Explanation in the Semantic Web: a survey of the state of the art

Rakebul Hasan, Fabien Gandon

▶ To cite this version:

Rakebul Hasan, Fabien Gandon. Explanation in the Semantic Web: a survey of the state of the art. [Research Report] RR-7974, Inria. 2012. hal-00702277

HAL Id: hal-00702277

https://hal.inria.fr/hal-00702277

Submitted on 29 May 2012

Explanation in the Semantic Web: a survey of the state of the art

Rakebul Hasan, Fabien Gandon

Explanation in the Semantic Web: a survey of the state of the art

Rakebul Hasan*, Fabien Gandon*

Project-Team Wimmics

Abstract: Semantic Web applications use interconnected distributed data and inferential capabilities to compute their results. The users of Semantic Web applications might find it difficult to understand how a result is produced or how a new piece of information is derived in the process. Explanation enables users to understand the process of obtaining results. Explanation adds transparency to the process of obtaining results and enables user trust on the process. The concept of providing explanation has been first introduced in expert systems and later studied in different application areas. This paper provides a brief review of existing research on explanation in the Semantic Web.

Key-words: Explanation, justification, Semantic Web, trust

* INRIA Sophia Antipolis Méditerranée

L'explication dans le Web sémantique: une enquête sur l'état de l'art

Résumé : Ce document présente un panorama de l'état de l'art des explications dans le Web sémantique

Mots-clés : Explication, justification, Web sémantique, confiance

Contents

1   Introduction

Semantic Web applications use interconnected distributed data and inferential capabilities to compute their results. A user might not be able to understand a Semantic Web application has solved a given query deriving new information and integrating information from across the Web, and therefore the user might not trust the result of that query. Semantic Web applications should provide explanations about how they obtain results in order to ensure their effectiveness and increase their user acceptance [20]. Semantic Web applications should not only provide explanations about how the answers were obtained, they should also explain and allow users to follow the flows of information between them [19].

Expert systems were among the first software systems to include explanation facilities. Expert systems were intended to partially incorporate and augment human expertise in a well-defined, protocol oriented domain. The need for understanding why a system has failed to meet certain requirement has given rise to the need for explanation facilities. Expert systems were required not only to solve problems within a well-defined domain but also to impart an understanding of a given field by providing explanation of a given concept or entity. Expert systems without the explanation facilities became subject to credibility problems, especially in the safety critical domain and in the systems with rich intellectual contents. Explanation facilities were introduced in the expert systems with the intention of providing an understanding of why and how a particular conclusion has been reached. Overviews of explanation in expert systems can be found in [12, 21, 27]. Explanation facilities in expert systems have evolved from reasoning trace oriented explanations, primarily useful for developers and knowledge engineers, to more user oriented interactive explanations justifying why a system behavior is correct, to casual explanations generated in a decoupled way from the line of reasoning. The realization of the explanation facilities in expert systems were motivated by enabling transparency in problem solving, imparting an understanding of why and how a conclusion has been reached, and hence enabling trust on the reasoning capabilities of the expert systems. These developments motivated adaptation and development of explanation facilities in other fields such as machine learning [10, 26], case-based reasoning [8, 23], recommender systems [28], and Semantic Web.

In the this paper, we provide a review of the existing approaches to provide explanation in the Semantic Web. In section 3, we provide an overview of different approaches to represent the metadata which enable support for reasoning and provenance information. In section 4, we provide an overview of different approach for generation and presentation of explanation. In section 5, we focus our discussion on the important aspects of explanation approaches in the Semantic Web. In section 6, we conclude the paper.

2   Designing Explanation-Aware Semantic Web Applications

McGuinness *et al.* [19] investigate the Semantic Web application paradigms from an explanation perspective. Question answering on the Semantic Web, which is an interactive process involving human users and software agents, requires more processing steps than simple database retrieval. The paradigm shift introduced by the Semantic Web applications, from answering queries by retrieving explicitly stored information to using inferential capabilities, generates new requirements to ensure their effective use: *"applications must provide explanation capabilities showing how results were obtained"*. Explanations improve users' understating of the process of obtaining results and add transparency to the process. Explanations make the results and process of obtaining results more credible. McGuinness *et al.* analyse the Semantic Web application paradigms in the context of explanations with the aim of identifying explanation requirements.

2.1 Characterization of Semantic Web Applications

McGuinness *et al.* characterize the important features of the Semantic Web applications from an explanation perspective:

Collaboration involves interaction and sharing of knowledge between agents that are dedicated to solve a particular problem. Semantic Wikis and applications developed via integration of Semantic Web services and multi-agent systems are examples of collaborative Semantic Web applications. Managing provenance, trust and reputation are the important issues form a collaboration perspective. Semantic Wikis, for example, enable storing project history and utilizing tools to perform intelligent queries on this history, and subsequently enable more transparent content management. Furthermore, in the reactive multi-agent systems, individual agents are often not autonomous and therefore the "intelligence" is a collective one which comes from interactions between agents. These heterogeneous and loosely-coupled agents or systems component must be able to discover new services by accessing service descriptions.

Autonomy of an individual agent can be seen as the ability to act independently. The degree of autonomy in traditional Web-based applications, such as search engines, is very little as they take input from users to achieve their goals. Semantic Web applications on the other hand have more autonomy. For example, a shopping agent can autonomously decide which services to call and compose, which content sources to use, how to enhance a query with background knowledge, to provide more useful and efficient answers to user's questions.

Use of ontologies provide support for heterogeneous and distributed data integration and allow dealing with inconsistencies of data from multiple sources in the Semantic Web applications. Other possible uses of ontologies are in content search and context search. In content search, search engines can go beyond statistical methods by using background knowledge bases to enhance search queries. In context search, search engines can exploit context information, such as location and preferences encoded in background ontologies, to improve search results. Ontologies also play important role in describing domain knowledge, problem areas, and user preferences in the context of agents with reasoning capabilities. Ontologies are often used as common vocabularies among multiple agents allowing describing agent communication protocols.

2.2 Explanation Criteria

Given these features of Semantic Web applications, McGuinness *et al.* identify criteria such as different types of explanations and consumption of explanation by humans or machine agents to analyze explanation requirements. The transitions logs of the manipulation steps which have been performed to derive results are known as justifications. These justifications enable provision of detailed explanation. Human understandable explanation generated from abstraction of these justifications is an important type of explanation providing details about what has been done to produce a conclusion. Provenance metadata enable providing another kind of explanation with details of information sources. Trust is another important subject in the context of explanation in the Semantic Web, especially, in distributed settings such as large online social networks. Representation, computation, combination, and presentation of trust present challenging research questions in this context. For machine consumption, explanations should be represented using standards in order to enable interoperability. Representation of justifications and provenance metadata in an interoperable way enables external software agents to make sense of explanation metadata. Human computer interface (HCI) issues such as the level of user expertise and the

context of problem should be considered in the explanations that are aimed for human consumption. Visual explanations should be provided in a manageable way. Explanations should be presented with options for different degree of details such as summarized view and focused view allowing navigation between related information.

2.3   Explanation Requirements

Following the discussion of these explanation criteria, the authors discuss how these criteria relate to the Semantic Web application features discussed previously. From explanation and collaboration perspective, trust and reputation are important issues. If answers generated by Semantic Web applications are to be believed by users, the Semantic Web applications and agents will need to provide explanation of how the answers were obtained, especially, in cases where these Semantic Web agents and applications collaborate to generate complex results. In a distributed setting, users should be provided with explanation of the flow of information between the involved agents in the reasoning process. In addition, it is also important to provide explanation based on provenance metadata in this context. Explanation based on provenance will add transparency in the problem solving process by providing details about information sources used to obtain the answers. Explanation becomes even more important in applications with higher degree of autonomy. Autonomous agents should provide explanation on their complex process of obtaining results which might include complex logical inferences or statistical methods. Explanation plays an important role in applications with lower degree of autonomy as well. For example, in search engines, explanation facilitates improved query refinement by enabling users to better understand the process of obtaining search results. Ontologies play key role in the context of supporting and providing explanation. For example, ontologies can be effectively used to develop an interlingua to enable an interoperable explanation.

