# PML 2: A Modular Explanation Interlingua

**Deborah L. McGuinness[1], Li Ding[1], Paulo Pinheiro da Silva[2], Cynthia Chang[1]**

[1] Knowledge Systems, Artificial Intelligence Laboratory, Stanford University, 345 Serra Mall, Stanford, CA 94305,
{dlm | ding | csc} @ ksl.stanford.edu
[2] Department of Computer Science, University of Texas at El Paso, 500 W. University Ave., El Paso, TX 79968
paulo @ utep.edu

**Abstract**

In the past five years, we have designed and evolved an interlingua for sharing explanations generated by various automated systems such as hybrid web-based question answering systems, text analytics, theorem proving, task processing, web services execution, rule engines, and machine learning components. In this paper, we present our recent major updates including: (i) splitting the interlingua into three modules (i.e. provenance, information manipulation or justifications, and trust) to reduce maintenance and reuse costs and to support various modularity requirements; (ii) providing representation primitives capable of representing four critical types of justifications identified in past work. We also discuss some examples of how this work can be and is being used in a variety of distributed application settings.

## 1. Introduction

Users are increasingly demanding more from question answering systems. They are asking for results, but in addition to results, they are also asking for the corresponding explanations (i.e. information about how the results were generated). This explanatory information is required to help users determine how and when to act on the results. These increased explanatory requirements are especially important when users are obtaining results integrated from distributed and heterogeneous systems. In this paper, we focus on declarative explanation representation. These declarative representations facilitate automated system transparency where users may inspect results along with the sources and processes leading to those results.

Our earlier work focused on explaining results generated by hybrid web-based reasoning systems, such as the question answering systems developed for 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 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, users mainly 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 deductive inference. In this setting, explanations concerning information sources were not critical.

As automated systems become more hybrid and include more diverse components, more information sources are being used and 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 provenance, i.e. metadata about the source of information used such as who authored it, and when it was last updated. Under certain circumstances, such as intelligence settings that motivated the DTO's Novel Intelligence for Massive Data program, provenance concerns often dwarf all others when explanations were required.

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. Users often lack trust knowledge about most portions of huge information spaces such as the Web. Thus, in addition to knowing the authoring sources, it is important to share explicit meta-data about the trustworthiness of information and source such as "I trust Joe's recommendations" or "I trust population data in the CIA World Factbook". In these situations the meta-data may be user authored. In other settings, trust knowledge and trust computation such as link analysis or revision analysis can be represented with the help of ontologies describing operators for trust propagation and aggregation (Zeng, et.al., 2006).

In this paper, we are attempting to address the explanation requirements for a wide range of situations, and we have settings where three different aspects of explanation sometimes dominate to the point that the other aspects are of secondary consideration. Therefore, we have taken on a rationalization and redesign of our original representation Interlingua. Our new modular design can support applications that only desire to capture provenance, and potentially later expand to capturing information about information manipulation steps and trust.

In the rest of this paper, we will introduce our explanation Interlingua and describe the three associated ontologies concerning provenance, information manipulation, and trust. Our work begins with an expansion of the original Proof Markup Language ontology to make it more modular. The result, the subject of this paper, is Proof Markup Language version 2 (PML 2). We will introduce PML 2 using some current examples. We will demonstrate how PML 2 is being used in a few highly integrated systems that leverage many kinds of reasoning, learning, task processing, and text analytics.

## 2. Use Case

To illustrate how PML 2 supports explanation generation, we use a simple scenario where a question is answered by a theorem prover. For example, suppose that John is visiting Stanford and he is aware of an online tour guide agent that has information about local restaurants. The agent is equipped with and an embedded theorem prover to support question answering. In this case the JTP hybrid reasoner (Fikes, et. al., 2003) is used to provide answers. Further, John is aware of a seafood restaurant nearby called Tonys, and since John likes to sample the restaurant specialty and he has some food restrictions, he asks the following question:

```
What type of food is Tonys' specialty?
```

Upon receiving the question, the agent translates it into an internal representation, i.e. a Knowledge Interchange Format (KIF) query (Genesereth and Fikes, 1992):

```
(type TonysSpecialty ?x)
```

Then the agent will run a question-answering process and come up with an answer in KIF.

```
(type TonysSpecialty ShellFish)
```

Given that the answer is encoded in KIF, the on-line agent may invoke a parser that translates the KIF sentence into English.

```
Tony's specialty is Shellfish.
```

John may want to know how this answer was derived and which background knowledge was used to support it. Further, using the KSL Wine Agent[1], John may want to ask a number of other questions such as what type of wines are recommended with Tony's Specialty, what specific wines on the wine list match that recommendation, etc. (Hsu, McGuinness, 2003).

