Chapter I
Explaining Semantic Web Applications

Deborah L. McGuinness
Tetherless World Constellation, Rensselaer Polytechnic Institute (RPI), and Stanford University, KSL, USA

Vasco Furtado
University of Fortaleza, UNIFOR, Brazil

Paulo Pinheiro da Silva
University of Texas at El Paso (UTEP), USA

Li Ding
Tetherless World Constellation, Rensselaer Polytechnic Institute (RPI), and Stanford University, KSL, USA

Alyssa Glass
Stanford University, KSL, USA

Cynthia Chang
Tetherless World Constellation, Rensselaer Polytechnic Institute (RPI), and Stanford University, KSL, USA

ABSTRACT

In this chapter, we introduce the concept of explanation for Semantic Web applications by providing motivation, description, and examples. We describe the Inference Web explanation toolkit that provides support for a broad range of explanation tasks ranging from explaining deductive reasoning, to information extraction, to hybrid integrated learning systems. We argue that an explanation solution such as the one we endorse is required if we are to realize the full potential of hybrid, distributed, intelligent Web agents that users can trust and use.

INTRODUCTION

Question answering on the Semantic Web (SW) typically includes more processing steps than database retrieval. Question answering can be viewed as an interactive process between a user and one or more intelligent software agents. Using queries, user preferences, and context, intelligent agents may locate, select and invoke services and, if necessary, compose these services to produce requested results. In other words, the web paradigm shifts from one where users mainly retrieve explicitly stated stored information to a paradigm where application results are answers to potentially complex questions that may require inferential capabilities in addition to information retrieval. Web applications with question answering capabilities may still use information retrieval techniques to locate answers, but they may also need to use additional semantics such as encoded term meanings to support additional methods of information access (such as targeted database queries or knowledge base queries) along with information manipulations (such as reasoning using theorem provers, or inductive or deductive methods). Examples of this new, more complex reality include the automatic composition of web services encoded in OWL-S or semi-automatic composition of services as provided by workflows. Ontology-enhanced search is another example of how Semantic Web technology can provide and is providing new directions for a category of "smart" search applications. Many other SW applications are emerging with a common theme of increasing knowledge and autonomy. This new context generates an additional requirement for effective use of SW applications by typical users: *applications must provide explanation capabilities showing how results were obtained*. Explanations are quickly becoming an essential component in establishing agent credibility (e.g., Glass et al, 2008) and result credibility (e.g., Del Rio and Pinheiro da Silva, 2007) by providing process transparency, thereby increasing user understanding of how results are derived. Explanations can also identify information sources used during the conclusion derivation process. In the context of the SW, explanations should be encoded in a way that they can be directly or indirectly consumed by multiple agents, including both human users and software systems.

In this chapter we describe explanation as a special kind of pervasive SW functionality, in the sense that a SW application may need to provide transparency concerning its results. We first analyze some distinct application paradigms in the SW context, and for each paradigm we identify explanation requirements. We then describe a general framework, called Inference Web (IW) (McGuinness and Pinheiro da Silva, 2004) that includes the Proof Markup Language (PML) (McGuinness, et al., 2007, Pinheiro da Silva, McGuinness, Fikes, 2006), a modularized ontology describing terms used to represent provenance, justifications and trust relations. IW includes a set of tools and methods for manipulating PML-encoded result justifications. Using Inference Web, and its PML interlingua, applications may provide interoperable and portable explanations that support intelligent, interactive application interfaces. After the description of the IW framework and the PML interlingua, we will exemplify how PML and IW have been used to explain the results and behaviors of a wide range of applications including intelligent personal agents, information extraction agents, and integrated learning agents.

A CONCEPTUAL FRAMEWORK FOR EXPLAINING RESULTS FROM SEMANTIC WEB APPLICATIONS

We investigate the correspondence between SW application paradigms and their explanation requirements.

Semantic Web Application Characterization

SW applications are geared to take advantage of vast amounts of heterogeneous data with potentially varying amounts of semantic markup. They concentrate on identifying and meaningfully combining available semantic markup in order to derive complex results. Below we briefly characterize the SW applications features considered important from an explanation perspective: collaboration, autonomy, and use of ontologies.

Collaboration

Collaboration requires agents to interact and share knowledge with the common goal of solving a particular problem. Collaboration raises issues concerning how to create, use, and share a combination of provenance, trust and reputation throughout distributed reasoning processes. Wikis, for example, are gaining popularity as collaborative tools for human agents, although they do not provide a precise infrastructure for recording and reusing provenance information. A *Semantic Wiki* is a wiki application enhanced with Semantic Web technologies that support wiki content annotation that goes beyond simple structured text and untyped hyperlinks. Semantic Wikis provide the ability to represent metadata about content, term meanings, and inter-relationships. Provenance support is typically somewhat limited, in both ordinary wikis and in semantic wikis, to keeping track of which author (if a login authentication process is included) made which updates and when.

Content Management Systems (CMS) are one of the most common uses of wikis for knowledge management. Semantic Wikis aim to enhance ordinary wikis by allowing users to make their internal knowledge more explicit and formal, enabling search methods that go beyond simple keyword search. In this case, provenance information may be included in these searching capabilities. Other collaborative systems are aimed at Personal Information Management (PIM) or community knowledge management. The ability to store project history, and to utilize tools that access and perform intelligent queries over this history, is one of the benefits brought by Semantic Wikis used for content management.

The collaborative characteristic is also prominent in applications developed via the integration of multi-agent systems and Semantic Web services. In this situation, collaborating agents are software programs such as digital assistants that manage electronic information. These collaborating agents can proactively engage in tasks on behalf of their users to find, filter, assess and present information to the user in a more appropriate manner (Maes, 1994). Several types of multi-agent applications have been developed such as office organization (Pyandath & Tambe, 2002); technical support (Sullivan et al. 2000); and information retrieval (Rhodes et al., 1996). Again, most of these collaborating agents provide little support for storing and retrieving provenance information about how they work internally, and in particular, they provide only limited access to information about how they collaborate. However, end user activities may require the integration of multi-agent systems and Semantic Web services. Personal agents may also need user models, to allow them to better perform tasks in compliance with user needs and preferences.

Distributed solutions for multi-agent problems can alternatively be represented using a reactive multi-agent architecture. In these domains, the individual agents have little autonomy. The "intelligence" used to solve problems comes from intensive inter-agent communication. This paradigm is typically used on the web, where heterogeneity and loosely-coupled distributed systems are common. Thus, interactions between agents or system components must not be rigidly specified at design time, but opportunistically built though the use of new services as they become available. Prior knowledge of such services is thus not necessary (and often not practical nor desirable). Instead, agents must discover services by accessing a *service description* that can be semantically described by means of ontologies in which descriptive expressions or concepts are attached to services.

Autonomy

An individual agent's autonomy controls its ability to act independently. Barber and Martin (1999) consider an agent's degree of autonomy with respect to a particular goal that the agent is actively pursuing. Within this context, they define the degree of autonomy to be (1) the degree to which the decision making process was used to determine how that goal should be pursued; and (2) how free the agent is from intervention by other agents. Traditional web-based applications have very little autonomy, since they primarily take direct input from the user and retrieve information consistent with the query. For example, a typical web search engine's primary interaction mechanism is based on communication between the user and the search engine. The degree of autonomy of the search engine is said to be low because the user is required to reformulate and resubmit the query when the original query is not satisfactorily answered by the engine. In contrast with typical search engines, SW applications have more autonomy while pursuing goals. For example, online shopping agents have autonomy over how to find answers to shopping queries concerned with product location, price comparison, or rating information. ShopBot can make several autonomous decisions, such as which content sources to use, which services to call and compose, and how to enhance the query with background representation information, all in an attempt to answer the user's question as efficiently and usefully as possible. In general, the development of autonomous problem-solving software agents in the Semantic Web is increasingly gaining popularity.

Use of Ontologies

Semantic Web applications are increasingly using large amounts of heterogeneous semantic data from multiple sources. Thus, the new generation of Semantic Web applications must be prepared to address issues associated with data of varying quality. Intelligence in these large-scale semantic systems comes largely from the system's ability to operate effectively with large amounts of disparate data.. In this context, ontologies are used to support information integration as well as to identify inconsistencies between data coming from multiple sources. Ontologies are being used to provide declarative specifications of term meanings. Agents can then decide to use a term meaning as specified in a particular ontology, and when multiple agents decide to use the same definition of a term (for example by referencing the same term in the same ontology), they can communicate more effectively. Usage of the same term, now with the same meaning, helps improve consistency across applications.

Content search and context search are other typical uses of ontologies. In content search, search engines use background knowledge bases to enhance queries and thus improve results. When the background knowledge bases contain term definitions, semantic query engines may be able to retrieve answers that are inferred by the query, no longer restricting the search to exact user-provided terms. Search engines can go beyond statistical clustering methods, which while effective, have limitations largely associated with training data sets. In context search, search engines may consider the user's context when processing a search. For example, a search engine may utilize a user's geographic location as well as known preferences when retrieving answers. Information about geographic location and preferences may be encoded in background ontologies.

Ontologies describing domain knowledge, user preferences, and problem areas are often used in creating agents with reasoning capabilities. These ontologies are often used to establish a common vocabulary among multiple agents. Personal agents' learning capabilities are also important, as such capabilities can increase the agents' level of autonomy (e.g., the Cognitive Assistant that Learns and Organizes (CALO, 2008). Personal agents can act alone or communicate with others in order to accomplish their task; in these cases, ontologies describing communications protocols are also necessary.