In [20], McGuinness *et al.* present the following requirements for an infrastructure to enable applications to generate distributed and portable justifications, and subsequently presenting the justifications as user-friendly explanations for any produced answer:

Support for knowledge provenance enables to establish user trust on background reasoners in question answering settings by enabling users to understand the source information of information used in the reasoning process. Knowledge provenance meta information may include source name, date and authors, authoritativeness of the source, degree of confidence, etc.

Support for reasoning information enables recording traces of information manipulation steps performed by the reasoners to obtain results. These reasoning traces may include meta-information about reasoners and actions performed by reasoners to derive answers, e.g. detailed trace of inference rules applied to obtain a conclusion.

Support for explanation generation enables generation of human understandable explanations from reasoning traces and knowledge provenance. These knowledge provenance and reasoning traces are commonly known as justifications. Users may require explanation of *what* rules have been applied or *how* manipulation steps were performed to obtain a result. These user requirements about what kinds of explanation are required should be taken into account while recording justification meta-information.

Support for proof presentation enables presenting explanations in different forms, e.g. graph-based or textual presentation of the reasoning steps. Explanations should be presented with different degree of details taking into account the users' expertise and the context of problems.

We discuss these requirements in more details in 3 and 4.

2.4   Explanation-Aware Software Design (EASD) Principles

Forcher *et al.* present the explanation-aware system design (EASD) principles in [9]. The EASD principles concern two key aspects, namely the development and the runtime of a system. The integration of explanation capabilities in a system during its development should not be too complicated and should not effect system performance and efficiency. The provided explanations should be understandable and sufficient for the users. Designing understandable explanation includes considering goals, preferences, knowledge of the system, and knowledge of the domain of users. Users should be provided with feedback mechanism to provide feedback about explanations to enable a dialog for a better understanding of explanations. A better understanding of explanation enables users to understand how and why a system reached a conclusion. This understanding enables transparency to the reasoning process of a system. An explanation of a reached conclusion should be a simplification of the actual process that the system went through to reach the conclusion. Explanations should have different kind of presentation such as natural language explanation, graphical explanation, or interactive explanation. In addition, there should be different kind of explanation such as *concept explanation* explaining a certain concept, and *action explanation* explaining the cause of a fact, action, or situation. The authors highlight that EASD does not intend to add explanation capability in the already developed systems. Instead, EASD provides guidelines for the developers to integrate explanation capabilities during the development process. Concerning the runtime aspect, the system must be aware of the explanation scenarios and must be able provide explanation accordingly during its runtime. The authors consider three participants in any explanation scenario:

- *Originator*: the system or the agent that solves the problem.

- *User*: the addressee of the explanation.

- *Explainer*: the responsible for presentation, computation, and communication of explanation.

Different aspects of an explanation scenario such as the reactions of users upon receiving explanation, style of the explanation, and different types of user interaction play important role in different type of applications. These guidelines form the basis of EASD principle.

Forcher *et al.* complement the EASD principles with an abstract architecture of a multi-layered explanation model, shown in Figure 1. The *tracing process* constructs the *trace model* comprising of the information regarding the behavior of the originator. What goes in the trace model is predefined with respect to the explanation requirements of users. However, the process of formulating the information need results in extra construction overhead which should be kept to a minimum level. The *selection process* filters the required information and produces the *selection model* for providing explanation to users in a certain situation. The *construction process* produces the *construction model* by adding supporting information to the selection model or removing unrelated information from the selection model. The construction model contains the domain and context dependent information. The *externalisation process* transforms the construction model to the *externalisation model* containing a formal description for communicating explanations in various communication forms such as texts, charts, tables, etc. Finally, the *presentation process* applies the suitable layouts and styles to the externalisation model and transforms the externalisation model to the *presentation model* which is the final presentable explanation to end users. The *mediation model* contains the information about explanations such as why an

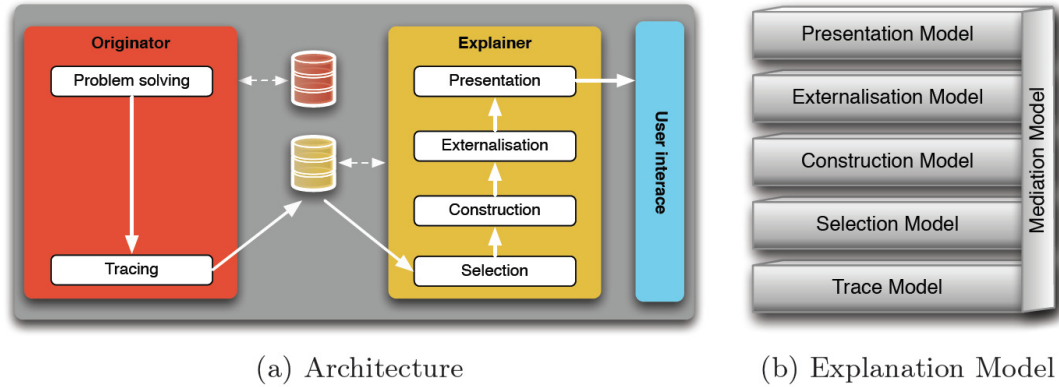(a) Architecture   (b) Explanation Model

Figure 1: Explanation-Aware Software Design.

explanation is provided in an explanation scenario, how the presentation and the content of the explanation correspond to each other.

3 Metadata for Explanation

McGuinness *et al.* [18] describe explanation in the context of Semantic Web as "Semantic Web metadata about how results were obtained". In the following subsections, we present the existing approaches to represent explanation metadata.

3.1 Proof Markup Language (PML)

According to McGuinness *et al.* [18], provenance information are important for a large number of users to believe the answers. Moreover, often provenance information (e.g. source of a particular piece of information) are the only information demanded by users as explanation. Therefore, provenance metadata concerning information sources such as how, when, and from whom any given piece of data is obtained are important aspects of explanation metadata. Another important aspect of explanation metadata is the information manipulation traces. In the Semantic Web applications, producing a result involves several information manipulation steps which can derive conclusions. The transaction logs of these information manipulation steps, namely the "traces", allow explaining how a result has been derived. The metadata representing the manipulation steps including conclusions, the information manipulation operations, and its antecedents are commonly known as justifications. These justifications facilitate rich explanation of how a conclusion has been drawn. Representing trust related information is another is another aspect of explanation metadata.

Proof Markup Language (PML)[1][25] is an interlingua to represent explanation metadata. PML consists of three OWL ontologies for representing provenance information, justification information, and trust information. Figure 2 shows the concepts of PML.

The PML provenance ontology (PML-P) provides primitives for annotating real world things. An instance of *IdentifiedThings* represents a real world thing and its properties allow annotating

---

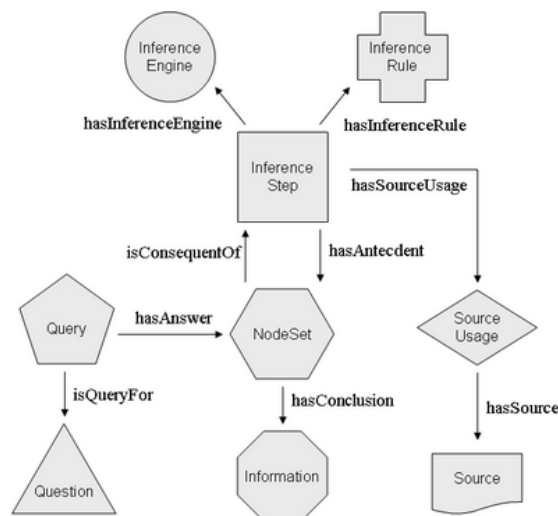[1]http://tw.rpi.edu/portal/Proof_Markup_Language

Figure 2: PML concepts.

metadata such as name, creation date-time, description, owners and authors. Two key provenance related subclasses of *IdentifiedThings* are: *Information* and *Source*. The concept *Information* represents information such as a text fragment, or a logical formula. The concept *Source* represents information hosts and containers such as a web page, a document, and a person. PML-P provides a simple and extensible taxonomy of *Source*. The *SourceUsage* concept allows capturing metadata about accessing a source at a certain time. PML-P provides some auxiliary concepts such as *Format*, *InferenceRule*, and *Language* to enable annotations for information formats, inference rules, and languages.

