeSet* and *InferenceStep* are Inference Web concepts used for building proofs and explanations. Their instances are not registered in the IW Registry.

the Inference Web, *Tactics* are used by the explanation system to simplify the proof presentation. As shown in the example on the Inference Web site at: http://www.ksl.stanford.edu/software/IW/Ex1/ , they can be used to compress the deductive process and present explanations without including as much of the reasoning engine specific information.

Languages are also useful for characterizing *InferenceEngines* since engines are able to support a *Language* as far as they are able to support the *Axioms* of the *Languages*.

4 Discussion

In some ways, the registry arose because it was a necessary component of our solution to our goal of explaining provenance and deduction in distributed settings such as the web. However, the registry is much more than a simple resource for generating explanations. The registry serves as a resource for much more than information content. In its simplest mode as a collection of input offered by authors of ontologies, reasoners, languages, and language axiom sets, it can become a valuable collection of sources of information.

As an author of many ontology-based applications, it is often the case that we would like a quick source of ontological information concerning a particular subject. The registry contains information about ontologies, their update information, their authoritativeness rating, as well as being connected to information concerning how often any particular ontology was accessed for proof delivery from Inference Web. Even a simple starting point such as the structure used in the DAML ontology library[2] can be extremely useful for searching for ontologies submitted, term names used in multiple ontologies etc. The entry of an ontology may contain this information along with meta-information about when it was updated, by whom, for what purpose, and some information about how the ontologies were used (at least for proof generation and delivery). As Inference Web proofs become more used for trust and validation applications, it is possible that the information about how often ontologies are used or particular terms in ontologies are used could be an indicator of the value of the ontology. Also, the ontology library can be connected to merging and diagnostic tools such as Chimaera and Anchor Prompt that provide support in merging ontologies and in Chimaera's case ontology diagnostics as well.

Also, as an author of many applications that use reasoners, it is often the case that during an initial design and feasibility study, we need to consider what reasoner(s) make sense to use in a project. In its simplest form, the registry contains listing information about reasoners, contact information, possibly licensing information, along with the inference methods supported. In the expected case it also contains information about the inference rules used by particular reasoners and for fully supported reasoners, such as JTP in the current Inference Web implementation, it also contains tactics for generating explanations. The registry also is connected to information about how often any particular reasoner was used in proof delivery thus it has a measure of how often other applications that require proofs have used any particular reasoner. It can also be connected to other repositories of information about reasoners such as the QPQ project[3], which is a repository for peer-reviewed source code for deductive software components

Also, as an author of some axiom sets for reasoner implementation, it can be valuable to look at other axiomatic specifications. The registry contains a set of core inference rules implemented by different reasoners and also contains language axiom sets that have been used with particular reasoners. We expect to find over time that there may be multiple axiom sets for the same language as different reasoners may find that different axiomatic specifications are more useful/efficient for particular uses. Application developers may find value in inspecting the alternative axiom sets. Researchers who are interested in checking axiom consistency (e.g., [Baclawski, et al, 2002]) may also make use of the axioms. Similar to the cases above, the registry is also connected to information about how often a particular inference rule or axiom was used in any proof delivered by the Inference Web. Thus if one is optimizing a reasoner according to expected use, one could consider putting more work into optimizing strategies for handling the most used inference rules.

The registry also contains a listing of languages such as DAML [Connolly, et al., 2001] or OWL [Dean, et al., 2002] that have axiomatic specifications, e.g., [Fikes-McGuinness, 2001]. It can be useful to see if any reasoners have associated language axiom sets for any particular language. The registry also contains a listing of tactics used to generate explanations. The precise logical specification of the tactics (along with the explanations) may be useful in a number of ways for implementers. Also, some theoreticians may find access to the KIF statements to have value for things such as verification through systems such as Specware[Specware, 2001].

In summary, we continue to find uses for the growing body of information in the registry. While this began as a support mechanism for distributed explanations, we

[2] http://www.daml.org/ontologies

[3] http://www.qpq.org/

believe that the Registry will have broader impact for many uses in information integration.

5. Conclusion

We have described a registry-based approach to proofs and their explanations in distributed settings such as the web. We introduced the concepts valuable from an explanation perspective and described how they are stored and used in the Inference Web registry. We also discussed some of the many benefits obtainable once even a partial registry is web accessible. The registry approach introduced in this paper provides a uniform strategy for trust disclosure for answers produced by heterogeneous tools querying diversified sources of information on the web. Inference Web is available for use at http://www.ksl.stanford.edu/software/iw/.

