The Mobile Wine Agent: Pairing Wine with the Social Semantic Web

Evan W. Patton[1] and Deborah L. McGuinness[1]

Tetherless World Constellation
Rensselaer Polytechnic Institute
Department of Computer Science
110 8th Street
Troy, NY 12180
[pattoe, dlm]@cs.rpi.edu

Abstract. The Tetherless World Mobile Wine Agent is a Semantic Web application for making wine and food recommendations to users. In order to make the interface easier for users, instances are generated through a common interface that uses data from the underlying ontology to drive user interaction. The agent is being integrated with social applications, such as Facebook and Twitter, to allow users to leverage and share generated content with other individuals on the World Wide Web. Anyone can contribute data by constructing an RDF graph and making it available over the web in XML, which the agent will read and incorporate into its internal graph. Thus, users of the wine agent can access menus published in RDF on the web. By bringing together semantics, data sharing, and extensibility, the Mobile Wine Agent demonstrates how semantics and user interactions can work in tandem to grow the Semantic Web.

1 Introduction

The primary goal of the Semantic Web is to redefine how we store, access, and think about dat [1]. This new approach to data, juxtaposed with the proprietary APIs of social web applications such as Facebook and Twitter, requires developers and data providers alike to rethink the openness of data. The Tetherless World Mobile Wine Agent is an application which uses Semantic Web technologies to describe food and wine pairings while providing social interaction and recommendation sharing via current social web platforms such as blogging, Facebook, and Twitter.

In this paper, we will describe the Mobile Wine Agent's ontology-sensitive interface for generating instances and classes, the underlying command architecture system for the agent, and how the agent integrates with social web applications, specifically Twitter and Facebook. Additionally, we will highlight how to utilize existing web technologies to provide a seamless user experience with the Wine Agent.

Wine and food are a rich domain and the history of recommender systems goes back almost twenty years [2] and has evolved with the tools used to represent knowledge [3, 4]. Given the ubiquity of the mobile phone and the projected growth of mobile devices as a computing platform in developing nations, it is critical to the growth of the Semantic Web that semantically-enabled applications be capable of running on small, low-powered devices. The first half of the decade saw recommender systems moving into this space (see [5, 6]) along with the fundamental underpinnings for storing context-based information in OWL [7]. More recently, semantically enabled systems (see [8]) have been taking advantage of GPS data to provide context-sensitive recommendations to users. This new iteration of the Wine Agent demonstrates how an existing Semantic Web application can be adapted to reflect dynamic changes in methods of social interaction and how moving such applications to mobile devices can enrich the user experience through context-sensitive reasoning.

2 Wine Agent Architecture

The Mobile Wine Agent expands on the food and wine ontologies which were released as part of the World Wide Web Consortium's OWL 1.0 specification [9], which themselves are OWL representations of the ontology which powers the Knowledge Systems Laboratory's Wine Agent [3] which evolved from the ontology developed for an early description logic introduction [2] and a tutorial and intelligent tutoring exercise [10]. Significant work has been done to update these ontologies with references to Linked Data[1] through Geonames[2] and Freebase[3]. In addition to these updated ontologies, the agent provides another core ontology which describes the commands it accepts (see Sec. 8) and the output it generates as RDF. The server itself runs Jena [11] and Pellet [12] with incremental reasoning [13] enabled to increase performance. Any semantically-enabled application can use the RESTful interface of the Wine Agent server using the agent ontology over HTTP to gain access to its data and reasoning capabilities.

The mobile client application currently runs on Apple's iPhone® and iPod touch® platforms (see Fig. 1) with an Android® version planned. The earlier iterations of these devices have a 400 MHz process with 80 MB available RAM. Due to these computational limitations, the system was designed such that basic reasoning could be performed locally on the device while any complex reasoning would take place on an external server. The device keeps a cached copy of the ontology stored locally in RDF to improve startup times and data access times. The current version of the ontology contains ∼5500 triples and the reasoner takes an average of 1572 ms (σ=12 ms) to load the ontology from NAND and consumes 827 KB of memory. The primary reason for keeping the ontology on the device is to drive the user interface to assist users in generating food and wine descriptions (see Sec. 4). The resulting interface is ontology-aware and can use information present to simplify user interaction.

Performing the full breadth of OWL reasoning is not currently supported by the client's reasoner. Complicated processing can be costly in terms of battery life

[1] http://linkeddata.org/

[2] http://www.geonames.org/

[3] http://www.freebase.com/

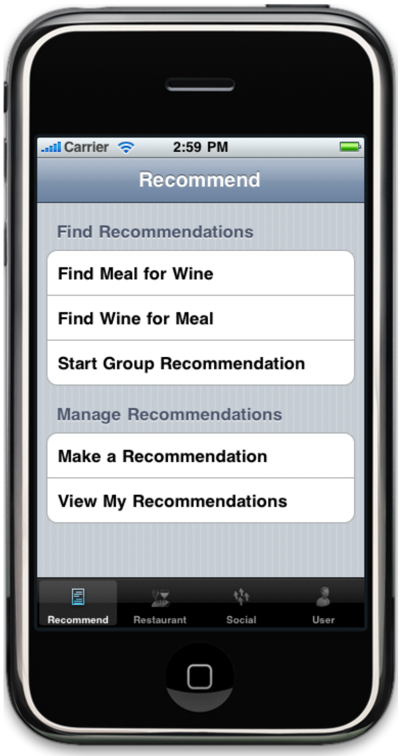


Fig. 1. Mobile Wine Agent for the iPhone.

and responsiveness of the device. Therefore, it is necessary to offload complicated reasoning tasks (such as finding appropriate recommendations) to the remote server. When external reasoning is required, the client transmits a list of URLs for the server to load and then issues an appropriate command to tell the server what information the client needs to continue. Additionally, the user's food/wine preferences are pushed to the server if they have not been made available for sharing.