Explanation Issues

Given these Semantic Web application features which impact the need for explanation, we identify a set of criteria for analyzing the required explanations. These criteria include such issues as whether explanations are expected to be consumed by humans or machine agents; varying characteristics of these agents; and the resulting types of explanations that should be provided.

Explanation Types

System transparency allows users to see how answers are generated and how processes within and among agents have evolved to support answer generation. Transparency allows users to access lineage information that often appears hidden in the complex Semantic Web network. Note that explanations should be viewed as a web of interconnected objects recording source information, source assertions and assumptions, intermediate results, and final results instead of as a single "flat" annotation. Results from Semantic Web applications may be derived from a series of information manipulation steps, each of which applies a primitive information manipulation operation, e.g., an inference or extraction rule, on some antecedents and produces a conclusion. Note that an information manipulation step may be any kind of inference and is not limited to those that are used in sound and complete reasoners. Thus this representation can handle statistical methods, standard logical inference, or even non-logical information transformation methods. A justification may be viewed as a transaction log of 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 justification. These transaction logs may be quite detailed, so it is also important to be able to provide explanations that are abstractions of these logs.

Another kind of explanation can be obtained from provenance metadata that contains annotations concerning information sources, (e.g., when, from where, and by whom the data was obtained). Provenance metadata connects statements in a knowledge base to the statement sources such as web pages and publications, including annotations about data collection or extraction methods. Criticality of provenance is evident. Users demand detailed provenance metadata before they will accept and believe answers (e.g., Cowell, et al, 2006; Del Rio and Pinheiro da Silva, 2007). In some settings such where an initial evaluation of usefulness is made, provenance metadata (e.g., source, recency, and authoritativeness) is the only information that users need.

Trust in the Semantic Web is another subject of growing importance in the explanation context. 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 (e.g., McGuinness, et. al, 2006).

Human or Machine Consumption

Semantic Web applications typically require explanation for both human and machine consumption. Software agents require representation of justifications, provenance and trust in a standard format in order to enable interoperability. An interoperable justification specification can be used to generate explanations of an agent's reasoning process as well as of the sources used by the agent during the problem solving process. Explanations aimed at either humans or software agents can be generated from the internal justification, provenance, and trust representations. When the explanations are aimed at humans, the explanations must also include human computer interface (HCI) considerations. For instance, the display of an explanation may take into consideration the level of expertise of the user, e.g., expert or non-expert, as well as the context of the problem (e.g., Del Rio and Pinheiro da Silva, 2007a). HCI researchers have approached the explanation problem by proposing intelligent question-answering systems (e.g., Maybury, 2003), intelligent help systems (e.g., Lieberman and Kumar, 2005), and adaptive interfaces (e.g., Wagner and Lieberman, 2003).

Visualization Capabilities

Explanations can be viewed as Semantic Web metadata representing how results were obtained. In distributed settings such as the Web, representation interoperability is paramount. A variety of "user friendly" rendering and delivery modes are required to present information to different types of users in varying contexts. As explanations may need to be delivered to users with a variety of skill levels, visual representation must be 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 consider several possible presentation modes. Implemented interfaces for each of these views can be seen in McGuinness, et al, 2006.

Global View. The entire process of explanation may be presented via a graphical display of a justification graph. The idea is to provide a view of the global structure of the reasoning process used by a question answering system. Common issues include how portions of information composing the explanation will be presented (for example, whether they are displayed in an English translation of the justification encoding, or in the reasoner's native language); or whether to restrict the depth and width of the explanation graph (e.g., with using notions such as lens magnitude and width options in the Inference Web browser). A useful feature in these kinds of views is to provide clickable hot links to enable access to additional information.

Focused View. Merely providing tools for browsing 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, for instance, to focus on one step of the justification, and to display that step using a natural language template style for presentation. Further focus on explanations can be provided by suggested context-appropriate follow up questions.

Filtered View. Alternative options may also be chosen, such as seeing only the assertions (ground facts) upon which a given result depended; only the sources used for ground assertions; or only the assumptions upon which the result depended. Another possible view is the collection of sources contributing information used to derive the result. 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 the result. Initially, these users may not want to view all the details of the information manipulations (but they do want the option of asking follow-up questions when necessary).

Abstraction View. Machine-generated justifications are typically characterized by their complexity and richness of details that may not be relevant or interesting to most users. Filtering explanation information and providing only one type of information (for example, only showing the information sources) are some of the strategies used to deal with the large volume of data in justifications. These strategies translate the detailed explanation into a more abstract and understandable one.

In fact, this diversity of presentation styles is critical for broad acceptance of SW results. As we have interviewed users both in user studies (e.g., Cowell, et al, 2006; Del Rio and Pinheiro da Silva, 2007; Glass, et al., 2008) and in ad hoc requirements gathering, it was consistently true that broad user communities require focus on different types of explanation information and on different explanation formats. For any user segment that prefers a detailed trace-based view, there is another complementary and balancing user 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 a discourse-style presentation component.

Explanation Issues vs. Semantic Web Application Characteristics

Having independently considered facets of both complex Semantic Web contexts and requirements for successful explanations, we now address how these issues relate to each other, providing requirements for explaining a broader range of SW applications.

Explanation and Collaboration

Trust and reputation are important issues in the context of collaborative applications and have been studied in the context of traditional wikis like Wikipedia (e.g., McGuinness, Zeng et al., 2006). The advent of semantic wikis introduces new concerns and requirements in terms of explanation. Autonomy among SW agents is continuously increasing, and if users are expected to believe answers from these applications, SW applications must support explanations. This requirement becomes even more important when SW applications collaborate to generate complex results.

As personal agents mature and assume more autonomous control of their users' activities, it becomes more critical that these agents can explain the way they solve problems on behalf of humans. The agents must be able to tell the user why they are performing actions, what they are doing, and they must be able to do so in a trustable manner. Justifications and task processing explanations are essential to allow personal agents to achieve their acceptance goals. In addition, the learning skill presented by some personal agents amplifies the need for explanation since it introduces a degree of variability resulting from learning results. Justifications concerning agent's internal reasoning for learning new knowledge as well as explanations concerning usage of knowledge sources are examples of what must be explained. Distributed reasoning requires explanation capabilities to help users understanding the flow of information between the different agents involved in a problem solving process. These capabilities also allow users to understand the process taken by the distributed problem solvers. Additionally, provenance explanations are of interest since users might want to know information about each one of the learners and problem solvers used, as well as wanting to know information about each source of information that was used. Issues of trust and reputation are particularly likely to modify user's trust in agents' answers.

Explanation and Autonomy

In applications for which the degree of autonomy is low (for instance, a Google-based search query), no explicit explanation is provided. One could assume that aspects of explanatory material are implicitly embedded in the answers. In such settings, the user needs to have enough information to understand the context of the answers (e.g., the links selected by the query engine represent an information retrieval response to the query, and the answers include links to the sites containing the information). It is assumed that explaining why a search engine has selected a set of links is implicitly understood by the user (for instance, the search engine considers the provided answers to be the best responses, with some suitable definition of best which may rely on reverse citations, recency, etc.). The existence of a ranking mechanism is fundamental for the success of the interaction process because query reformulation depends on that ability. Understanding the process that led the search engine to provide an answer to a query facilitates the process of query refinement.

Even applications with low degrees of autonomy may experience demand from users for some forms of explanation. Users may want to know how a search engine got its answers, for example, if the answers were selected using certain purchased keywords or other advertising promotions, or if the answers depended on out-of-date source material. The information needs to be presented in an understandable manner, for instance, by displaying answers using purchased keywords in a different style.

Justifications become even more important in applications with higher degrees of autonomy. Autonomous agents can follow complex inference process, and justifications are an important tool for them to provide understandable information to end users.

Explanations and Ontologies

Ontologies can be used effectively to support explanations for a wide array of applications, ranging from relatively simple search applications to complex autonomous problem solving. For example, consider a contextual database search agent which considers user preferences when answering queries. Explanations of why a given solution was provided in a given context are particularly important when the solution does not match the user's specified preferences. Similarly, explanations are important when a particular contextual query results in different answers in different contexts (for example, when answers are dependent on the user's geographic location).

INFERENCE WEB: AN ONTOLOGY-ENHANCED INFRASTRUCTURE SUPPORTING EXPLANATIONS

We now explore Inference Web in the context of addressing the problem of providing explanations to justify the results and behaviors of Semantic Web services and applications. IW provides tools and infrastructure for building, maintaining, presenting, exchanging, combining, annotating, filtering, comparing, and rendering information manipulation traces, i.e., justifications. IW services are used by agents to publish justifications and explanations for their results that can be accessible digitally – on the web, on a local file system, or distributed across digital stores. Justification data and explanations derived from justifications are encoded using terms defined by the Proof Markup Language (PML) justification, provenance, and trust ontologies. The PML ontologies are specified in OWL and are easily integrated with Semantic Web applications. The ontologies include terms such as sources, inference rules, inference steps, and conclusions as explained later.