While this is a simple example, the wine agent questions and reasoning steps were designed to be in an accessible "common sense" domain yet they were constructed to be isomorphic to the questions and reasoning steps as observed in a family of description logic-based telecommunications equipment configurators (McGuinness & Wright, 1998) and home theater configurators (McGuinness, et. al., 1995).

In some intelligence settings, e.g., (Cowell, et. al., 2006, Murdock, et. al., 2006), we have users who want to ask questions about what sources were relied on to obtain an answer. In some military settings, e.g., (Myers, et. al., 2007), we have users who want to ask what the system is doing, why it has not completed some processing, and what learned information was leveraged to obtain an answer. In some 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., 2006).

## 3. Representation Components

Our PML explanation ontologies include primitive concepts and relations for representing knowledge provenance. PML 1 (Pinheiro da Silva et al., 2003) provided a single integrated ontology for use in representing information manipulation activities. PML 2 expands and improves upon the work on PML 1 and improves upon it 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 refinement introduces new representational primitives, such as *information*, to enable better reference to identified-things.

PML 2 provides vocabulary for three types of explanation metadata:

- The *provenance ontology* (also known as PML-P) focuses on representational primitives used for describing properties of identified-things such as information, language and sources (including organization, person, agent, services), which are useful for providing lineage.
- The *justification ontology* (also known as PML-J) focuses on representational primitives used for explaining dependencies among identified-things. This includes constructs for representing how conclusions are derived.
- The *trust relation ontology* (also known as PML-T) focuses on representational primitives used for explaining belief and trust assertions.

In what follows, we introduce some important concepts from each module with definitions and examples.

[1] http://www-ksl.stanford.edu/projects/wine/explanation.html

### 3.1. Provenance Ontology

The goal of the provenance ontology (also called PML-P[1]) is to provide a set of extensible representational primitives that may be used to annotate the provenance of information. This includes, for example, representing which sources were used and who encoded the information. The foundational concept in PML-P is *IdentifiedThing*. An instance of IdentifiedThing refers to an entity in the real world, and its properties annotate the entitiy's properties such as name, description, create date-time, authors, and owner.

PML-P includes two key subclasses of *IdentifiedThing* motivated by knowledge provenance representational concerns: *Iinformation* 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 a logical language or a natural language fragment. PML-P users can simply use the value of information's *hasRawString* property if they just want 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 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/docume
nts/tonys_fact.kif</pmlp:hasURL>
   <pmlp:hasLanguage rdf:resource=
"http://inferenceweb.stanford.edu/registry/LG/KIF.owl#KIF" />
</pmlp:Information>
```

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

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, and 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.

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

PML-P provides options for encoding fine-grained references to a span of a text through its *DocumentFragmentByOffset* concept. This is a sub-class of *Source* and *DocumentFragment*. The example below shows how the offset information about *#ST* can be used to support an application that highlights the corresponding span of text in a raw source document (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. The KANI system supported by DTO's Novel Intelligence for Masssive Data program used this feature extensively. More examples can be seen in (Murdock, et. al. 2006, and Welty, et. al., 2005).

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

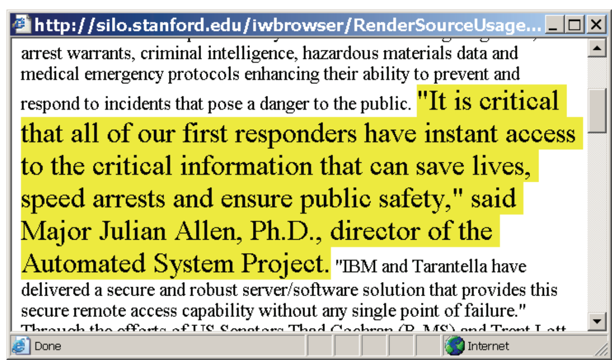


Figure 1: Raw Text fragment with highlighted segment used by text analytics components and represented in PML 2.

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 dependencies between information and sources, i.e. when and how a piece of information was obtained from a source. PML 2 includes a more sophisticated notion of SourceUsage. The encoding below simply shows how PML is used to represent 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.

### 3.2. Justification Ontology

The goal of the justification ontology is to provide the concepts and relations used to encode traces of process (or processes) executions used to derive a conclusion. A justification requires concepts for representing conclusions, one or more sets of conclusion antecedents, and the information manipulation steps used to transform/derive conclusions from sets of antecedents. Note that antecedents may also be conclusions derived from other antecedents). The justification vocabulary has two main concepts:

A *NodeSet* includes structure for representing a conclusion and a set of alternative *InferenceStep*s each of which can provide an alternative justification for a conclusion. The term *NodeSet* is chosen because it captures the notion of a set of nodes (with *InferenceSteps*) 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.

An *InferenceStep* represents a justification for the conclusion of the corresponding NodeSet. The term inference here refers to generalized information manipulation step, so it could be a standard logical step of inference, an information extraction step, simply any computation process step, or an assertion of a fact or assumption. It can also be a very complex process that may not necessarily be able to be described in terms of more atomic processes such a web service or application functionality. Properties of *InferenceStep* 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 of four typical types of justifications for a conclusion:

TYPE I – 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>
```