The PML justification ontology (PML-J) provides primitives for encoding justifications about derivation of a conclusion. A justification in PML-J is not limited to a logical reasoning step. A justification can be a step of a computation process, e.g. an information extraction step or a web service execution. A justification can also be a factual assertion or an assumption. An instance of *NodeSet* hosts a set of alternative justifications for a conclusion. The *InferenceStep* concept allows encoding additional justification related details such as a set of antecedent *NodeSet* instances, inference rules using the *InferenceRule* concept, and inference engines using the InferenceEngine concept.

The PML trust ontology (PML-T) provides primitives for representing trust assertions concerning sources and belief assertions concerning information. PML-T complements the established users' understanding by allowing explicit representation and sharing of trust related assertions.

3.1.1 Lightweight use of PML

The authors in [24] introduce a restricted subset of PML constructs and tools for encoding very basic justifications and performing tasks such as retrieval and browsing of provenance information. The authors present strategies to simplify encoding in PML. The authors present three *simplification assumptions. Simplification Assumption 1* is about not using alternative justifications so that each PML NodeSet has a single InferenceStep. The *Simplification Assumption 2* is about not encoding any knowledge about the inference mechanism itself i.e. the the rules

used to transform information in each steps of a given information manipulation process. The *Simplification Assumption 3* is about not encoding any knowledge about how information have been asserted form a given source i.e. not encoding the processes and services used behind. The authors provide an evaluation presenting a user study to verify whether provenance plays a role for scientists to correctly identify and explain quality of maps. Although the evaluation is not exactly about the lightweight encoding of PML, the authors claim that the results of the evaluation are significant for lightweight encoding of PML as the provenance information used in the study was encoded considering the recommendations for lightweight use of PML.

3.1.2   PML-Lite

PML-Lite[2] is an ongoing work with the aim of constructing a simple subset of the three modules of PML. PML-Lite takes an event based modeling approach. PML-Lite provides primitives to represent provenance of data flows and data manipulations. The *Event* class represents information manipulation and flow steps. The *Event* class is an equivalent class of the *NodeSet* class of PML-J. The *Operation* class represents a performed operation and it is an equivalent class of the *InferenceRule* class in PML-P. The input and output data are represented by the *Data* class. The *Information* class and the *Document* class of PML-P are defined as subclasses of *Data*. The *Agent* class of PML-Lite is directly defined as a subclass of *Thing* unlike PML-P where it is defined as a subclass of *Source*. The *Agent* class of PML-Lite represents the actor performing an operation. Finally, PML-Lite allows representing the place of a performed operation through its *Location* class

3.2   AIR Justification Ontology (AIRJ)

AIR (**A**ccountability **I**n **R**DF) [15] is a Semantic Web-based rule language focusing on generating and tracking explanation for inferences and actions. AIR provides features such as coping with logical inconsistencies by allowing isolation of reasoning results which can cause inconsistencies in global state; scoped contextualized reasoning; and capturing and tracking provenance information such as deduction traces, or justifications. AIR provides two independent ontologies. One ontology allows the specification of AIR rules and the other one allows describing justifications. Figure 3 shows the concepts of the AIR justification ontology. The prefixes pmll, pmlp, pmlj, air in Figure 3 are namesapce prefixes for PML-Lite, PML-P, PML-J, AIR rule ontology respectively.

The reasoning steps of the AIR reasoner are considered as events and modeled as subclasses of *pmll:Event*. These different events are represented as *BuiltinAssertion*, *BuiltinExtraction*, *ClosingTheWorld*, *ClosureComputation*, *Dereference*, *Extraction*, and *RuleApplication*. Rules are considered as operations. *air:Rule* represents rules and it is defined as a subclass of *pmll:Operation*. The ontology also provides properties to enable representing variable mappings in the performed operations.

4   Generation and Presentation of Explanation

As discussed in explanation requirements, what types of explanations are generated and how they are presented to users are important criteria for success of explanation-aware systems. In this section, we discuss how the existing the Semantic Web systems generate explanations, what they explain, and how these explanations are presented.
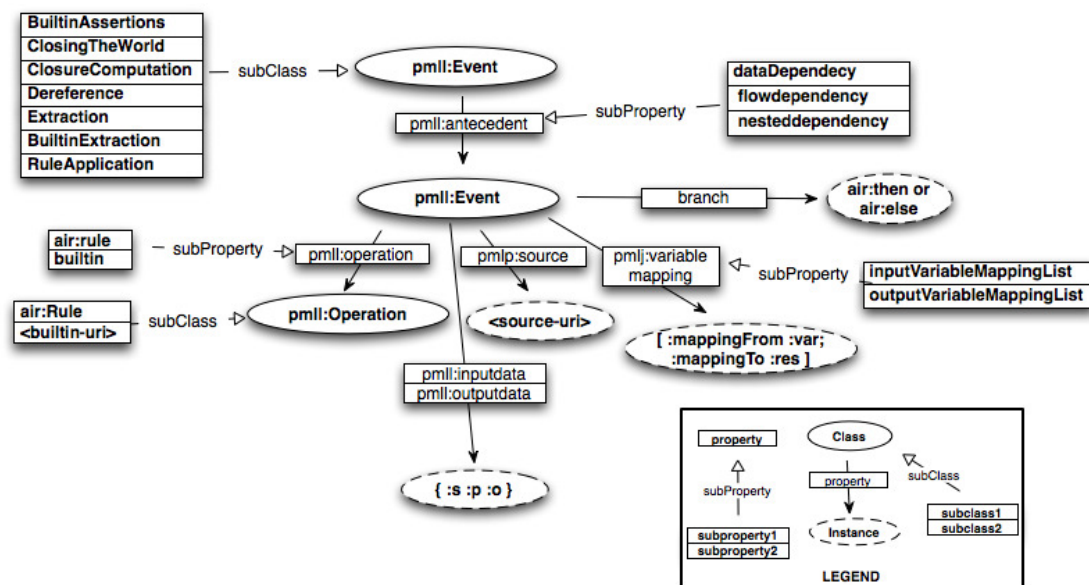
---

[2]http://tw.rpi.edu/web/project/TAMI/PML-Lite

Figure 3: AIR Justification Ontology: the concepts without any prefix are in the airj namespace.

4.1 Inference Web

Inference Web [18, 19, 20] is an explanation infrastructure which addresses explanation requirements of web services discovery, policy engines, first order logic theorem provers, task execution, and text analytics. Information manipulation traces of these various kinds of systems are encoded using the Proof Markup Language (PML). These metadata in PML are also known as PML proofs. Inference Web provides a set of software tools and services for building, presenting, maintaining, and manipulating PML proofs.

Figure 4 shows the architecture of inference web infrastructure. IWBase provides an interconnected network of distribute repositories of explanation related meta information. IWBase provides a registry-based solution for publishing and accessing information. Content publishers can register metadata and other supporting information such as inference rules, and inference engines. IWBase provides services to populate PML proofs. IWBase exposes the populated metadata as PML documents and provides browsing interfaces to access them. These PML documents can be also accessed by resolving their URI references. Figure 5 shows the browsing interface of IWBase.