PML is an on-going, long-term effort with several goals and contributions to explaining Semantic Web application results and behaviors. Our earlier version of PML focused on explaining results generated by hybrid web-based reasoning systems, such as the question answering systems of DARPA's High Performance Knowledge Base program and its subsequent Rapid Knowledge Formation program. The requirements obtained for this initial explanation phase were similar to explanation requirements gathered for expert systems where knowledge bases were generated from reliable source information and using trained experts. Information in these systems was assumed to be reliable and recent. Thus, agent users only needed explanations about information manipulation steps, i.e. how the results were derived in a step by step manner from the original knowledge base via inference. In this setting, explanations concerning information sources used to derive results were not required.

As automated systems become more hybrid and include more diverse components, more information sources are used and thus users are seldom in a position to assume that all information is reliable and current. In addition to information manipulation, users may need explanations about information provenance. Under certain circumstances, such as intelligence settings that motivated DTO's Novel Intelligence for Massive Data program, provenance concerns often dwarfed all others when explanations were required (Cowell, et. al., 2006).

As automated systems begin to exploit more collaborative settings and input may come from many unknown authoring sources, notions of trust and reputation may become more critical. Meta information may be associated with authoring sources such as "I trust Joe's recommendations" or "I trust population data in the CIA World Factbook"). In these situations the meta-information may be user authored. In other settings, trust or reputation information may be calculated using techniques such as link analysis or revision analysis (Zeng, et.al. 2006).

Our goal is to go beyond explanation for traditional knowledge-based systems, and instead address explanation needs in a wide range of situations. We have settings where three different aspects of explanation sometimes dominate to the point that the other aspects are of secondary consideration. We thus took on a rationalization and redesign of our original representation Interlingua so that it could be modular. We can now support applications that only desire to focus on provenance (initially or permanently ignoring issues related to information manipulation and trust.). While these applications may later expand to include those concerns, they need not import ontologies with terms defined for those situations.

Using PML

To illustrate how PML supports explanation generation, we use a simple wine agent scenario. While this example is intentionally oversimplified, it does contain the question answering and explanation requirements in much more complicated examples. We have implemented a wine agent (Hsu, McGuinness, 2003) that suggests descriptions of wines to go with foods. The agent uses PML as its explanation interlingua, and a theorem prover capable of understanding and reasoning with OWL and outputting PML (Fikes, et. al., 2003)). The agent is capable of making wine recommendations to coordinate with meal courses (such as "Tony's specialty"). Before customers choose to follow the agent's recommendation, they may be interested in knowing a description of Tony's specialty, so that they can evaluate if the suggested wine pairing meets their desires. In this scenario, they would find that Tony's specialty is a shellfish dish and the wine agent suggests some white wines as potential matches. The user may want to know how the description of the matching wine was produced, and if the wine agent used other sources of information, such as commercial online wine web sites or hand built backend databases.

In some intelligence settings, e.g., (Cowell, et. al., 2006, Murdock, et. al., 2006), users often want to ask questions about what sources were relied on to obtain an answer. In some military settings, e.g., (Myers, et. al., 2007), users often want to ask what the system is doing, why it has not completed something, and what learned information was leveraged to obtain an answer. In other settings, such as collaborative social networks, users may be interested in either reputation as calculated by populations or trust as stated and stored by users, e.g., (McGuinness, et. al., 2006b). These setting are further elaborated in the following section.

Our PML explanation ontologies include primitive concepts and relations for representing knowledge provenance. Our original version of PML (Pinheiro da Silva et al., 2003) provided a single integrated ontology for use in representing information manipulation activities, the extended version of PML (called PML 2) improves the original version by modularizing the ontologies and refining and expanding the ontology vocabulary. This also broadens the reach covering a wider spectrum of applications for the intelligence, defense, and scientific communities. The modularization serves to separate descriptive metadata from the association metadata to reduce the cost of maintaining and using each module. The vocabulary expansion refines the definition and description structure of existing PML concepts; and it also adds several new primitive concepts to enrich expressiveness. For example, instead of simply serializing a piece of information into a text string, PML uses the concept of information as the universal reference to any piece of data, and enables explicit annotation (for instance, of format, language, and character encoding) about the string that serializes the piece of information.

PML provides vocabulary for three types of explanation metadata:

- The provenance ontology (also known as PML-P) focuses on annotating identified-things (and in particular, sources such as organization, person, agent, services) useful for providing lineage.
- The justification ontology (also known as PML-J) focuses on explaining dependencies among identified-things including how one identified-thing (e.g., information) is derived from other identified-things (e.g. information, services, agents).
- The trust relation ontology (also known as PML-T) focuses on representing and explaining belief assertions.

Provenance Ontology

The goal of the provenance ontology (also called PML-P[a]) is to annotate the provenance of information, e.g., which sources were used, who encoded the information, etc. The foundational concept in PML-P is *IdentifiedThing*. An instance of IdentifiedThing refers to an entity in the real world, and its properties annotate its metadata such as name, description, creation date-time, authors, and owner. PML-P includes two key subclasses of IdentifiedThing motivated by knowledge provenance representational concerns: *Information* and *Source*.

The concept Information supports references to information at various levels of granularity and structure. It can be used to encode, for example, a formula in logical languages or a natural language text string. PML-P users can simply use the value of information's *hasRawString* property to store and access the content of the referred information as a string. They may optionally annotate additional processing and presentation instructions using PML-P properties such as *hasLanguage*, *hasFormat*, *hasReferenceUsage* and *hasPrettyNameMappingList*. Besides providing representational primitives for use in encoding information content as a string, PML-P also includes primitives supporting access to externally referenced content via *hasUrl*, which links to an online document, or *hasInfoSourceUsage*, which records when, where and by whom the information was obtained. This concept allows users to assign an URI reference to information. The example below shows that the content of a piece of information (identified by

#info1) is encoded in the Knowledge Interchange Format (KIF) language and is formatted as a text string. The second example below shows that the content of information (identified by #info_doc1) can be indirectly obtained from the specified URL, which also is written in KIF language.

```
<pmlp:Information rdf:about="#info1">
  <pmlp:hasRawString>(type TonysSpecialty SHELLFISH)
    h</pmlp:hasRawString>
  <pmlp:hasLanguage rdf:resource= "http://inferenceweb.stanford.edu/registry/LG/KIF.owl#KIF" />
  <pmlp:hasFormat>text</pmlp:hasFormat>
</pmlp:Information>
```

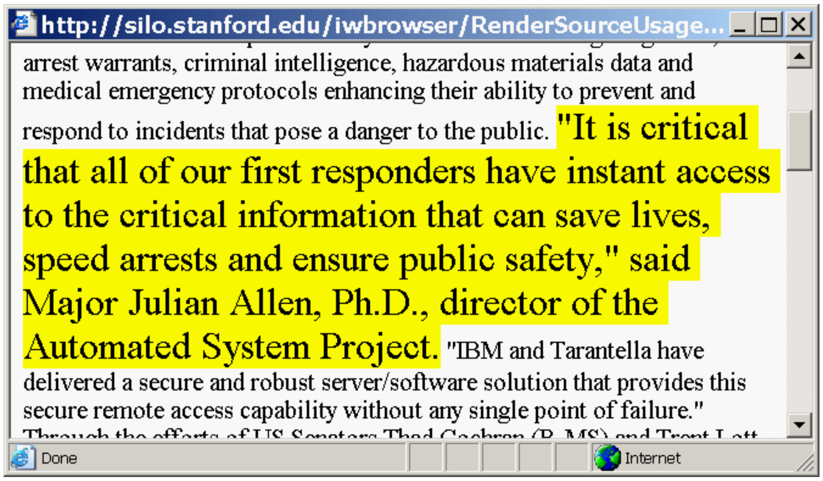
```
<pmlp:Information rdf:about="#info_doc1">
  <pmlp:hasURL>http://iw.stanford.edu/ksl/registry/storage/documents/tonys_fact.kif</pmlp:hasURL>
  <pmlp:hasLanguage rdf:resource= "http://inferenceweb.stanford.edu/registry/LG/KIF.owl#KIF" />
</pmlp:Information>
```

The concept source refers to an information container, and it is often used to refer to all the information from the container. A source could be a document, an agent, or a web page, and PML-P provides a simple but extensible taxonomy of sources. The Inference Web Registry (McGuinness and Pinheiro da Silva, 2003) provides a public repository for registered users to pre-register metadata about sources so as to better reuse such metadata. Our current approach, however, does not demand a centralized or virtual distributed registry; rather, it depends on a search component that finds online PML data and provides search service for users' inquiry.

```
<pmlp:Document rdf:about="#STE">
  <pmlp:hasContent rdf:resource="#info_doc1"/>
</pmlp:Document>
```

In particular, PML-P provides options for encoding finer grained references to a span of a text through its *DocumentFragmentByOffset* concept.

