Wine Agent: Semantic Web Testbed Application

Eric I. Hsu and Deborah L. McGuinness
Knowledge Systems Laboratory
Stanford University
{ehsu, dlm}@ksl.stanford.edu

Abstract

The Wine Agent is a demonstration system that uses an underlying domain ontology to provide suitable wines for a given meal. In doing so it serves as a testbed, not only for the logical domain description, but additionally for emerging Semantic Web technologies that process, infer, justify, and execute the pairings. Specifically, it combines the the DAML+OIL and OWL Web-based description logics with the JTP theorem prover. The resulting knowledge base can be queried remotely via the DQL query language. Suitable pairings are explained within the Inference Web apparatus, and then transacted via a preliminary implementation of Web Services. Besides serving as a prototype for these methodologies, the wine agent has provided useful empirical lessons regarding reasoning via Semantic Web axioms, language requirements, and requirements for explanation, as well as pragmatic issues concerning implementation and integration.

1 Overview

The Wine Agent is accessed over the Web, at http://onto.stanford.edu:8080

The main interface, shown in Figure 1, allows the user to select a type of course that will be served from a mock-up menu. Alternatively, they can select a specific individual food, in which case the Wine Agent will make an additional set of inferences in order to determine the type of course for that food, and then proceed as if the user had chosen that type.

The result of one such interaction, where the user has chosen a pasta with spicy red sauce, is shown in Figure 1. The Wine Agent lists the requisite properties of a suitable wine, in terms of the wine's color, sugar, body, and flavor. This statement is appended with a link to an inference web explanation, and then followed by a listing of individual wines meeting the desired characteristics. A link labeled "Web Inventory Search" instructs the agent to search various external sites for these wines, offering them for direct purchase. In case this provides an inadequate number of options, the Wine Agent also identifies general varietals featuring the targeted characteristics. Each is displayed as a link to a specific merchant, wine.com, whose inventory of that specific varietal can be browsed or searched by the user.

The Wine Agent employs a suite of emerging Semantic Web formalisms in order to derive this functionality from a core ontology that describes the general properties of meals, foods, and wines. The ontology is expressed using the description logics

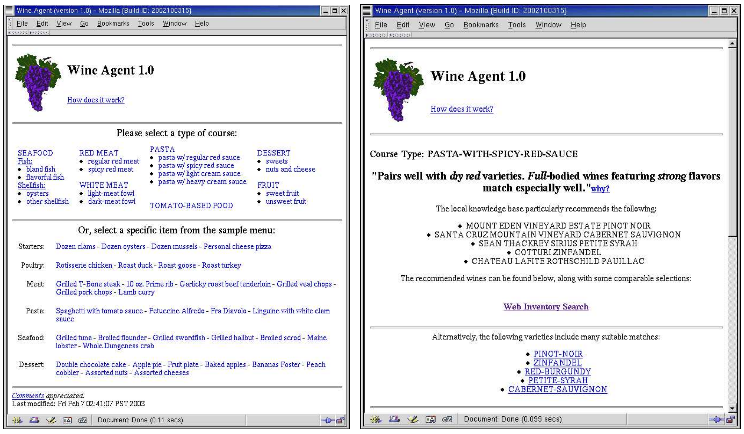

Figure 1: Wine Agent Front Page and Example Response.

DAML+OIL and OWL, which are designed for distribution over the Web via namespaces and a syntax based on mark-up. Using DQL, or “DAML Query Language”, to express and transact a series of queries, the Wine Agent initiates inference by the JTP theorem prover. JTP uses set of DAML+OIL/OWL axioms to interpret the ontology as first-order-logic and perform model-elimination theorem proving. Besides answering queries, it also outputs proofs for use with the Inference Web, which is a system for registering explanations for inferences made by various systems distributed across the Web. Finally, the links to external sites are accomplished via an incomplete implementation of Web Services.

The following sections will detail each of these steps in turn, followed by discussion and conclusions.

2 Domain Ontology: DAML+OIL/OWL

The domain ontology is general-purpose, usable by any system wishing to reason about foods and wines. Here, though, the relevant terms include meal courses pairing foods with drinks, specializations for the various types of food, varietals of wines, individual wines, and wine properties, in addition to various subsidiary definitions. Figure 2 shows an excerpt of a description for meal courses featuring pastas with spicy red sauces; individual food items like “Fra Diavolo” might reference this class as their type.