IWSearch searches for PML documents on the Web and maintains an inventory of these documents. Users can then search for PML documents using different search interfaces offered by IWSearch. Inference Web provides a browser called IWBrowser which can display PML proofs and explanations in number of different formats and styles. These rich presentations include a directed acyclic graph (DAG) view known as global view, a focused view enabling step-by-step navigation between related explanations, a filtered view to show selected parts of an explanation, an abstraction view which shows abstract views of explanations with different degrees of details, and finally a discourse view which allows follow-up questions. Figure 6 shows an example presentation by IWBrowser. It shows the abstraction view for a proof. The IWAbstractor component allows users to write abstraction patterns for PML proofs and matches these patters against
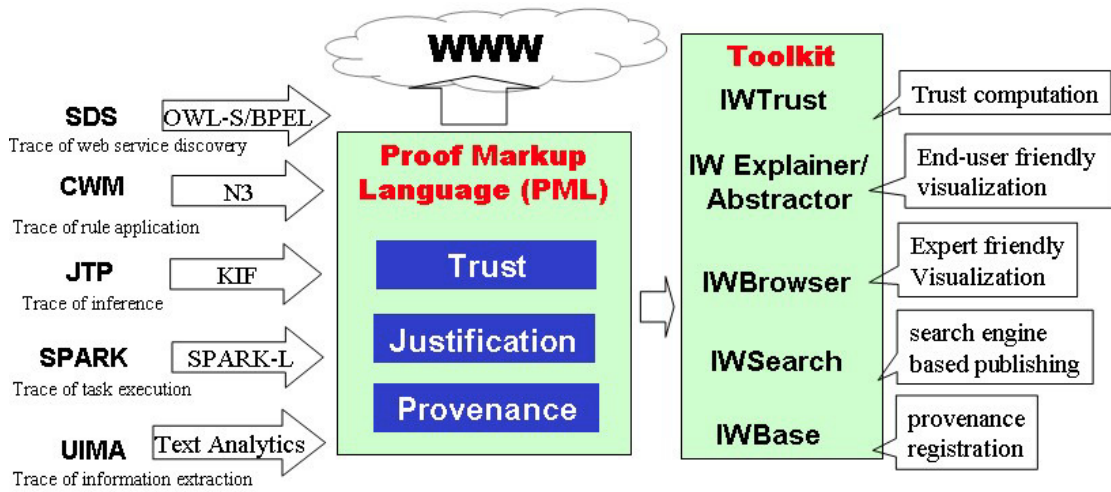
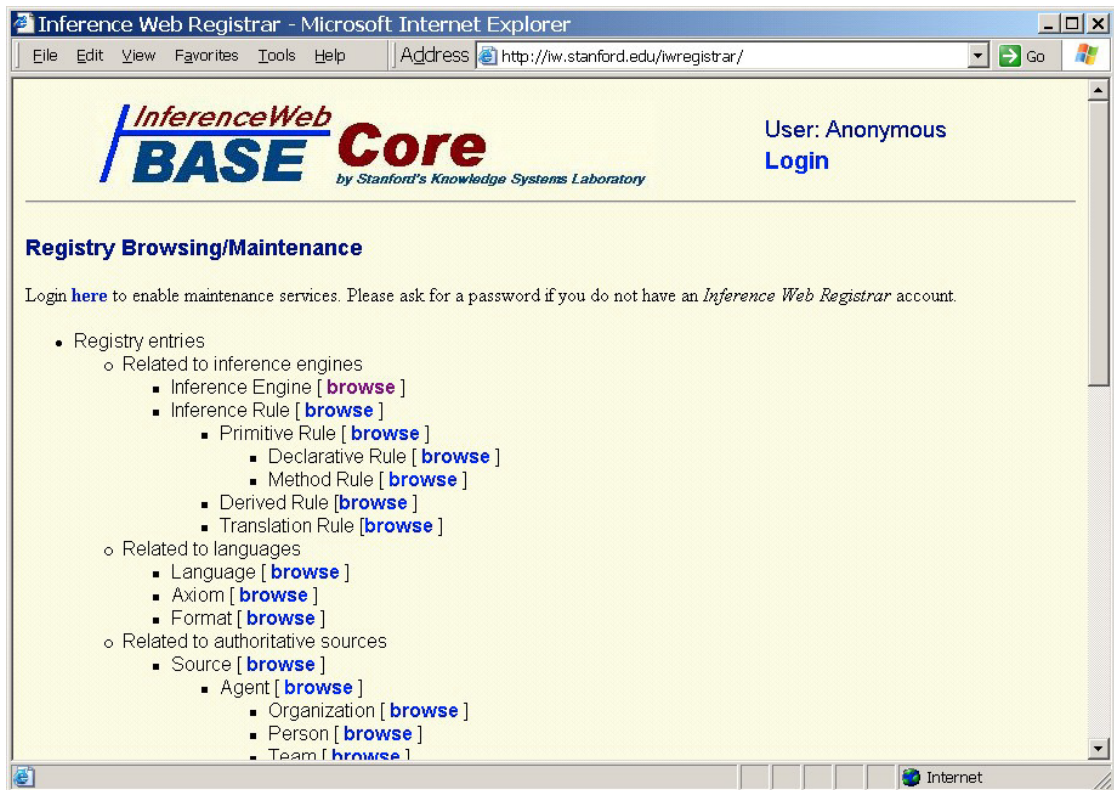Figure 4: Inference Web Explanation Infrastructure.

Figure 5: The browsing interface of IWBase.

proofs to provide an abstract view. In figure 6 for instance, an abstract view with many fewer steps is shown in the left side than the original proof shown in the middle with blue outline.
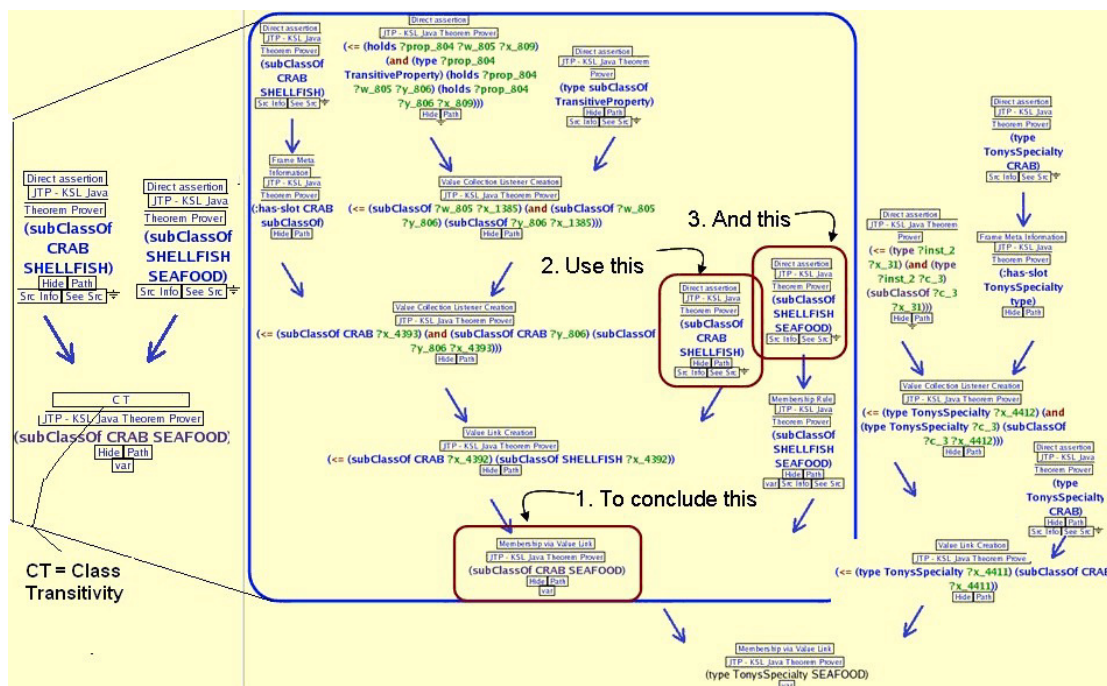

Figure 6: An example of abstraction view.

Inference Web provides a general trust infrastructure called IWTrust which provide explanation with trust related information and allows filtering out unreliable information. IWTrust includes representation of trust aspects and trust computation services. In addition, it provides a browser with trust view to render the trust annotation in an article. Figure 7 shows an example of trust view. Different fragments of an article are rendered in different colors depending on the trustworthiness of these fragments when a user clicks the trust tab in the browser. This allows users to have an understanding of trustworthiness of rendered information just by looking at an article.