Acknowledgements

Many people have provided valuable input to our work. Thanks in particular go to colleagues at KSL including Richard Fikes, Jessica Jenkins, Gleb Frank, Eric Hsu, 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

[Baclawski, et al, 2002] Kenneth Baclawski, Mieczyslaw M. Kokar, Richard J. Waldinger, and Paul A. Kogut: Consistency Checking of Semantic Web Ontologies. International Semantic Web Conference 2002: 454-459.

[Borgida et al., 1999] Alex Borgida, Enrico Franconi, Ian Horrocks, Deborah McGuinness, and Peter Patel-Schneider. ``Explaining ALC subsumption" Proceedings of the International Workshop on Description Logics (DL-99), Linköping, Sweden, July 1999, pp 33-36.

[Buneman et al, 1999] Peter Buneman, Alin Deutsch and Wang-Chiew Tan. ``A Deterministic Model for Semistructured Data'' Proceedings of the Query Processing for Semistructured Data and Non-standard Data Formats, 1999, pp 14-19.

[Buneman et al, 2001] Peter Buneman, Sanjeev Khanna and Wang-Chiew Tan. "Why and Where: A Characterization of Data Provenance'' Proceedings of 8th International Conference on Database Theory, January 2001, pp 316-330.

[Connolly et al, 2001] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. DAML+OIL (March 2001) Reference Description. W3C Note 18 December, 2001. http://www.w3.org/TR/daml+oil-reference.

[Dean et al., 2002] Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah McGuinness, Peter Patel-Schneider, and Lynn Andrea Stein. OWL Web Ontology Language 1.0 Reference. World Wide Web Consortium (W3C) Working Draft 29 July 2002. Latest version is available at http://www.w3.org/TR/owl-ref/.

[Fikes and McGuinness, 2001]. Richard Fikes and Deborah L. McGuinness. An Axiomatic Semantics for RDF, RDF-S, and DAML+OIL (March 2001). World Wide Web Committee (W3C) Note 18 December 2001. http://www.w3.org/TR/daml+oil-axioms.

[McGuinness, 1996] Deborah L. McGuinness. 1996. *Explaining Reasoning in Description Logics.* Ph.D. Thesis, Rutgers University, Technical Report LSCR-TR-277.

[McGuinness, 2000]. Deborah L. McGuinness. ``Conceptual Modeling for Distributed Ontology Environments," In the Proceedings of the Eighth International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues (ICCS 2000), August 14-18, 2000, Darmstadt, Germany. http://www.ksl.stanford.edu/people/dlm/papers/iccs00-abstract.html

[McGuinness, et al., 2000] Deborah L. McGuinness, Richard Fikes, James Rice, and Steve Wilder. An Environment for Merging and Testing Large Ontologies. In the Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, USA. April 2000. http://www.ksl.stanford.edu/people/dlm/papers/kr00-abstract.html.

[McGuinness and Pinheiro da Silva, 2003] Deborah McGuinness and Paulo Pinheiro da Silva. Inference Web: Portable and Shareable Explanations for Question Answering ". In the Proceedings of the American Association for Artificial Intelligence Spring Symposium Workshop on New Directions for Question Answering. Stanford University, Stanford, CA. March 2003.

[McGuinness and Wright, 1998] Deborah L. McGuinness and Jon Wright. ``An Industrial Strength Description Logic-based Configurator Platform". *IEEE Intelligent Systems*, Vol. 13, No. 4, July/August 1998, pp. 69-77.

[Pinheiro da Silva and McGuinness, 2003] Paulo Pinheiro da Silva and Deborah L. McGuinness. Combinable Proof Fragments for the Web. Submitted for publication.

[Smith et al., 2003] Michael Smith, Deborah L. McGuinness, Raphael Volz and Chris Welty. Web Ontology Language (OWL) Guide Version 1.0. World Wide Web Consortium (W3C) Working Draft. Available at http://www.w3.org/TR/owl-guide.

[Specware, 2001] http://www.specware.org/

[Sun and Lannom, 2002] Sam X. Sun and Larry Lannom. Handle System Overview. CNRI, September, 2002. http://www.ietf.org/internet-drafts/draft-sun-handle-system-10.txt.