The first portion of the definition identifies the term as a type of meal course whose food item is constrained to be some sort of pasta with spicy red sauce, via a `<daml:Restriction>`. Alternatively, the definition ends with restrictions on the properties of a any drink that might be associated with such a course, without referencing their properties directly. Rather, in this logic restrictions can be reified, and the above can be interpreted as constraining all pasta courses featuring spicy red sauce to be a subclasses of all meal courses meeting the required constraints.

On the other side of the match, wines are represented as individuals. Suppose some wine has been defined as a Pauillac whose maker is Chateau Lafite Rothschild. Together with other statements in the ontology, this allows the reasoner to deduce

```
<rdfs:Class rdf:ID="PASTA-WITH-SPICY-RED-SAUCE-COURSE">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <rdfs:Class rdf:about="#MEAL-COURSE"/>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#FOOD"/>
      <daml:toClass rdf:resource="#PASTA-WITH-SPICY-RED-SAUCE"/>
    </daml:Restriction>
  </daml:intersectionOf>
  <rdfs:subClassOf
    rdf:resource="#DRINK-HAS-RED-COLOR-RESTRICTION"/>
  <rdfs:subClassOf
    rdf:resource="#DRINK-HAS-FULL-BODY-RESTRICTION"/>
  <rdfs:subClassOf
    rdf:resource="#DRINK-HAS-STRONG-FLAVOR-RESTRICTION"/>
  <rdfs:subClassOf
    rdf:resource="#DRINK-HAS-DRY-SUGAR-RESTRICTION"/>
</rdfs:Class>
```

Figure 2: Ontology Excerpt: Pasta with Spicy Red Sauce.

many additional facts: that this a Medoc wine from Bordeaux, in France, and that it is red, to name a few. The definition for all Pauillacs is is shown in Figure 3.

```
rdfs:Class rdf:ID="PAUILLAC">
  <rdfs:subClassOf rdf:resource="#FULL-BODY-RESTRICTION"/>
  <rdfs:subClassOf rdf:resource="#STRONG-FLAVOR-RESTRICTION"/>
  <rdfs:subClassOf rdf:resource=
   "#CABERNET-SAUVIGNON-INDIVIDUAL-GRAPE-SLOT-RESTRICTION"/>
  <rdfs:subClassOf rdf:resource=
   "#GRAPE-SLOT-MAX-CARDINALITY-1-RESTRICTION"/>
  <daml:intersectionOf rdf:parseType="daml:collection">
     <rdfs:Class rdf:about="#MEDOC"/>
     <daml:Restriction>
        <daml:onProperty rdf:resource="#REGION"/>
        <daml:hasValue rdf:resource="#PAUILLAC-INDIVIDUAL"/>
     </daml:Restriction>
  </daml:intersectionOf>
</rdfs:Class>
```

Figure 3: Ontology Excerpt: Pauillac.

Most of the wine's distinguishing properties are specified here; one exception is that its red color arises from being a Medoc. In addition, wines of this type must be unblended, i.e. made from only one varietal of grape, cabernet sauvignon. In addition they must be grown in the Pauillac region. Again, thanks to namespacing, these terms are all available for reference by any system across the Web, not just the Wine Agent.

3 Reasoning: JTP

The next step is to infer the appropriate match using JTP, which is a general-purpose theorem prover developed at the Stanford Knowledge Systems Lab. Using standard XML parsers, it interprets the DAML+OIL/OWL ontology as a series of first-order logical statements. Using an added set of Axioms capturing the meaning of such statements, JTP can thus seek to infer arbitrary sentences entailed by the KB.

In practice, the axiomatization is extremely difficult, not only to conceive, but also to test throughly. One example is the interpretation of disjunctive restrictions, for instance the notion that a particular course requires wines with full *or* medium bodies. Such cases were sometimes handled incorrectly due to subtleties in the axiomaization. Further, this problem was not discovered until the completion of the initial version of the Wine Agent.