4.2 OntoNova Question Answering

OntoNova [2] is an ontology-based question answering system in chemistry domain. OntoNova provides explanation in natural language along with its answers. OntoNova generates answer justifications with meta-inferencing. OntoNova inference engine produce log files representing the proof tree for a given answer. Such a file is given as an input to a second meta-inference step. This second meta-inference step explains the proof tree in natural language describing how the answer was derived. OntoNova allows specifying meta-inference rules for the original rules for question answering. The two step method for providing explanation has advantage such as: (i) provision of additional information with explanations when proof trees do not contain enough information, (ii) filter explanation paths in case of redundancies for same results, (iii) provision of explanation with different degree of details, (iv) provision of personalized explanation for
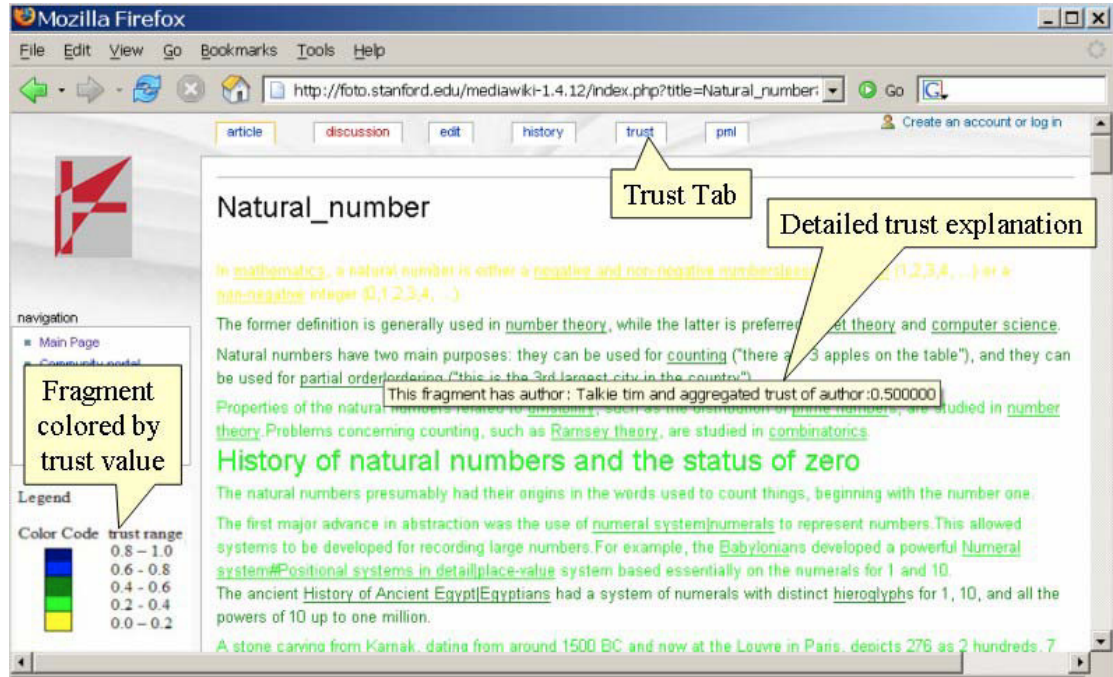
Figure 7: An example of trust view.

different contexts.

4.3 WIQA Framework

The *WIQA - Web Information Quality Assessment Framework* [6] provides functionalities for quality-based information filtering. The WIQA framework allows to employ different policies for information filtering. These policies combine content-based, context-based, and rating-based quality assessment metrics. An information consumer's understanding of the employed quality assessment metrics is a major factor that influence whether the information consumer trust or distrust any quality assessment result. The WIQA framework provides detailed explanation of information filtering process with the aim of supporting information consumers in their trust decisions.

The WIQA framework is able to provide explanation of why information satisfies a given WIQA-PL policy. The WIQA framework provides explanations in natural language for human consumption and explanation in RDF for further processing by software applications. The explanation generation process contains two steps. First, WIQA generates the parts of explanations of why constrains expressed as graph patters are satisfied. These different parts of explanations are generated using a template mechanism. In the second step, these explanation parts are supplemented with additional explanations of why constrains expressed as extension functions are satisfied.

The structure and the content of explanations are defined by using a template mechanism. Figure 8 shows the policy "Only accept information from information providers who have received more positive than negative ratings." The graph pattern in lines 7-10 describes the constrains that

```
1. NAME "More positive ratings"
2. DESCRIPTION "Only accept information from information providers who
3.              have received more positive than negative ratings."
4. PATTERN
5.      {  EXPL "The information was asserted by "
6.              ?authority " and " wiqa:morePositiveRatings .
7.         GRAPH fd:GraphFromAggregator
8.              { ?GRAPH swp:assertedBy ?warrant .
9.                ?warrant swp:authority ?authority .
10.               FILTER wiqa:morePositiveRatings(?authority) . }
11.     }
12.
```

Figure 8: An example of WIQA-PL policy.

must be satisfied for this policy. The policy uses an extension function *wiqa:MorePositiveRatings*. The policy contains an explanation template in lines 5-6. The explanation template has a reference to the *wiqa:MorePositiveRatings* extension function for generating explanations related to this extension function. Figure 9 shows the explanation of why a triple matches the policy shown in figure 8. Line 13 of the explanation is generated by the explanation template and explains why the constrain expressed as graph pattern is satisfies. Lines 14-25 are generated by the extension function *wiqa:MorePositiveRatings* and explain its reasoning process.

In addition to the natural language-based explanations, WIQA provides RDF-based explanations. WIQA describes the explanation trees (parts and subparts of an explanation) using the Explanation (EXPL) Vocabulary[3]. WIQA provides the feature of construct template to enable describing the contents of RDF explanations. A construct template is defined by CONSTRUCT EXPLANATION keywords and a set of triple patterns. The contents of an RDF explanation are generated by taking the matched triples of the set of triple patterns defined in the construct template for this RDF explanation. The RDF-based explanations can be used by other software applications to do further processing in their application-specific ways.

4.4 Proof Explanation Using Defeasible Logic

The authors in [3, 4] present a nonmonotonic rule system based on defeasible logic which is able to answer queries along with proof explanations. Defeasible logic enables reasoning with incomplete and inconsistent information. The traces of the underlying logic engine are transformed to defeasible logic proofs. The authors introduce an extension to the RuleML[4], a unifying family of Web rule languages, to enable formal representation of explanations using defeasible logic. The proofs generated from the traces are represented using this RuleML extension. Software agents can consume and verify these proofs. In addition, the authors present graphical user interfaces to visualize the proofs and interact with them. Finally, the authors present an agent interface to enable multi-agent systems to interact with their system.

4.5 Explanation of entailments in OWL ontologies

Horridge *et al.* present two fine-grained subclasses of justifications called laconic justifications and precise justifications [14]. Laconic justifications are the justifications whose axioms do not

---

[3]http://www4.wiwiss.fu-berlin.de/bizer/triqlp/
[4]http://ruleml.org

```
1.  The triple:
2.
3.   Siemens AG has positive analyst report: "As Siemens agrees
4.   partnership with Novell unit SUSE ..."
5.
6.  fulfills the policy:
7.
8.   Only accept information from information providers who
9.   have received more positive than negative ratings.
10.
11. because:
12.
13.  The information was asserted by Peter Smith and
14.  Peter Smith received the following numbers of ratings:
15.        - 3 positive ratings (see detail 1)
16.        - 2 negative ratings (see detail 2)
17.
18.  Detail 1: Peter Smith received positive ratings from:
19.        - John Reynolds
20.        - Mary O'Conner
21.        - Elisa Armstoen
22.
23.  Detail 2: Peter Smith received negative ratings from:
24.        - Dave Berser
25.        - Colin Marwick
26.
```

Figure 9: Explanation of why a triple matches a policy.