TYPE II – 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.

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

TYPE IV – regular (antecedent/consequent) justification. The conclusion is derived from a certain list of antecedents and an application of an inference rule. Note of course that PML supports combinations of justification encodings so entire chains of inference rule applications may be encoded. In the example below, assume there are two direct assertions with their unique content.

- #answer31, “*'(subClassOf CRAB SHELLFISH)' has being directly asserted in KSL's Tony's Specialty Ontology as a span of text between byte offset 56 and byte offset 82 as of 10:30 on 2005-10-17*”
- #answer32, “*'(or (not(subClassOf CRAB ?x)) (type TonysSpecialty ?x))' has been directly asserted in Deborah as of 10:30 on 2005-10-17*”,

For example, a theorem prover such as JTP can derive a justification as presented below. It can be read as the sentence “*'(type TonysSpecialty SHELLFISH)' is derived from the application of the General Modus Ponens rule on the two premises #answer31 and #answer32*”.

```
<pmlj:NodeSet rdf:about="#answer3">
   <pmlp:hasConclusion rdf:resource= “#info1” />
   <pmlp:isConsequentOf>
     <pmlp:InferenceStep rdf:about="#step3">
        <pmlp:hasInferenceEngine rdf:resource=
"http://inferenceweb.stanford.edu/registry/IE/JTP.owl#JTP" />
```

```
      <pmlp:hasInferenceRule rdf:resource= "
http://inferenceweb.stanford.edu/registry/DPR/GMP.owl#GMP" />
      <pmlp:hasAntecedentList rdf:resource="#list1"/>
   </pmlp:InferenceStep>
  </pmlp:isConsequentOf>
</pmlj:NodeSet>

<pmlp:AntecedentList rdf:about="#list1 ">
   <ds:first  rdf:resource="#answer31"/>
   <ds:res  rdf:resource="#list2"/>
</pmlp:AntecedentList >
<pmlp:AntecedentList rdf:about="#list2 ">
   <ds:first  rdf:resource="#answer32"/>
</pmlp:AntecedentList >
```

It is notable that antecedents are maintained in an ordered list to (i) ensure consistent presentation of the antecedents during user interaction, and (ii) support some inference engines that use the order of input data.

Type IV justifications can be expanded as an integration of multiple justifications, i.e. one conclusion has more than one justification. The following example is a new justification for #info1 derived by integrating two different justifications for #info1: #step2 justifies #info1 via direct assertion and #step3 justifies #info1 via Generalized Modus Ponens inference:

```
<pmlj:NodeSet rdf:about="#answer4">
  <pmlp:hasConclusion rdf:resource= "#info1" />
  <pmlp:isConsequentOf  rdf:resource="step2"/>
  <pmlp:isConsequentOf  rdf:resource="step3"/>
</pmlj:NodeSet>
```

### 3.3. Trust Relation Ontology

The goal of the trust relation ontology is to provide an extensible set of primitives for use in encoding trust or reputation information associated with information sources. While PML-P and PML-J help users establish belief in information by exposing their knowledge provenance, PML-T complements them by enabling explicit representation and sharing of users' trust assertions (and systems trust calculations) related to other sources including other users.