Figure 1. Raw text fragment with highlighted segment used by text analytics components and represented in PML 2



arrest warrants, criminal intelligence, hazardous materials data and medical emergency protocols enhancing their ability to prevent and respond to incidents that pose a danger to the public. "It is critical that all of our first responders have instant access to the critical information that can save lives, speed arrests and ensure public safety," said Major Julian Allen, Ph.D., director of the Automated System Project. "IBM and Tarantella have delivered a secure and robust server/software solution that provides this secure remote access capability without any single point of failure." Through the efforts of US Senators Thad Cochran (R-MS) and Trent Lott

This is a sub-class of Source and *DocumentFragment*. The example below shows how the offset information about #ST can be used to highlight the corresponding span of text (see Figure 1). This type of encoding was used extensively in our applications that used text analytic components to generate structured text from unstructured input as explained below.

```
    <pmlp:DocumentFragmentByOffset rdf:
about="#ST">
        <pmlp:hasDocument rdf:
resource="#STE"/>
      <pmlp:hasFromOffset>62</pmlp:has-
FromOffset>
           <pmlp:hasToOff-
set>92</pmlp:hasToOffset>
</pmlp:DocumentFragmentByOffset>
```

As our work evolved, a number of our applications demanded more focus on provenance. We became increasingly aware of the importance of capturing information about the dependency between information and sources, i.e. when and how a piece of information was obtained from a source. PML 2 has a more sophisticated notion of *SourceUsage*. The encoding below simply shows how PML represents date information identifying when a source identified by #ST was used.

```
    <pmlp:SourceUsage rdf:
about="#usage1">
      <pmlp:hasUsageDateTime>2005-10-
17T10:30:00Z</pmlp:hasUsageDateTime>
     <pmlp:hasSource rdf:resource="#ST"/>
    </pmlp:SourceUsage>
```

Besides the above concepts, PML-P also defines concepts such as *Language*, *InferenceRule*, and *PrettyNameMapping*, which are used to represent metadata for application processing or presentation instructions.

Justification Ontology

The goal of the justification ontology is to provide concepts and relations used to encode traces of process executions used to derive a conclusion. A justification requires concepts for representing conclusions, and information manipulation steps used to transform/derive conclusions from other conclusions, e.g., step antecedents.

A *NodeSet* includes structure for representing a conclusion and a set of alternative information manipulation steps also called *InferenceSteps.* Each InferenceStep associated with a NodeSet provides an alternative justification for the NodeSet's conclusion. The term NodeSet is chosen because it captures the notion that the NodeSet concept can be used to encode a set of nodes from one or many proof trees deriving the same conclusion. The URI of a NodeSet is its unique identifier, and every NodeSet has exactly one URI.

The term inference in InferenceStep refers to a generalized information manipulation step, so it could be a standard logical step of inference, an information extraction step, a simple computation process step, or an assertion of a fact or assumption. It could also be a complex process such as a web service or application functionality that may not necessarily be describable in terms of more atomic processes. InferenceStep properties include *hasInferenceEngine* (the agent who ran this step), *hasInferenceRule* (the operation taken in this step), *hasSourceUsage*, *hasAntecedentList* (the input of this step), and others.

PML2 supports encodings for several typical types of justifications for a conclusion. Three justification examples are as follows:

An unproved conclusion or goal. A NodeSet without any InferenceStep can be explained as an inference goal that still needs to be proved. Unproved conclusions happen when input information encoded in PML2 is provided to an agent.

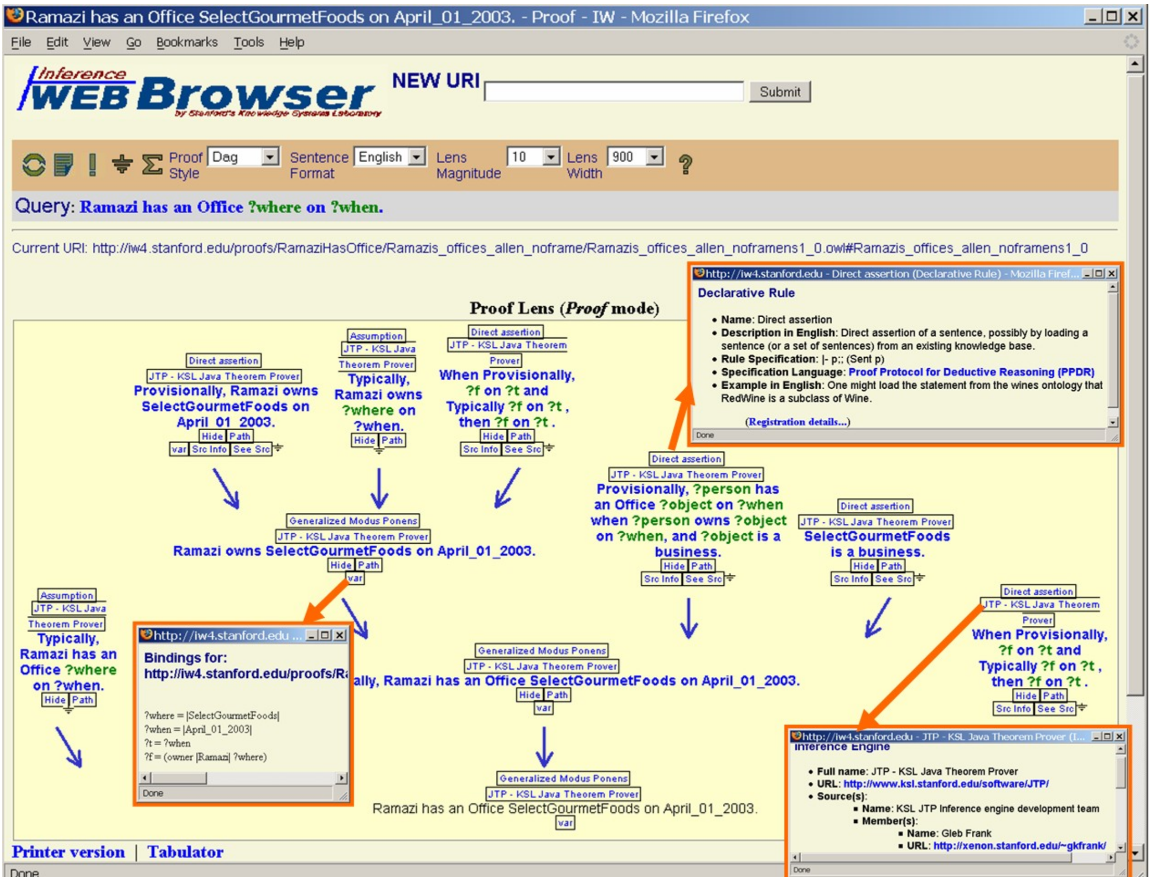
```
   <pmlj:NodeSet rdf:about=”#answer1”>
<pmlp:hasConclusion rdf:resource=“#info1”/>
 </pmlp:hasConclusion>
   </pmlj:NodeSet>
```

Assumption. The conclusion was directly asserted by an agent as an assumption. In this case, the conclusion is asserted by a source instead of being derived from antecedent information.

Direct assertion. The conclusion can be directly asserted by the inference engine. In this case, the conclusion is not derived from any antecedent information. Moreover, direct assertion allows agents to specify source usage. The following example shows that “‘(type TonysSpecialty SHELLFISH)’ has been directly asserted in Stanford’s Tony’s Specialty Example as a span of text between byte offset 62 and byte offset 92 as of 10:30 on 2005-10-17”

```
   <pmlj:NodeSet rdf:about=”#answer2”>
 <pmlp:hasConclusion rdf:resource=”#info1”
/>
    <pmlp:isConsequentOf>
    <pmlp:InferenceStep rdf:about=”step2”>
   <pmlp:hasInferenceEngine rdf:resource=
“http://inferenceweb.stanford.edu/registry/IE/
JTP.owl#JTP” />
    <pmlp:hasInferenceRule rdf:resource=
“http://inferenceweb.stanford.edu/registry/
```

Figure 2. Trace-oriented explanation with several follow-up question panes



Ramazi has an Office SelectGourmetFoods on April_01_2003. - Proof - IW - Mozilla Firefox

Proof Style: Dag
Sentence Format: English
Lens Magnitude: 10
Lens Width: 900

Query: Ramazi has an Office ?where on ?when.

Current URI: http://iw4.stanford.edu/proofs/RamaziHasOffice/Ramazis_offices_allen_noframe/Ramazis_offices_allen_noframens1_0.owl#Ramazis_offices_allen_noframens1_0

Proof Lens (*Proof* mode)

Direct assertion / JTP - KSL Java Theorem Prover: Provisionally, Ramazi owns SelectGourmetFoods on April_01_2003.

Assumption / JTP - KSL Java Theorem Prover: Typically, Ramazi owns ?where on ?when.

Direct assertion / JTP - KSL Java Theorem Prover: When Provisionally, ?f on ?t and Typically ?f on ?t , then ?f on ?t .

Generalized Modus Ponens / JTP - KSL Java Theorem Prover: Ramazi owns SelectGourmetFoods on April_01_2003.

Direct assertion / JTP - KSL Java Theorem Prover: Provisionally, ?person has an Office ?object on ?when when ?person owns ?object on ?when, and ?object is a business.

Direct assertion / JTP - KSL Java Theorem Prover: SelectGourmetFoods is a business.

Direct assertion / JTP - KSL Java Theorem Prover: When Provisionally, ?f on ?t and Typically ?f on ?t , then ?f on ?t .

Assumption / JTP - KSL Java Theorem Prover: Typically, Ramazi has an Office ?where on ?when.

Generalized Modus Ponens / JTP - KSL Java Theorem Prover: ...ally, Ramazi has an Office SelectGourmetFoods on April_01_2003.

Generalized Modus Ponens / JTP - KSL Java Theorem Prover: Ramazi has an Office SelectGourmetFoods on April_01_2003.

http://iw4.stanford.edu - Direct assertion (Declarative Rule) - Mozilla Firef...