contain any superfluous parts. Precise justifications are derived from laconic justifications and each of whose axioms represents a minimal part of the justification. The authors also present an optimized algorithm to compute laconic justifications showing the feasibility of computing laconic justifications and precise justifications in practice. The authors provide a Protégé ontology editor[5] plugin as a tool to compute these types of justifications[6]. This tool shows justification-based explanations of entailments. Figure 10 shows examples of explanations. A user can select an entailment from the list of entailments shown in the left handside. The right handside shows the explanation for the selected entailment.
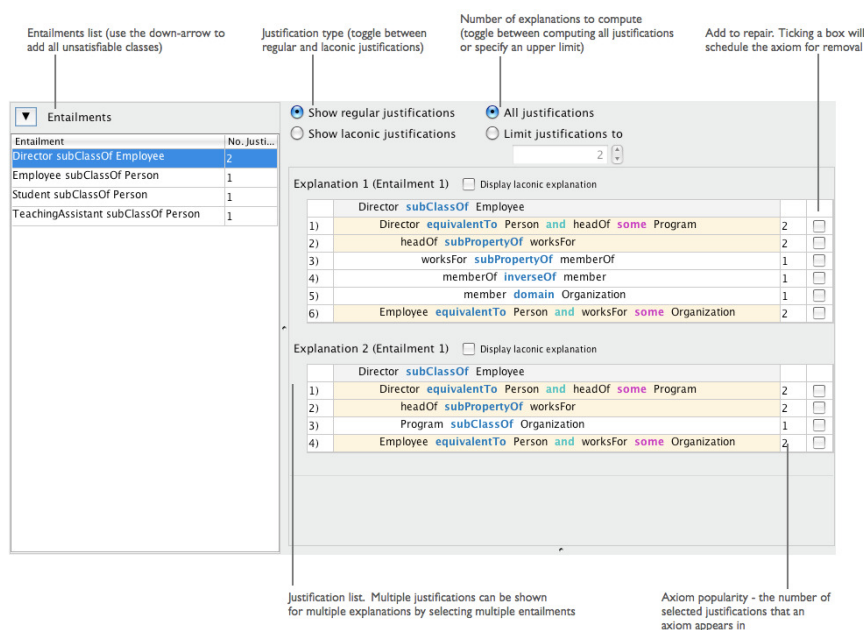

Figure 10: An explanation based on laconic and precise justifications

4.6   Knowledge in a Wiki (KiWi)

Kotowski and Bry [17] argue that explanation complements the incremental development of knowledge bases in the frequently changing wiki environments. The authors present a semantic wiki called KiWi[7] which takes a rule-based inconsistency tolerant reasoning approach that has the capability of explaining how a given piece of information was derived. The reasoning approach also allows knowledge base updates in an efficient way by using reason maintenance. The authors argue that providing explanation is important for supporting users' trust and facilitates determining main causes of inconsistencies. KiWi stores the justifications of all derivations and use them for explanation and reason maintenance. KiWi presents explanations as natural language explanations and as tree-based explanations. Figure 11 shows the textual explanation for a triple in KiWi. Figure 12 shows a part of interactive explanation tree. The number nodes are

---

[5]http://protege.stanford.edu/
[6]http://owl.cs.manchester.ac.uk/explanation/
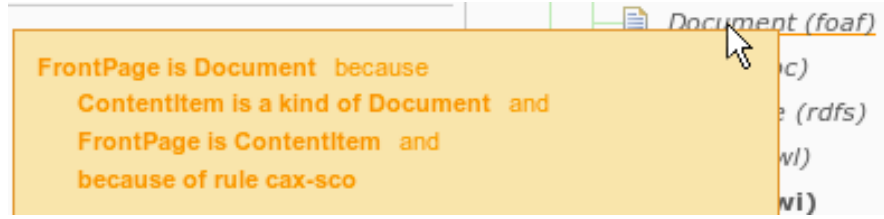[7]http://www.kiwi-project.eu/

Figure 11: An example of textual explanation in KiWi.

the triple ids corresponding to the triples in KiWi. Support nodes represent justifications. A selected derivation path is highlighted in yellow. A textual explanation of a selected derivation is shown on the right side.
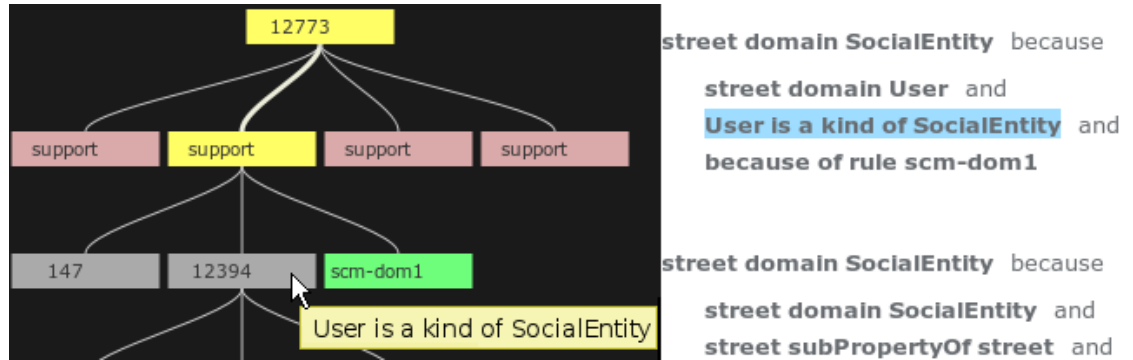

Figure 12: An example of interactive explanation tree in KiWi.

4.7   KOIOS Semantic Search Engine

Forcher *et al.* [9] describe the realization of EASD approach in a semantic search engine called KOIOS. The KOIOS semantic search engine enables keyword-based search on RDF data. The KOIOS search engine first computes a set of relevant SPARQL queries from a set of given keywords. Users then select the appropriate queries to query a triple store containing RDF data. The search results are provided with explanations about how they are computed. The explanations justify how keywords are mapped to concepts and how concepts are connected. In addition, the explanations interpret the performed queries in an understandable way. The authors introduce a set of ontologies to formally describe the content of explanations provided by KOIOS. The KOIOS Process Language (KPL) is used to describe the behavior of the originator. The Mathematical Graph Language (MGL) is used to realise the graph based view of the process model. Finally, another ontology called VGL is used for visualizing graph based information. In addition, KOIOS includes a set of rules to transform a certain model described in RDF. The trace model is described in RDF and transformed step-by-step to a presentation model using a set of rules to provide different views of explanation. Figure 13 depicts a keyword-based search for everything that Barak Obama wrote. The left white panel shows the graphical representation of one of the corresponding SPARQL queries of the search, and the right white panel shows the

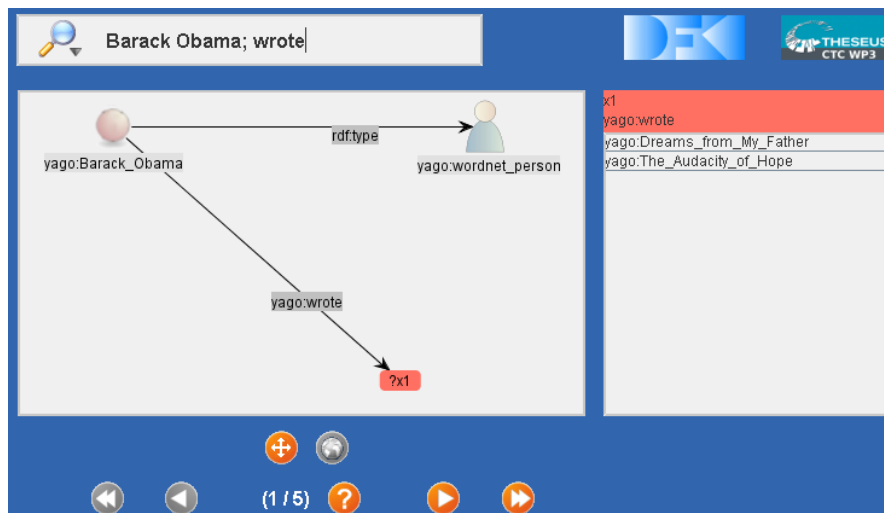results of the query. Users also can get a textual explanation of any concept from the graphical

Figure 13: Graphical User Interface of KOIOS.

representation of the query by clicking on any concept. KOIOS also provide justification-based explanation. Figure 14 shows an example of justification-based explanation in KOIOS explaining keyword to type mapping.
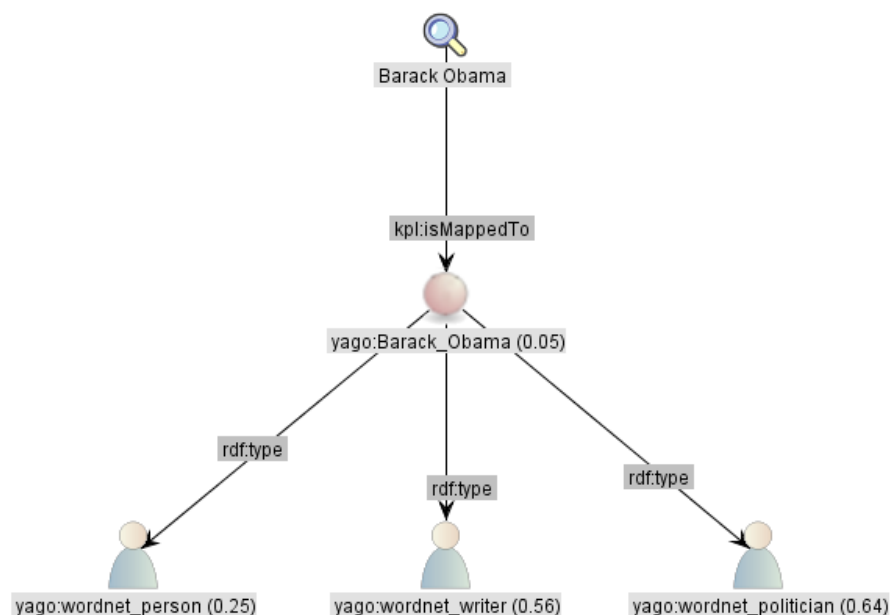

Figure 14: Justification of keyword mapping.