Currently, PML-T provides basic vocabulary for asserting statements like "agent A believes information B" and "agent C trusts agent D", and its has been used to encode trust information about text fragments in Wikipedia (McGuinness, et. al., 2006). We provided a viewer that could filter Wikipedia content by the trust ratings (as either encoded by users or calculated by our link and revision-history based trust algorithms. We also encoded IWTrust for propagating trust information (Zaihrayeu, 2005). The current trust models are rather simple. More complex trust models (de Cock and Pinheiro da Silva, 2006) can be added to PML-T. PML-T provides a framework for encoding trust relations and does not prescribe a way for representing trust itself. For example, let #fragX be a document fragment from a particular Wikipedia article. We may compute and expose the agent the belief that the Wikipedia community holds in a particular fragment (identified by #fragX). The code below encodes a belief value of .84.

```
<pmlt:FloatBelief rdf:about="#belief1 ">
  <pmlt:hasBelievingAgent rdf:resource= "#wikipedia" />
  <pmlt:hasBelievedInformation rdf:resource= "#info_fragX" />
  <pmlt:hasFloatValue>0.84</pmlt:hasFloatValue >
</pmlt:FloatBelief>

<pmlp:Information rdf:about="#info_doc1">
 <pmlp:hasInfoSourceUsage          rdf:resource="#fragX"          />
</pmlp:Information>
```

Note that the raw value of the trust rating is not necessarily the important content but instead the value in relation to other values may be considered useful. If a consistent method is used for calculating values, then the relative comparisons may be meaningful.

## 4. Discussion

Our work on providing explanation infrastructure originally focused on providing a unified solution to a broad range of explanation needs with one unified representation. The original driving force was the information manipulation explanation. Over the last few years, we have gained requirements increasing the breadth of representational primitives required and also increasing the breadth of the types of users, settings, and question types. We have also found a wide diversity of settings where explanation needs sometimes focus on one aspect of explanation.

A number of explanation efforts have highlighted the need for explanation systems that provide detailed and sometimes extensive support for representing and reporting provenance. Our work explaining hybrid systems that integrate disparate components have exemplified this need more than others. Two major types of components that support this position are text analytics and learning. This is not surprising since both fields rely on algorithms that introduce, increase or propagate uncertainty to explanation conclusions. Additionally, work on integration of scientific information has generated representation and explanation needs for provenance. A nice survey growing issues and concerns related to data provenance for e-Science can be found in (Simmhan, et. al., 2005).

The focus on provenance has led us to include increased expressive capabilities in the primitives for encoding how information has been captured. It also led us to the modularization where the provenance ontology now can stand alone. This provides benefits to users since they now can import (and learn) a smaller ontology, thus reducing the learning curve and overhead. One goal of our work is to minimize the requirements on users for representation and use of the system. Users of PML 2, can now choose to provide simple explanations that focus only on provenance and these explanations and implementations can simply ignore the other portions of the representation and infrastructure.

The requirements for trust relations representation and special presentation mechanisms for trust relation information also evolved with an expanding user and application base. The current trust relation representation has been used in a few relatively simple but large applications. It is a topic for future enhancements increasing first the breadth of representational primitives.

Our work on explaining information manipulation steps may be considered a next logical step to explaining expert systems, along the style of (Scott, et. al, 1984, Swartout, et. al., 1991, Wick and Thomspons, 1992). The difference between this and our original work was that our work needed to be set in distributed web-based environments. The difference between our recent work and the previous work is the additional integration with provenance and trust relation representation and presentation styles. Our work on explaining provenance may also be considered a next logical step in the data lineage work, along the style of (Buneman, 2001). The difference between this database driven work on provenance and our knowledge base driven work on provenance is that our encodings support provenance integrated with theorem proving results and trust encodings. Although many of the representational primitives are similar or identical, the focus on reasoning sometimes requires additional primitives, but more importantly the final explanations need to span provenance, reasoning, and trust considerations.

## 5. Conclusion

This paper described through a number of illustrative examples how a family of three ontologies (PML-P, PML-J and PML-T) is used to encode information about agent's responses. The information about how the agent generated the response, what the response depended on, and associated trust information may be used to generate explanations. These explanations may increase a user's understanding about how responses were generated and thus may facilitate user acceptance of the results. The ontologies are presented as a modular family where users interested in provenance only can use just the provenance ontology, while users interested in justifications and trust may use those ontologies (importing the provenance ontology). The ontologies are being used in a wide range of applications as explanation primitives. Inference Web has been updated to provide a tool suite capable of manipulating, presenting, and validating PML 2.

## References


P. Buneman, S. Khanna, and W. Tan. Why and Where: A Characterization of Data Provenance. In *the Proceedings of the 8th International Conference on Database Theory*, pp. 316-330, January, 2001.