Declarative Rule

- **Name**: Direct assertion
- **Description in English**: Direct assertion of a sentence, possibly by loading a sentence (or a set of sentences) from an existing knowledge base.
- **Rule Specification**: |- p;; (Sent p)
- **Specification Language**: Proof Protocol for Deductive Reasoning (PPDR)
- **Example in English**: One might load the statement from the wines ontology that RedWine is a subclass of Wine.

(Registration details...)

http://iw4.stanford.edu ...

Bindings for:
http://iw4.stanford.edu/proofs/Ra

?where = |SelectGourmetFoods|
?when = |April_01_2003|
?t = ?when
?f = (owner |Ramazi| ?where)

http://iw4.stanford.edu - JTP - KSL Java Theorem Prover (I...

Inference Engine

- **Full name**: JTP - KSL Java Theorem Prover
- **URL**: http://www.ksl.stanford.edu/software/JTP/
- **Source(s)**:
  - **Name**: KSL JTP Inference engine development team
  - **Member(s)**:
    - **Name**: Gleb Frank
    - **URL**: http://xenon.stanford.edu/~gkfrank/

```
DPR/ToId.owl#ToId" />
      <pmlp:hasSourceUsage rdf:
resource="#usage1" />
    </pmlp:InferenceStep>
    </pmlp:isConsequentOf>
   </pmlj:NodeSet>
```

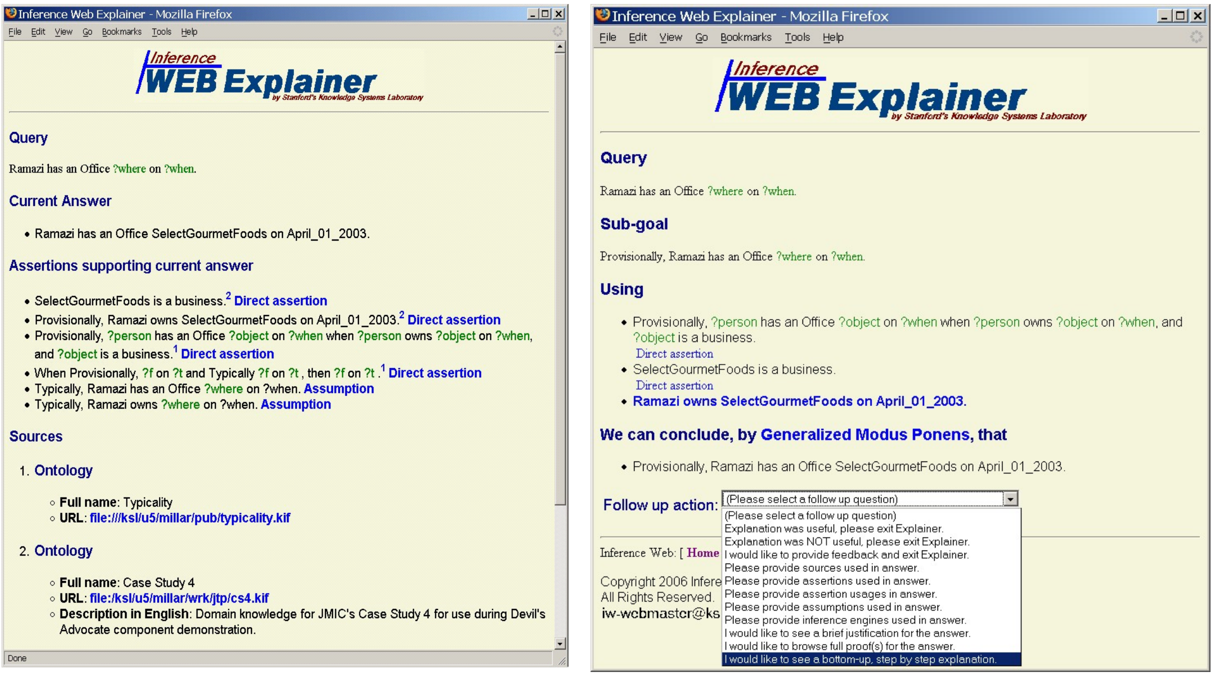
TOOLS FOR MANIPULATING EXPLANATION IN PML

To address the need to support multiple visualization modes for explanation, Inference Web provides rich presentation options for browsing 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.

Global View. Figure 2 depicts a screen shot from the IW browser in which the *Dag* proof style has been selected to show the global structure of the reasoning process. The sentence format 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. In Figure 3a, 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

Figure 3. (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


Inference Web Explainer - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Inference
WEB Explainer
by Stanford's Knowledge Systems Laboratory
Done
Inference Web Explainer - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Inference
WEB Explainer
by Stanford's Knowledge Systems Laboratory

Query

Ramazi has an Office ?where on ?when.

Current Answer

- Ramazi has an Office SelectGourmetFoods on April_01_2003.

Assertions supporting current answer

- SelectGourmetFoods is a business.[2] Direct assertion
- Provisionally, Ramazi owns SelectGourmetFoods on April_01_2003.[2] Direct assertion
- Provisionally, ?person has an Office ?object on ?when when ?person owns ?object on ?when, and ?object is a business.[1] Direct assertion
- When Provisionally, ?f on ?t and Typically ?f on ?t , then ?f on ?t .[1] Direct assertion
- Typically, Ramazi has an Office ?where on ?when. Assumption
- Typically, Ramazi owns ?where on ?when. Assumption

Sources

1. Ontology
   - **Full name**: Typicality
   - **URL**: file:///ksl/u5/millar/pub/typicality.kif
2. Ontology
   - **Full name**: Case Study 4
   - **URL**: file:/ksl/u5/millar/wrk/jtp/cs4.kif
   - **Description in English**: Domain knowledge for JMIC's Case Study 4 for use during Devil's Advocate component demonstration.

Query

Ramazi has an Office ?where on ?when.

Sub-goal

Provisionally, Ramazi has an Office ?where on ?when.

Using

- Provisionally, ?person has an Office ?object on ?when when ?person owns ?object on ?when, and ?object is a business.
  Direct assertion
- SelectGourmetFoods is a business.
  Direct assertion
- Ramazi owns SelectGourmetFoods on April_01_2003.

We can conclude, by Generalized Modus Ponens, that

- Provisionally, Ramazi has an Office SelectGourmetFoods on April_01_2003.

Follow up action: (Please select a follow up question)

(Please select a follow up question)
Explanation was useful, please exit Explainer.
Explanation was NOT useful, please exit Explainer.
I would like to provide feedback and exit Explainer.
Please provide sources used in answer.
Please provide assertions used in answer.
Please provide assertion usages in answer.
Please provide assumptions used in answer.
Please provide inference engines used in answer.
I would like to see a brief justification for the answer.
I would like to browse full proof(s) for the answer.
I would like to see a bottom-up, step by step explanation.

Inference Web: [Home

Copyright 2006 Infere
All Rights Reserved.
iw-wcbmaster@ks

of context-dependent follow-up questions.

Filtered View. Figure 3b is the result of the user asking to see the sources.

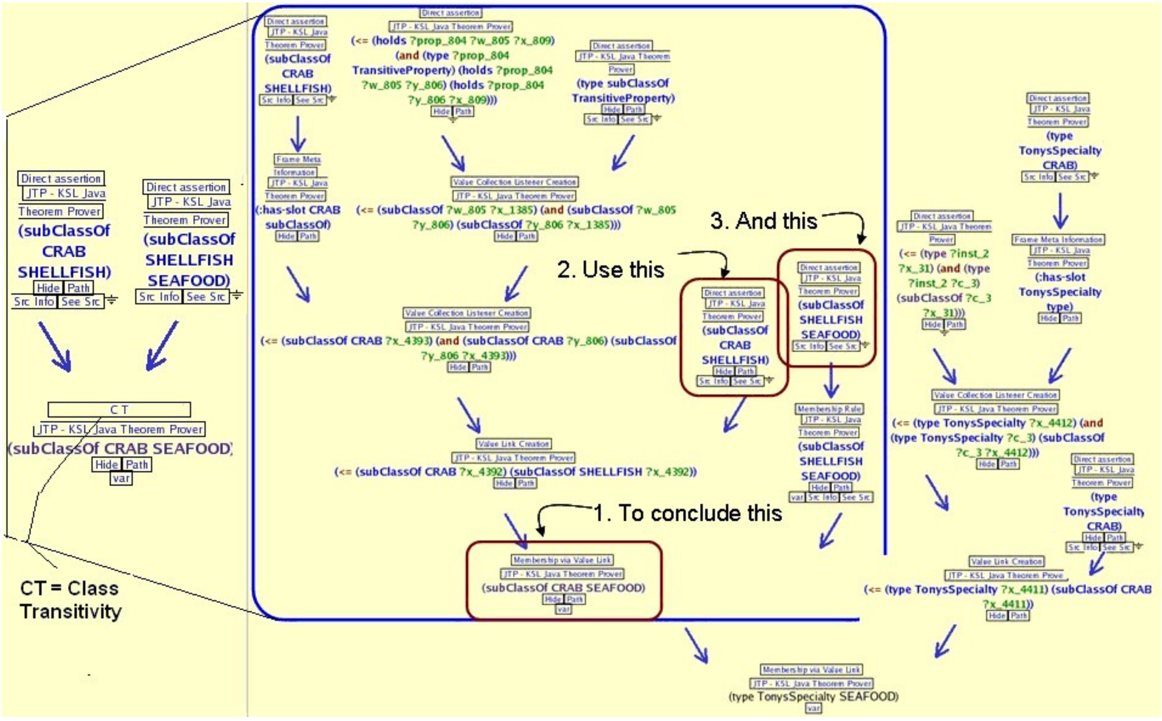
Abstraction View. Inference Web approaches this issue with two strategies:

- 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 4. 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 in the middle of Figure 4 (inside the blue round-angled box). The user may specify an abstraction rule to reduce the multi-step proof fragment into a one-step proof fragment (class-transitivity inference) on the left side of Figure 4.