4 Querying the KB: DQL

The DAML Query Language can be though of as a combination of SQL and TCP; it is designed to query remote knowledge bases, and additionally to manage the transmission of responses and refinements during a query-answering session. That is, it allows for incomplete answer sets, requests for further answers, discontinuous sessions, and additional functionality useful for controlling the semantic level of a client/server interaction.

A typical query first arrives at the server hosting the wines KB via DQL. Wine Agent specifies the premises for the query concerning a hypothetical wine being paired with the food in question, and that the server must provide as many answers as it can find. The server replies with a preliminary (and possibly incomplete) set of answers, as well as a process handle in case the user wishes to send future messages requesting more answers. In conforming to DQL, JTP knows to interpret the premises as assertions that it must make before performing the queries, and then retract once the session terminates.

5 Justifying the Answers: Inference Web

Performing inference over the Web drives multiple needs for explanation. First is *provenance*: systems must verify that inferences and underlying descriptions come from trustworthy and recent sources. Another priority is *transparency*, the exposure of the employed reasoning methodology, accomplished here via browser navigation. *Interoperability* consititutes a long-term goal: the sharing and composition of proofs from distributed, heterogeneous collections of reasoners. Finally, *reasoning abstraction* is a necessity for breaking proofs into fragments that human users can understand.

The Inference Web project seeks to meet these challenges via a registry of proofs and provers, as well as an inference browser for traversing their explanations. As previously mentioned, the wine agent participates by outputting proof explanations in the required format, caching them, registering them, and then linking to them through the IW browser.

6 Performing Transactions: Web Services

The final step for each Wine Agent session is to link to external merchants, both for individual wines and for varietals. The process of querying their inventories and performing purchase transactions can be seen as a matter of Web Services. However, because the merchants in question do not yet use these emerging standards, the implementation is incomplete.

Specifically, our system identifies matching varietals and wines via their names within the web-accessible general-purpose ontology. If wine.com were to use the same lexicon, the two sites could build up a protocol of seeking, displaying, and selling targeted wines by direct reference. Instead, the Wine Agent relies on existing syntactic methods by scraping screens from the merchants and linking into their search engines with targeted parameters already in place, via HTML "GET" formats. In the future, hopefully, this will not be the case!

7 Conclusions

Pedagogically, the Wine Agent prototype is an operational specification for integrating emerging standards for web query languages (DQL), description logics (DAML+OIL), distributed web explanation (Inference Web), and services markup languages (DAML-S) to build an integrated web agent. The system is motivated by an example application in the OWL Guide [6] that is in turn grounded in work on the CLASSIC description logic [2].

Though a successful integration in the end, the prototype was much more difficult to develop than originally expected. One reason is that the information web sites used by the system are not marked up with DAML-S. Even upon switching to screen-scraping, such sites still present a challenge because they do not provide as much representational detail as does the ontology. The project required additional integration work between the JTP reasoner, the DQL query langauage, and the axiomatization of the employed description logics. Finally, preformance issues by the reasoner necessitated the caching of popular and time-consuming queries.

8 Bibliography

[1] Merryll K. Abrahams, Deborah L. McGuinness, Rich Thomason, et al. NeoClassic Tutorial: Version 1.0. Artificial Intelligence Research Department, AT&T Labs Research, 1996.

[2] Ronald J. Brachman, Alex Borgida, Deborah L. McGuinness, et al. Living with CLASSIC: When and How to Use a KL-ONE-Like Language. In *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, ed. John Sowa. San Mateo, California: Morgan Kaufmann, 1991, pages 401–456.

[3] Richard Fikes, Pat Hayes, Ian Horrocks, ed. DAML Query Language (DQL) Abstract Specification. August, 2002. http://www.ksl.stanford.edu/projects/dql.

[4] Deborah L. McGuinness and Paulo Pinheiro da Silva. Infrastructure for Web Explanations. To appear in *Proceedings of 2nd International Semantic Web Conference (ISWC2003)*, Sanibel, FL. Springer, October 2003.

[5] Sheila McIlraith and David Martin. Bringing Semantics to Web Services. *IEEE Intelligent Systems*, 18(1):90–93, January/February, 2003.

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