4.8   Explanation in AIR

The AIR reasoner annotates all the performed actions and dependencies between these actions. These annotations are recursively converted to AIR justifications. AIR provides a feature to enable explanation in natural language. Rule authors can specify a natural language description in a definition of rule itself. These natural language descriptions can contain variables. These variable values are replaced with the current value during the reasoning process. AIR also provide a feature to declaratively modify justifications. This allows the degree of details in AIR justifications to be selectively controlled. The authors of AIR suggest registering AIR with the Inference Web infrastructure in order to provide explanations from AIR justifications. This provides the possibility to use the Inference Web explanation toolkit for AIR justifications.

5   Discussion

The research we have reviewed exposes several dimensions of explanation in the context of the Semantic Web:

1. **Infrastructure:** With the increasing growth of sharing Semantic Web data as part of Linked Data [5] initiatives, it is important that data publishers can publish their data along with explanation related metadata with ease. Explanation infrastructures should be able to accommodate common data publishing principle. Semantic Web explanation infrastructures should also address heterogeneous and distributed nature of the Web. Inference Web intent to address these issues. For example, explanation metadata can be described in PML documents and resolved using the URIs of the documents. However, Inference Web provides a centralized solution for publishing and consuming explanation metadata. Explanation metadata should be registered in the Inference Web repository to use their facilities. Moreover, the published metadata using Inference Web facilities have compatibility issues with Linked Data principles. For instance, not all the resources in PML documents are identifiable by URIs as there are blank nodes in PML documents. Our ongoing work [11] on applying the Linked Data principles intent to address these issues. With regard to diversity of different representation, explanation metadata should be published promoting interoperability. The W3C PROV-DM data model [22] can be used as an interchange data model across different systems in this regard. Different systems can define their explanation metadata model as application-specific and domain-specific extensions of PROV-DM. Applications across the Web can then make sense of explanation metadata in a unified manner. Consumers of these explanation metadata can use explanation presentation and visualization tools according to their needs.

2. **Target:** Human users and software agents both are target of explanation in Semantic Web applications. In the existing approaches, human users are provided with natural language explanations or graphical explanations promoting easy comprehension. Unlike the expert systems which were used by knowledgeable users and domain experts, the users of Semantic Web applications can have different background, skill level, and knowledge level because of the open nature of the Web. Level of user expertise should be taken into account while providing explanation. The presentations of explanations can change according to level of user expertise or context of user scenario. User profiling approaches might be applied to provide explanations addressing this issue. With regard to software agents, explanations should be described using open standards such as OWL ontologies. As pointed out previously, explanations also should be published using common data publishing principles such as Linked Data to enable external software agents to easily consume them.

	PML	PML-Lite	AIRJ	KOIOS	EXPL
Proof tree	Yes	Yes	Yes	No	No
Process description	Yes	Yes	Yes	Yes	No
Provenance	How, when, who	How, when, who, where	How, when, who, where	No information available	No
Granularity	Not strictly defined	Not strictly defined	Coarse grained /graph	No information available	Fine grained /triple
Presentation model	No	No	No	Yes	Yes
Trust information	Yes	No	No	No	No
Blank node	Yes	No	Yes	No information available	No

Table 1: Comparison of explanation vocabularies

3. **What is explained:** The existing research discuss explanation of information manipulation steps, operations, and proof trees of derived results. Additional provenance information such as how, when, and who provenance is provided in Inference Web explanations for more context and enable better understanding. Semantic Web applications use distributed interconnected data in their reasoning process. Explaining the network of data used in the reasoning process might be useful for users. This would enable users to understand the flow of information used in the reasoning process and have a better understanding to the data integration process performed by the reasoner. Explanation with details of complex computation processes might always not be as useful for non-expert users as they are for expert users. Exposing problem solving methods in certain scenarios might result in security threat. The existing research does not discuss how explanations, which expose problem solving methods, influence security and confidentiality of Semantic Web systems.

4. **Representation:** The reviewed vocabularies to represent explanation metadata for machine consumption allow to describe proof trees for answers, processes used to compute answers, different types of provenance information, model for how explanations should be presented to human users, and trust related information. Other important aspects of explanation vocabularies are granularity and existence of blank nodes in the data described using them. Table 1 presents a comparison of reviewed vocabularies taking these aspect into account. Proof trees encode logical deduction of conclusions. PML, PML-Lite, and AIRJ provide primitives to encode proof trees as justifications of answers. KOIOS and EXPL vocabularies do not provide primitives for encoding proof trees as they concern mainly describing computation process and structure of presentation information. Process description concerns describing the information manipulation steps or the algorithms that compute answers. All the reviewed vocabularies except EXPL allow describing processes used to compute answers. Provenance information provide additional context to explanation. PML allow describing how, when, and who provenance. PML-Lite allows

one additional provenance, location provenance. AIRJ inherits PML-Lite provenance features. KOIOS and EXPL do not provide any primitive to describe provenance. With respect to granularity, RDF statements can be made at several level of granularity such as triple or graph. PML, PML-Lite, and AIRJ do not strictly define their granularity. PML *pml:Information* class instances can refer to a triple, a graph URI, or even to a textual representation of a logical formula. PML-Lite and AIRJ follow similar approach. For instance, *pmll:outputdata* property can point to a graph represented by a set of triples. In [16], the authors introduce new AIRJ properties such as *airj:matchedgraph* to specifically make statements about graphs. The authors of KOIOS vocabularies do not provide details about granularity [9]. EXPL uses RDF reification primitives to provide a triple level fine grained granularity. Presentation model allows describing how and what should presented to the human users as explanation. KOIOS provide VGL vocabulary to describe visualization of explanation. EXPL allows describing the structure and contents of different parts of explanations that are presented to human users. PML, PML-Lite, and AIRJ do not provide any primitive to describe presentation of explanation. Declaratively specifying presentation model enables different types of user interface technologies to render explanation contents. One of the main motivation for providing explanation is user trust. Only PML allows describing trust related information. Explanation facilities should allow to describe, capture and processing over captured trust. As a Linked Data common practice, blank nodes are avoided while publishing data [13]. Blank nodes add additional complexities in data integration in a global dataspaces. It is not possible to make statements about blank nodes as they do not have identifiers. PML and AIRJ use RDF container concepts such as *NodeSetList*. RDF containers use blank nodes to connect a sequence of items [1]. This approach makes it difficult to publish the data described using PML and AIRJ as Linked Data. In our ongoing work in [11], we present the Ratio4TA vocabulary to represent justifications supporting graph level granularity. Graph level granularity gives a flexible control as a graph can contain a single triple or many triples. Our next step in Ratio4TA would be to define it as an extension of W3C PROV-DM to enable better interoperability.

5. **Presentation:** Explanations are presented to human users as natural language or as graphical explanation in the reviewed approaches. Different kinds of graphical representation have been used to present proof tree, user query, or the steps performed by information manipulation algorithms. As discussed previously, users with different level of expertise should be considered for Semantic Web applications. How to present complex information manipulation processes to the end users in an understandable way and how much details is useful in the context of Semantic Web need to be researched more. The EASD approach discuss providing context dependent explanations. Existing approaches such as [7] on context-aware data consumption can be also applied to address providing different types of explanation depending on different types of users. Presentation models can be declaratively defined and associated with different context related information such as different user profiles. Different user interface rendering methodologies can then be applied on the defined presentation models to provide the final explanation user interfaces.