We are building up abstraction patterns for domain independent use, e.g. class transitivity as

Figure 4. Example of an abstraction of a piece of a proof


Direct assertion
JTP - KSL Java Theorem Prover
(subClassOf CRAB SHELLFISH)
Direct assertion
JTP - KSL Java Theorem Prover
(subClassOf SHELLFISH SEAFOOD)
Hide
Path
Src Info
See Src
C T
JTP - KSL Java Theorem Prover
(subClassOf CRAB SEAFOOD)
var
CT = Class Transitivity
Direct assertion
(<= (holds ?prop_804 ?w_805 ?x_809) (and (type ?prop_804 TransitiveProperty) (holds ?prop_804 ?w_805 ?y_806) (holds ?prop_804 ?y_806 ?x_809)))
(type subClassOf TransitiveProperty)
Frame Meta Information
(:has-slot CRAB subClassOf)
Value Collection Listener Creation
(<= (subClassOf ?w_805 ?x_1385) (and (subClassOf ?w_805 ?y_806) (subClassOf ?y_806 ?x_1385)))
(<= (subClassOf CRAB ?x_4393) (and (subClassOf CRAB ?y_806) (subClassOf ?y_806 ?x_4393)))
Value Link Creation
(<= (subClassOf CRAB ?x_4392) (subClassOf SHELLFISH ?x_4392))
3. And this
2. Use this
1. To conclude this
Membership Rule
(subClassOf SHELLFISH SEAFOOD)
Membership via Value Link
(subClassOf CRAB SEAFOOD)
(<= (type ?inst_2 ?x_31) (and (type ?inst_2 ?c_3) (subClassOf ?c_3 ?x_31)))
(type TonysSpecialty CRAB)
(:has-slot TonysSpecialty type)
(<= (type TonysSpecialty ?x_4412) (and (type TonysSpecialty ?c_3) (subClassOf ?c_3 ?x_4412)))
JTP - KSL Java Theorem Prove
(<= (type TonysSpecialty ?x_4411) (subClassOf CRAB ?x_4411))
(type TonysSpecialty SEAFOOD)

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 to users than others. Additionally, the context and user model 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, IW provides a *discourse view*, which selects trace portions and presents them in simple natural language sentences. In this interaction mode, 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.

CASE STUDIES: PML IN ACTION

We will describe four applications that are using the IW framework and PML for explaining semantic information and behavior. We selected four applications that can be categorized differently following the conceptual framework.

Cognitive Personal Assistants: CALO Example

IW and PML have been used by a DARPA-sponsored cognitive agent system called CALO that can be told what to do, reason with available knowledge, learn from experience, explain its recommendations, and respond robustly to surprise. The cognitive agent's actions are supported by justifications that are used to derive and present understandable explanations to end-users. These justifications reflect both how the actions support various user goals, and how the particular actions chosen by the agent were guided by the state of the world. More specifically, our approach to PML task justification breaks down the justification of a question about a particular task *T* into three complementary strategies, described here using terminology from SPARK (Morley & Myers 2004), the task engine used by CALO:

- **Relevance:** Demonstrate that fulfilling *T* will further one of the agent's high-level goals, which the user already knows about and accepts
- **Applicability:** Demonstrate that the conditions necessary to start *T* were met at the time *T* started (possibly including the conditions that led *T* to be preferred over alternative tasks)
- **Termination:** Demonstrate whether one or more of the conditions necessary to terminate *T* has not been met.

This three-strategy approach contrasts with previous approaches to explanation, most of which dealt with explaining inference (Scott et al. 1984, Wick & Thompson 1992). Previous approaches generally have not dealt with termination issues, and they also generally have not distinguished between relevance and applicability conditions. These are critical aspects of task processing and thus are important new issues for explanation.

Behavior Justification in PML

In CALO context, PML documents contain encodings of *behavior justifications* using PML node sets. A task execution justification is always a justification of why an agent is executing a given task *T*. The final conclusion of the justification is a sentence in first order logic saying that *T* is currently being executed. There are three antecedents for this final conclusion, corresponding to the three strategies discussed above. Each antecedent is supported by a justification fragment based on additional introspective predicates.

It is important to note that all the task processing justifications share a common structure that is rich enough to encode provenance information needed to answer the explanation requests we have identified so far. By inspecting the execution state via introspective predicates, explanation components can gather enough provenance information to support a wide range of explanations.

Text Analytic Information Manipulations: KANI Example

KANI (Knowledge Associates for Novel Intelligence) (Welty, et. al., 2005, Murdock, et. al., 2006) is a DTO-sponsored intelligence analyst hybrid system that combines large scale information extraction with knowledge representation. In this section we focus on the relevance of provenance to support explanations of hybrid systems utilizing statistical and deductive inference.

In this setting, we can view all information manipulation steps in a PML justification as a kind of inference. We then generated a taxonomy of text analytic processes and tasks that can be viewed as inferences. The taxonomy was motivated by the need to describe and explain the dominant extraction tasks in UIMA[b], without overloading the system with more information than would be useful. One key was to generate a taxonomy that is adequate to accurately describe extraction task functionalities and simultaneously abstract enough to be able to hide details of the tasks from end users. Another key was to support explanations to end users of the integrated system, not authors of software components debugging their products.

We divided text extraction into three primitive areas: annotation, co-reference, and integration. We describe each briefly. Annotation tasks make assertions about spans of text that recognize a type or argument. Annotation inferences include:

1. **Entity recognition:** Determines that some span of text refers to an entity of a specified type. For example, a component could take the sentence "Tony Gradgrind is the owner of Tony's Foods" (the restaurant serving Tony's Specialty) and conclude that characters 0 to 14 of that sentence refer to some entity of type Person.
2. **Relation recognition:** Assigns a relation type to a span (e.g., a sentence describes a relation of type Owner).
3. **Relation annotation argument identification:** Determines and assigns values to the roles of a relation (e.g., a particular person is a participant in a given ownership relation instance).

Co-reference inferences utilize annotation inferences and further identify that multiple text spans actually refer to the same entity or relation.

1. **Entity identification:** Determines that a set of entity annotations refer to a particular instance.
2. **Relation identification:** Determines that a set of relation annotations refer to a particular relation instance.
3. **Extracted entity classification:** Determines that a particular co-referenced entity has a particular type. (e.g., the type of the entity referred to by "Gradgrind" is Person).
4. **Knowledge integration** inferences include mapping inferences providing access to provenance.
5. **Entity mapping:** Determines that an entity instance in the KB is derived from a set of entities and relation instances.

6. **Relation mapping:** Determines that a relationship in the target KB is derived from a set of entity and relation instances.
7. **Target entity classification:** Determines that an entity instance is an instance of an entity type in the target ontology.

We have registered these inferences in the IW registry and we use these information manipulation steps to explain all of the UIMA components used in our prototype system, which provides intelligence analyst support for analyzing documents and evaluating results of text statements.

Text Analytic Manipulation Descriptions

We use our taxonomy of text analytic manipulations in declarative descriptions encoding what was done to generate the extracted knowledge bases. UIMA generates a large extracted knowledge database containing its conclusions. We needed to take that as input (potentially augmented) and generate interoperable proof descriptions (a PML document) as an output.

The software component that produces PML documents for UIMA-based analysis processes begins with a specified result from a specified Extended Knowledge Database (EKDB) (e.g., TonyGradgrind is the Owner of TonysFoods). It follows the links in the EKDB from that conclusion back to the intermediate results and raw input that led to it. From these intermediate results, it is able to produce inference steps encoded in PML that refer to the corresponding tasks in the taxonomy. For example, if the EKDB records that characters 0 to 14 of some sentence were labeled as a Person and that this labeling was identified as specifying an occurrence of TonyGradgrind then the component would create an Entity Recognition inference step in PML for that labeling as well as coreference step for the result that the labeling is an occurrence of TonyGradgrind.

Transparent Accountable Data Mining: TAMI Example

TAMI (Weitzner, et. al., 2006) is an NSF-sponsored privacy-preserving system funded in the Cybertrust program. The idea is to provide transparency into the usage of data that has been collected, so that people may be able to see how data that has been collected about them has been used. In any accountable system, explanations are essential for providing transparency into the usage of information along with claims of compliance with privacy policies.

Usage policies are encoded concerning which organizations can use information for particular purposes. (The project specifically aims at usage instead of collection policies, so it is only use and reuse that is a topic for explanations). A transaction log is collected, which encodes data transfer information concerning transfers, policies, purposes, and organizations. Reasoning engines are used that evaluate the validity of transfer actions based on the encoded policies. These engines are instrumented to encode justifications for their determinations in PML, so that explanations can be provided about justified or unjustified transfers.

This system can be leveraged in a number of examples. One use case is in the explanation of justified or unjustified arrests. It is possible that data collected in compliance with rules for a particular purpose by an authorized agency may be reused to support a number of other conclusions. One prototype demonstration system in TAMI looks at arrests and then checks to see if they are justified according to their appropriate or inappropriate reuse of data that has been collected. Inference Web can then be used to explain why the system has determined that an arrest is legally justified or unjustified.