M. De Cock, P. Pinheiro da Silva. A Many Valued Representation and Propagation of Trust and Distrust. In *the Proceedings of International Workshop on Fuzzy Logic and Applications (WILF2005)*, Springer, Crema, 2005.

A. Cowell, D. L. McGuinness, C. Varley, D. Thurman. 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.

D. Ferrucci and A. Lally, Building an example application with the Unstructured Information Management Architecture, *IBM Systems Journal* 43, No. 3, 455-475. 2004.

R. Fikes, J. Jenkins, and G. Frank. JTP: A System Architecture and Component Library for Hybrid Reasoning. In *the Proceedings of the Seventh World Multiconference on Systemics, Cybernetics, and Informatics*. Orlando, Florida, USA. July 27 - 30, 2003.

M. R. Genesereth and R. Fikes, Knowledge interchange format, version 3.0 reference manual. *Technical Report Logic-92-1,* Computer Science Department, Stanford University. 1992.

E. Hsu and D. L. McGuinness,. Wine Agent: Semantic Web Testbed Application. In *the Proceedings of the Workshop on Description Logics.* Rome, Italy, 2003.

D L. McGuinness and P Pinheiro da Silva, 2003. Registry-Based Support for Information Integration. In *the Proceedings of IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, S. Kambhampati and C. Knoblock (Eds.), pages 117-122, Acapulco, Mexico. August, 2003.

D. L. McGuinness and P. Pinheiro da Silva, 2004. Explaining Answers from the Semantic Web: The Inference Web Approach. In *Web Semantics: Science, Services and Agents on the World Wide Web Special issue: International Semantic Web Conference 2003* Volume 1, Issue 4. 2004.

D. L. McGuinness, Lori A. Resnick, and C. Isbell. ``Description Logic in Practice: A CLASSIC: Application." In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, August, 1995.

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

D. L. McGuinness, H. Zeng, P. Pinheiro da Silva, L. Ding, D. Narayanan, and M. Bhaowal. Investigations into Trust for Collaborative Information Repositories: A Wikipedia Case Study. In the *Proceedings of the WWW2006 Workshop on the Models of Trust for the Web (MTW'06)*, Edinburgh, Scotland, May 22, 2006.

J. W. Murdock, D. L. McGuinness, P. Pinheiro da Silva, C. Welty, and D. Ferrucci, Explaining Conclusions from Diverse Knowledge Sources. In *the Proceedings of the 5th International Semantic Web Conference (ISWC2006)* Athens, Georgia, USA, November 5th - 9th, 2006.

K. Myers, P. Berry, J. Blythe, K. Conley, M. Gervasio, D. McGuinness, D. Morley, A. Pfeffer, M. Pollack, and M. Tambe, An Intelligent Personal Assistant for Task and Time Management, to appear in *AI Magazine*, 2007.

P. Pinheiro da Silva, D. L. McGuinness and R. Fikes, 2006. A Proof Markup Language for Semantic Web Services. *Information Systems*, 31(4-5), June-July, pp 381-395.

P. Pinheiro da Silva, D. L. McGuinness and R. McCool. Knowledge Provenance Infrastructure. *IEEE Data Engineering Bulletin*. 26(4), pp. 26-32, December 2003.

A. Scott, W. Clancey., R. Davis, R., and E. Shortliffe.. Methods for Generating Explanations. In Buchanan and Shortliffe, (eds.), Rule-Based Expert Systems, Addison-Wesley, 1984.

Y. Simmhan, B. Plale, and D. Gannon. A Survey of Data Provenance in e-Science. In the SIGMOD Record, Vol. 34, No. 3, Sept, 2005.

W. Swartout, C. Paris, and J. Moore. Explanations in Knowledge Systems: Design for Explainable Expert Systems. IEEE Expert: Intelligent Systems and Their Applications, vol. 6, No. 3, pp 58-64, 1991.

C. Welty, J. W. Murdock, P. Pinheiro da Silva, D. L. McGuinness, D. Ferrucci, and R. Fikes. Tracking Information Extraction from Intelligence Documents. In *the Proceedings of the 2005 International Conference on Intelligence Analysis* (IA 2005), McLean, VA, USA, 2-6 May, 2005.

M. Wick, W. Thompson. Reconstructive Expert System Explanation. Artificial Intelligence 54(1-2): 33-70, 1992.

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

H. Zeng, M. Alhossaini, L. Ding, R. Fikes, D. L. McGuinness. Computing Trust from Revision History. In *the Proceedings of the 2006 International Conference on Privacy, Security and Trust (PST 2006)*, Markham, Ontario, Canada.