6. **Interaction:** Explanations should be provided with navigation support to enable end users to discover more related information. Other interaction models such as follow up or feedback mechanisms might also be useful in certain contexts. Inference Web provides explanation with navigation and follow up support. How users can interact trust information based on explanation would be another interesting area to explore.

7. **Trust:** The reviewed research lacks studies about understanding how explanations influ-

ence user trust in the context of Semantic Web. In contrast to the expert systems, Semantic Web applications have new dimensions such as openness and distributed. Furthermore, Semantic Web applications have much broader and diverse user base than expert systems. How these aspects of Semantic Web influence trust needs to be studied more. Explaining the reasoning process might not be enough in the context of Semantic Web. In the existing work, only Inference Web provides an infrastructure for trust which includes a vocabulary to describe trust related information. However, how users trust can be captured and processed for further trust assertions is not discussed. Another interesting area to explore would be how explanation approaches can be applied to explain trust itself. For instance, explanation can be provided about who have trusted a given resource or about trust rating calculations.

6 Conclusion

In this paper, we have presented an overview of the design principles of explanation-aware Semantic Web systems, what kind of explanation metadata these systems use to provide explanation and how they represent these metadata, and finally how explanation is generated and presented to end users. We have also discussed the important aspects of ongoing research relating to explanation in the context of Semantic Web.

Acknowledgments

References

[1] Rdf semantics. W3C recommendation (Feb 2004), `http://www.w3.org/TR/rdf-mt/`

[2] Angele, J., Moench, E., Oppermann, H., Staab, S., Wenke, D.: Ontology-based query and answering in chemistry: Ontonova project halo. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) The Semantic Web - ISWC 2003, Lecture Notes in Computer Science, vol. 2870, pp. 913–928. Springer Berlin / Heidelberg (2003)

[3] Antoniou, G., Bikakis, A., Dimaresis, N., Genetzakis, M., Georgalis, G., Governatori, G., Karouzaki, E., Kazepis, N., Kosmadakis, D., Kritsotakis, M., Lilis, G., Papadogiannakis, A., Pediaditis, P., Terzakis, C., Theodosaki, R., Zeginis, D.: Proof explanation for the semantic web using defeasible logic. In: Zhang, Z., Siekmann, J. (eds.) Knowledge Science, Engineering and Management, Lecture Notes in Computer Science, vol. 4798, pp. 186–197. Springer Berlin / Heidelberg (2007)

[4] Bassiliades, N., Antoniou, G., Governatori, G.: Proof explanation in the dr-device system. Web Reasoning and Rule Systems pp. 249–258 (2007)

[5] Berners-Lee, T.: Linked data. W3C Design Issues `http://www.w3.org/DesignIssues/LinkedData.html` (2006), [Online; accessed 24-May-2012]

[6] Bizer, C.: Quality-Driven Information Filtering in the Context of Web-Based Information Systems. Ph.D. thesis, Freie Universität Berlin, Universitätsbibliothek (2007)

[7] Costabello, L.: Dc proposal: Prissma, towards mobile adaptive presentation of the web of data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) The Semantic Web - ISWC 2011, Lecture Notes in Computer Science, vol. 7032, pp. 269–276. Springer Berlin / Heidelberg (2011)

[8] Doyle, D., Tsymbal, A., Cunningham, P.: A review of explanation and explanation in case-based reasoning. Tech. Rep. TCD-CS-2003-41, Trinity College Dublin (2003)

[9] Forcher, B., Sintek, M., Roth-Berghofer, T., Dengel, A.: Explanation-aware system design of the semantic search engine koios. In: Proceedings of the ECAI-10 workshop on Explanation-aware Computing(ExACt 2010) (2010)

[10] Glass, A.: Explanation of Adaptive Systems. Ph.D. thesis, Stanford University (2011)

[11] Hasan, R., Gandon, F.: Linking justifications in the collaborative semantic web applications. In: Proceedings of the 21st international conference companion on World Wide Web. pp. 1083–1090. WWW '12 Companion, ACM, New York, NY, USA (2012), `http://doi.acm.org/10.1145/2187980.2188245`

[12] Haynes, S.: Explanation in Information Systems: A Design Rationale Approach. Ph.D. thesis, The London School of Economics (2001)

[13] Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool, 1st edn. (2011), `http://linkeddatabook.com/`

[14] Horridge, M., Parsia, B., Sattler, U.: Laconic and precise justifications in owl. In: Proceedings of the 7th International Conference on The Semantic Web. pp. 323–338. ISWC '08, Springer-Verlag, Berlin, Heidelberg (2008), `http://dx.doi.org/10.1007/978-3-540-88564-1_21`

[15] Kagal, L., Jacobi, I., Khandelwal, A.: Gasping for air-why we need linked rules and justifications on the semantic web. Tech. Rep. MIT-CSAIL-TR-2011-023, Massachusetts Institute of Technology (2011), `http://hdl.handle.net/1721.1/62294`

[16] Khandelwal, A., Ding, L., Jacobi, I., Kagal, L., McGuinness, D.: Pml based air justification. `http://tw.rpi.edu/proj/tami/PML_Based_AIR_Justification` (2011), [Online; accessed 24-May-2012]

[17] Kotowski, J., Bry, F.: A perfect match for reasoning, explanation and reason maintenance: Owl 2 rl and semantic wikis. In: Proceedings of 5th Semantic Wiki Workshop, Hersonissos, Crete, Greece (31st May 2010) (2010), `http://www.pms.ifi.lmu.de/publikationen/#PMS-FB-2010-5`

[18] McGuinness, D.L., Ding, L., Glass, A., Chang, C., Zeng, H., Furtado, V.: Explanation interfaces for the semantic web: Issues and models. In: Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI'06) (2006)

[19] McGuinness, D.L., Furtado, V., Pinheiro da Silva, P., Ding, L., Glass, A., Chang, C.: Explaining semantic web applications. In: Semantic Web Engineering in the Knowledge Society (2008), http://www.igi-global.com/reference/details.asp?id=8276

[20] McGuinness, D.L., da Silva, P.P.: Explaining answers from the semantic web: the inference web approach. Web Semantics: Science, Services and Agents on the World Wide Web 1(4), 397 – 413 (2004), `http://www.sciencedirect.com/science/article/pii/S1570826804000083`, international Semantic Web Conference 2003

[21] Moore, J., Swartout, W.: Explanation in expert systemss: A survey. Tech. rep., University of Southern California (December 1988)

[22] Moreau, L., Missier, P.: Prov-dm: The prov data model. World Wide Web Consortium, Fourth Public Working Draft (2012)

[23] Roth-Berghofer, T.R.: Explanations and case-based reasoning: Foundational issues. In: Funk, P., Calero, P.A.G. (eds.) Advances in Case-Based Reasoning. pp. 389–403. Springer-Verlag (September 2004)

[24] Pinheiro da Silva, P., McGuinness, D.L., Del Rio, N., Ding, L.: Inference web in action: Lightweight use of the proof markup language. In: Proceedings of the 7th International Semantic Web Conference (ISWC'08). pp. 847–860 (2008)

[25] Pinheiro da Silva, P., McGuinness, D.L., Fikes, R.: A proof markup language for semantic web services. Information Systems 31(4-5), 381–395 (2006), also KSL Tech Report KSL-04-01, January, 2004

[26] Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., Drummond, R., Herlocker, J.: Toward harnessing user feedback for machine learning. In: Proceedings of the 12th international conference on Intelligent user interfaces. pp. 82–91. IUI '07, ACM, New York, NY, USA (2007), `http://doi.acm.org/10.1145/1216295.1216316`

[27] Swartout, W., Paris, C., Moore, J.: Explanations in knowledge systems: design for explainable expert systems. IEEE Expert 6(3), 58 –64 (June 1991)

[28] Tintarev, N., Masthoff, J.: A survey of explanations in recommender systems. In: Data Engineering Workshop, 2007 IEEE 23rd International Conference on. pp. 801 –810 (April 2007)