Integrated Learning Systems: GILA Example

GILA (Generalized Integrated Learning Architecture) is a DARPA-sponsored intelligent agent that integrates the results of multiple learners to provide intelligent assistant services. The initial domain is airspace control order deconfliction. GILA uses multiple independent learning components, a meta reasoning executive, and other components to make recommendations about ways to resolve conflicts in an existing airspace control order. In order to be operational, it must be able to explain its recommendations to end users and auditors. In addition, the explanations may be uses by learners and the meta executive to choose appropriate recommendations and assign credit and blame.

DISCUSSION

Explanation has been an active line of research since at least the days of expert systems, where explanation research largely focused on explaining rule-based systems. Today, explanation in rule systems is once again a research. Rule systems are now being integrated into hybrid settings, and now explanation must be done on both the rule components and the setting in which conclusions from those rule components are integrated and used. Also, theorem proving systems, such as Description Logic Reasoners, historically integrated explanation capabilities after usage increased and broadened. Early description logics that were broadly used, such as CLASSIC and LOOM provided some notion of explanation (e.g., McGuinness, 1996) in either insight into a trace or a proof theoretic-based approach to explanation. More recent explanation demands have inspired current generation tableaux-based DL reasoners to include some notion of explanation focusing on provenance, axiom usage, and clash detection (e.g., Parsia, et al, 2005, Plessers and Troyer, 2006). While all of these efforts are useful and important, today's explanation systems need to handle a much broader range of question answering styles and thus demand much more versatility and interoperability for their explanation infrastructure. Simultaneously, the infrastructure needs to be modular so that users with limited scope can support their applications without the burden of extra (unwanted) overhead. In our research on explaining provenance, we have recently modularized our explanation interlingua and the supporting background ontologies so that clients *only* interested in explaining provenance may use our infrastructure with the freedom of importing only the required modules.

Explanation requirements often arise in many settings that do not simply use standard deductive reasoning components. Our work, for example, has taken us into the realm of explaining text analytic components and a wide range of machine learning components. As a result, we have explored and are continuing to explore representation, manipulation, and presentation support for explaining systems that may use statistical, incomplete, and/or uncertain reasoning paradigms. Explanation research has also branched out into settings such as collaborative social networks, and we have engaged in research aimed particularly at explaining systems embedded in or leveraging large distributed communities. In many of the more recent research areas, we have found many requirements concerning trust, ranging from trust calculation to trust propagation, as well as presentation issues related to filtering by trust.

One relatively active area of provenance explanation is in the field of scientific applications. Increasingly, virtual collections of scientific data are being enabled by semantic technology (e.g., Virtual Observatories such as the Virtual Solar Terrestrial Observatory (McGuinness, et al, 2007). Such repositories are much more likely to be usable and to be used when provenance is maintained and available concerning where the data came from. More recently, there has been emphasis on additionally explaining the workflow from which it was produced. Thus, there is an emerging emphasis on explaining scientific provenance and workflow.

FUTURE RESEARCH DIRECTIONS

We have active research plans in a number of areas related to explanation.

1. **Learning.** Increasingly hybrid systems are depending on individual or multiple learning components to provide either ground facts or sometimes procedures. We are currently working multiple learning component authors to provide explanation components for learned information and learned procedures.
2. **Provenance.** The importance of provenance seems to be growing in many fields and we are focusing on providing relatively lightweight explanation solutions for provenance. We are also exploring special purpose needs of interdisciplinary scientific applications with respect to provenance.
3. **Trust.** Our current trust model is relatively simplistic and we are investigating ways of providing more representational primitives, methods for automatically suggesting trust ratings, and methods for intelligently combining and explaining combined trust values.
4. **Evaluation.** We have developed a PML validator that checks to see if an encoding is valid PML. We are extending that to provide an ontology evaluation module that not only checks for syntactic and semantic correctness, but also reviews (and explains findings concerning) ontology modeling styles.

CONCLUSION

In this chapter, we have explored the growing field of explanation. We noted that as applications become more autonomous, complex, collaborative, and interconnected, the need for explanation expands. We presented a modular interlingua capable of representing explanations that focus on provenance, justifications, and trust. We also presented the Inference Web infrastructure for manipulating explanations in a wide range of application settings. We provided examples in a diverse set of domains showing different settings where explanations are required, and then described how Inference Web and PML are being used to meet these needs. We also presented a number of different presentation paradigms for explanations.

ACKNOWLEDGMENT

We have benefited greatly by working with a number of excellent collaborators including Bill Murdock, Chris Welty, and Dave Ferrucci from IBM and Andrew Cowell, Dave Thurman, and colleagues from Battelle on NIMD, Michael Wolverton, Karen Myers, David Morley from SRI on CALO, Danny Weitzner, Tim Berners-Lee, Lalana Kagal, Chris Hanson, Gerry Sussman, Hal Abelson, Dan Connolly, Sandro Hawke, Kay Waterman, and colleagues from MIT on TAMI, and a large contingent of collaborators on GILA including Ken Whitebread, Martin Hofmann, Phil DiBona, Steve Wilder from Lockheed Martin and collaborators in multiple universities on the project related to learners and meta reasoning. This work has been partially supported by contract numbers: 55-00680, PO TT0687676, 5710001895-2, 2003*H278000*000, HR0011-05-0019, and F30602-00-1-0579.

REFERENCES

Barber, K., & Martin, C. (1999, May 1). Agent autonomy: Specification, measurement, and ydnamic adjustment. In *Proceedings of the Autonomy Control Software Workshop at Autonomous Agents 1999* (Agents '99), 8-15. Seattle,WA.

CALO (2008). http://www.ai.sri.com/project/CALO

Cowell, A.J., McGuinness, D.L., Varley, C.F., & Thurman, D.A. (2006). Knowledge-worker requirements for next generation query answering and explanation systems. In the *Proceedings of the Workshop on Intelligent User Interfaces for Intelligence Analysis, International Conference on Intelligent User Interfaces* (IUI 2006), Sydney, Australia.

Del Rio, N., & Pinheiro da Silva, P. (2007, June). Identifying and explaining map imperfections through knowledge provenance visualization. *Technical report UTEP-CS-07-43a*, University of Texas at El Paso, El Paso, TX.

Del Rio, N., & Pinheiro da Silva, P. (2007a, November 26-28). Probe-It! Visualization support for provenance. In *Proceedings of the Third International Symposium on Visual Computing (ISVC 2007)*, Lake Tahoe, NV/CA.

Dent, L., Boticario, J., McDermott, J. et al. (1992). A personal learning apprentice. In *Proceedings of the 10 National Conference on Artificial Intelligence*, San Jose, California: AAAI Press, pp. 96-103.

Dzbor, M., Motta, E., & Domingue, J.B. (2004). Opening up magpie via semantic services. In McIlraith et al. (eds), The Semantic Web - ISWC 2004, *Third International Semantic WebConference*. Hiroshima, Japan. *Lecture Notes in Computer Science*, 3298,Springer-Verlag.

Glass, A., McGuinness, D., & Wolverton, M. (2008). Toward establishing trrust in adaptive agents. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI'08)*, Gran Canaria, Spain. Also, KSL Technical Report KSL-07-04.

Guha, R., & McCool, R. (2003). Tap: A Semantic Web platform. *Computer Networks, 42*(5), 557-577.

Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., & Kettula, S. (2005). MuseumFinland - Finnish museums on the Semantic Web. *Journal of Web Semantics, 3*(2), 25.

Huynh, D., Mazzocchi, S., Karger, D. (2005, November 6-10). Piggy bank: Experience the Semantic Web inside your Web browser. In Gil et al. (eds), *The Semantic Web - ISWC 2005, 4th International Next Generation Semantic Web Applications ISWC 2005*. Galway, Ireland. *Lecture Notes in Computer Science*, 3729 Springer-Verlag.

Lashkari, Y., Metral, M., & Maes, P. (1994). Collaborative interface agents. In *Proceedings of the 12 National Conference on Artificial Intelligence*. Seattle, WA: AAAI Press, pp. 444-450.

Lieberman, H., & Kumar, A. (2005, September). Providing expert advice by analogy for on-line help, *IEEE/ACM Conference on Web Intelligence & Intelligent Agent Technology*, Compiègne, France.

Lopez, V., Motta, E., & Uren, V. (2006, June 11-14). PowerAqua: Fishing the Semantic Web. In York Sure and John Domingue (eds.), *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006*, Budva, Montenegro. *Lecture Notes in Computer Science 4011*, Springer, ISBN 3-540-34544-2.

Maes, P. (1994). *Agents that reduce work and information overload communications of the ACM, 37*(7), 31-40.

Maybury, M. (2003). New directions on question and answering, *AAAI Spring Sysmposium,* TR-SS-03-07, Stanford, CA.

McGuinness, D. L. (1996). Explaining reasoning in description logics. Ph.D. Thesis, Rutgers University. Technical Report LCSR-TR-277. Rutgers Department of Computer Science Technical Report Series.

McGuinness, D.L., & Pinheiro da Silva, P. (2004, October). Explaining answers from the Semantic Web: The inference Web approach. *Journal of Web Semantics*, *1*(4), 397-413.

McGuinness, D.L., Ding, L., Glass, G., Chang, C., Zeng, H., & Furtado, V. (2006a) Explanation interfaces for the Semantic Web: Issues and models. Presented in the *3rd International Semantic Web User Interaction Workshop (SWUI'06),* Co-located with the *International Semantic Web Conference*, Athens, Georgia, USA.

McGuinness, D.L., Zeng, H., Pinheiro da Silva, P., Ding, L., Narayanan, D., & Bhaowal. M. (2006b, May 22). Investigations into trust for collaborative information repositories: A Wikipedia case study. *WWW2006 Workshop on the Models of Trust for the Web (MTW'06)*, Edinburgh, Scotland.

McGuinness, D.L., Ding, L., Glass, G., Chang, C., Zeng, H., & Furtado, V. (2006a) Explanation interfaces for the Semantic Web: Issues and models. Presented in the *3rd International Semantic Web User Interaction Workshop (SWUI'06)*, Co-located with the *International Semantic Web Conference*, Athens, Georgia, USA.

McGuinness, D.L., Ding, L., Pinheiro da Silva, P., & Chang, C. (2007). A modular explanation interlingua. In the *Proceedings of the Explanation-aware Computing Workshop (ExaCt-2007)* co-located with the *Association for the Advancement of Artificial Intelligence*, Vancouver, BC.

McGuinness, D., Fox, P., Cinquini, L., West, P., Garcia, J., Benedict, J.L., & Middleton, D. (2007a, July 22-26). The virtual solar-terrestrial observatory: A deployed Semantic Web application case study for scientific research. In *proceedings of the Nineteenth Conference on Innovative Applications of Artificial Intelligence (IAAI-07).* Vancouver, BC, Canada.

Morley, D., & Myers, K. (2004). The SPARK agent framework. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-04),* New York, NY.

Mota, E., & Sabou, M. (2006). *Next generation Semantic Web applications*, ASWC.

Murdock, J.W., McGuinness, D.L., Pinheiro da Silva, P., Welty, C., & Ferrucci, D. (2006, November 5-9). Explaining conclusions from diverse knowledge sources. In the *Proceedings of the Fifth International Semantic Web Conference,* Athens, Ga.

Parsia, B., Sirin, E., & Kalyanpur, A. (2005) Debugging owl ontologies. In the *Proceedings of the World Wide Web Conference*, pp. 633-640.

Plessers, P, & Troyer, O. D. Resolving inconsistencies in evolving ontologies. In the *Proceedings of the European Semantic Web Conference*, pp. 200-214.

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., & Goranov, M. (2003). KIM – A Semantic Annotation Platform. In D. Fensel, K. Sycara, and J. Mylopoulos (eds.), *The Semantic Web - ISWC 2003, Second International Semantic Web Conference. Lecture Notes in Computer Science*, 2870, Springer-Verlag.

Pynadath, D.V., & Tambe, M. (2002). Electric elves: Adjustable autonomy in real-world multiagent environments. In socially intelligent agents – *Creating relationships with computers and robots.* Kluwer Academic Publishers.

Rhodes, B.J., & Starner, T. (1996). Remembrance agent: A continuously automated information retrieval system. *Proceedings, First international Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*. London, UK.

Schraefel, M.C., Shadbolt, N.R., Gibbins, N., Glaser, H., & Harris, S. (2004). CS AKTive space: Representing computer science in the Semantic Web. In *Proceedings of the 13th International World Wide Web Conference*.

Sullivan, D., Grosz, B., & Kraus, S. (2000). Intention reconciliation by collaborative agents. In *Proceedings of the Fourth International Conference on Multi-Agent Systems, IEEE Computer Society Press*, Boston, MA.

Wagner, E., & Lieberman, H. (2003, January). End-user debugging for electronic commerce. *ACM Conference on Intelligent User Interfaces*, Miami Beach.

Weitzner, D.J., Abelson, H., Berners-Lee, T., Hanson, C.P., Hendler, J., Kagal, L., McGuinness, D.L., Sussman, G.J., Krasnow-Waterman, K. (2006). Transparent accountable inferencing for privacy risk management. *Proceedings of AAAI Spring Symposium on The Semantic Web meets eGovernment*. Stanford University, USA: AAAI Press Also available as MIT CSAIL Technical Report-2006-007 and Stanford KSL Technical Report KSL-06-03.

Welty, C., Murdock, J.W., Pinheiro da Silva, P., McGuinness, D.L., Ferrucci, D., & Fikes, R. (2005). Tracking information extraction from intelligence documents. In *Proceedings of the 2005 International Conference on Intelligence Analysis (IA 2005),* McLean, VA, USA.

ADDITIONAL READINGS

Explanation Infrastructure:

Foundational paper: Deborah L. McGuinness and Paulo Pinheiro da Silva. Explaining Answers from the Semantic Web: The Inference Web Approach. *Journal of Web Semantics*. *1*(4). 397-413, October 2004.

Diverse Explanation Presentation Paradigms: Deborah L. McGuinness, Li Ding, Alyssa Glass, Cynthia Chang, Honglei Zeng and Vasco Furtado. Explanation Interfaces for the Semantic Web: Issues and Models. Presented in *the 3rd International Semantic Web User Interaction Workshop(SWUI'06)*, Co-located with the International Semantic Web Conference, Athens, Georgia, USA, November 6, 2006.

Explanation Interlingua:

Newest version: McGuinness, D.L.; Ding, L., Pinheiro da Silva, P., and Chang, C. A Modular Explanation Interlingu . Proceedings of the 2007 Workshop on Explanation-aware Computing (ExaCt-2007), Vancouver, Canada, July 22-23, 2007.

Original version: Paulo Pinheiro da Silva, Deborah L. McGuinness and Richard Fikes. A Proof Markup Language for Semantic Web Services. *Information Systems*. Volume 31, Issues 4-5, June-July 2006, Pages 381-395. Previous version, technical report, Knowledge Systems Laboratory, Stanford University.

Explanation and Trust Requirements Studies:

In Intelligence Settings: Cowell, A.; McGuinness, D.L.; Varley, C.; Thurman, D. Knowledge-Worker Requirements for Next Generation Query Answering and Explanation Systems. In the Proceedings of the Workshop on Intelligent User Interfaces for Intelligence Analysis, International Conference on Intelligent User Interfaces (IUI 2006), Sydney, Australia. 2006.

In Cognitive Assistant Settings: Glass, A.; McGuinness, D.L.; Wolverton, M. Toward Establishing Trust in Adaptive Agents. International Conference on Intelligent User Interfaces (IUI'08), Gran Canaria, Spain, 2008.

Selected Applications

Explaining Task Processing in Learning Settings: McGuinness, D.L.; Glass, A.; Wolverton, M.; Pinheiro da Silva, P. Explaining Task Processing in Cognitive Assistants that Learn. Proceedings of the 20th International FLAIRS Conference (FLAIRS-20), Key West, Florida, May 7-9, 2007.

Explaining Data Mining and Data Usage: Weitzner, D.J.; Abelson, H.; Berners-Lee, T.; Hanson, C.P.; Hendler, J.; Kagal, L.; McGuinness, D.L.; Sussman, G.J.; Waterman, K.K. Transparent Accountable Data Mining: New Strategies for Privacy Protection. Proceedings of AAAI Spring Symposium on The Semantic Web meets eGovernment. AAAI Press, Stanford University, Stanford, CA, USA, 2006.

Explaining Text Analytics: J. William Murdock, Deborah L. McGuinness, Paulo Pinheiro da Silva, Christopher Welty and David Ferrucci. Explaining Conclusions from Diverse Knowledge Sources. The 5th International Semantic Web Conference (ISWC2006), Athens, Georgia, USA, November 5th - 9th, 2006.

Explaining Intelligence Applications: Christopher Welty, J. William Murdock, Paulo Pinheiro da Silva, Deborah L. McGuinness, David Ferrucci, Richard Fikes. Tracking Information Extraction from Intelligence Documents. In Proceedings of the 2005 International Conference on Intelligence Analysis (IA 2005), McLean, VA, USA, 2-6 May, 2005.

Explanation, Trust, and Collaborative Systems:

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. WWW2006 Workshop on the Models of Trust for the Web (MTW'06), Edinburgh, Scotland, May 22, 2006.

Ilya Zaihrayeu, Paulo Pinheiro da Silva and Deborah L. McGuinness. IWTrust: Improving User Trust in Answers from the Web. Proceedings of 3rd International Conference on Trust Management (iTrust2005), Springer, Rocquencourt, France, 2005.

Zeng, H.; Alhossaini, M.; Ding, L.; Fikes, R.; McGuinness, D.L. Computing Trust from Revision History. The 2006 International Conference on Privacy, Security and Trust (PST 2006) Markham, Ontario, Canada October 30 -- November 1, 2006.

Patricia Victor, Chris Cornelis, Martine De Cock, Paulo Pinheiro da Silva. Towards a Provenance-Preserving Trust Model in Agent Networks. Proceeding of the WWW'06 Workshop on Models of Trust for the Web (MTW'06), Edinburgh, Scotland, May 22, 2006.

Patricia Victor, Chris Cornelis, Martine De Cock, Paulo Pinheiro da Silva. Gradual Trust and Distrust in Recommender Systems. Fuzzy Sets and Systems (to appear).

ENDNOTES

[a] The OWL encoding of PML-P is available at: http://iw.stanford.edu/2006/06/pml-provenance.owl

[b] http://www.research.ibm.com/UIMA/