3 The Ontologies

The Mobile Wine Agent relies on three core ontologies which describe the agent, food, and wine. While the wine ontology is substantially unchanged from its release as part of the OWL Specification (see [9]), the food ontology is substantially changed[4]. We have extended the food ontology to contain many more classes and

[4] The extended ontologies used by the Wine Agent can be found at: `http://wineagent.tw.rpi.edu/data/agent.rdf`

instances as well as added associations between foods and what they are made from. For example, pork products, such as bacon and ham, are declared to be made from pigs. This increased specificity affords users who do not to eat pork-based products to filter out dishes containing these ingredients. Sauces have also been separated from foods so that restauranteurs can identify specific sauces that are then realized as members of sauce classes. These classes can then be used to describe dishes, instead of being fixed to the class hierarchy as was the case in the old ontology. Flavor profiles have also been added so that individuals can filter out bitter or spicy foods depending on their particular palette. The result is a more expressive ontology to help make it easier for individuals to describe foods to the Wine Agent.

Often services use WSDL [14] with SOAP [15] bindings for interaction and require an additional ontology such as DAML-S [16], OWL-S [17], or the Web Services Markup Framework [18] to interpret the information in a Semantic Web context. To provide an RDF-friendly approach to interacting with the Wine Agent, there is an agent ontology which describes the commands, responses, errors, and other objects critical to the storage and communication of information the agent contains. Thus, all input and output from the Wine Agent are transmitted as RDF in XML (see Sec. 8) without requiring additional mappings from WSDL and SOAP to RDF and OWL.

Every command that the agent processes is a descendant of the `agent:Command` class. When the agent processes a command, it will return an `agent:Response`. A successful response will contain any necessary information for the client to use as a result of the command. In cases where processing fails, `agent:Error`, a subclass of `agent:Response` is returned with a human-readable string describing the error that occurred. Other classes include `agent:RecommendationList`, a subclass of `rdf:Seq`, which stores recommendations and `agent:Recommendation` which pairs dish classes with wine classes or dish instances with wine instances.

4 Ontology-sensitive Interfaces

Due to their small size, mobile devices require cleaner, more structured interfaces than their desktop counterparts and given the complex domain the Wine Agent operates over, it was important to construct a highly dynamic interface which could be used to describe foods and wines. In order to accomplish this goal, an RDFS reasoner that understands all six OWL restrictions and `intersectionOf` was built into the Mobile Wine Agent. This reasoner is utilized whenever the user describes a dish or wine to the agent.

When the user chooses to describe, for example, a wine for a meal, the client requests the class `vin:Wine` from the reasoner and passes it to the description generator interface. The generator then asks the reasoner for a list of properties that might apply to this class, which will include any property that has as its

`http://wineagent.tw.rpi.edu/data/food.rdf`

`http://wineagent.tw.rpi.edu/data/wine.rdf` rdfs:domain vin:Wine or any of its superclasses, as well as properties which lack a value on rdfs:domain. The user is presented with a picker view[5] listing the properties. When the user selects a property, the interface is adjusted based on whether the rdfs:range is the class owl:Class or not. In the event the range of the property is owl:Class an additional entry in the display is added entitled **Union Of** which allows the user to construct an anonymous union class. Otherwise, two entries are added: **One Of** and **Any** which allow the user to specify that the value of a property must be a member of the specified class (either an anonymous enumerated class or a previously named class).

The behavior of the interface when selecting the value for a property is also conditional. The generator asks for the appropriate values for the property (or subclasses of the range in the case of the **Any** operation) to present them to the user. If the property is an owl:FunctionalProperty or the supplied class, vin:Wine in this case, is a subclass of an owl:Restriction on the property with an owl:maxCardinality of 1, then the generator displays another picker view to limit the property selection to a single element. If neither of these conditions applies, then a generic table view is displayed where the user can select one or more values. The resulting property/value pair(s) are rendered on an alternate screen which keeps a cumulative representation of the user's choices. Upon completion, the description generator returns either a string with the necessary RDF to describe the individual created or an owl:Class which will match any individual with the specified properties.

5 Location-aware Services

Since the underlying platform for the Mobile Wine Agent is different from a standard desktop application, it can take advantage of features unique to mobile devices. Users can activate location updates via the Global Positioning System (GPS) for the Wine Agent. Once the agent receives information about the user's location, it presents a map with known restaurants serving their menus in RDF[6] (see Fig. 2). The user selects the appropriate restaurant, and the system will retrieve the menu and wine list for the reasoner to use. These lists are presented to the user so they can pick the appropriate meal or wine instead of manually inputting a description using the interface in Sec. 4. Restaurant selection is also used to increase reasoning performance by reducing the ABox to only contain dishes and wines available on the restaurants menu.

[5] The UIPickerView is a component of the iPhone SDK and allows for single-item selection from a list.

[6] We currently provide a number of RDF representations from HTML and PDF for some restaurants for demonstration purposes. Plans are underway to develop a Desktop Wine Agent which can be used by proprietors to generate RDF representations of menus and wine lists.

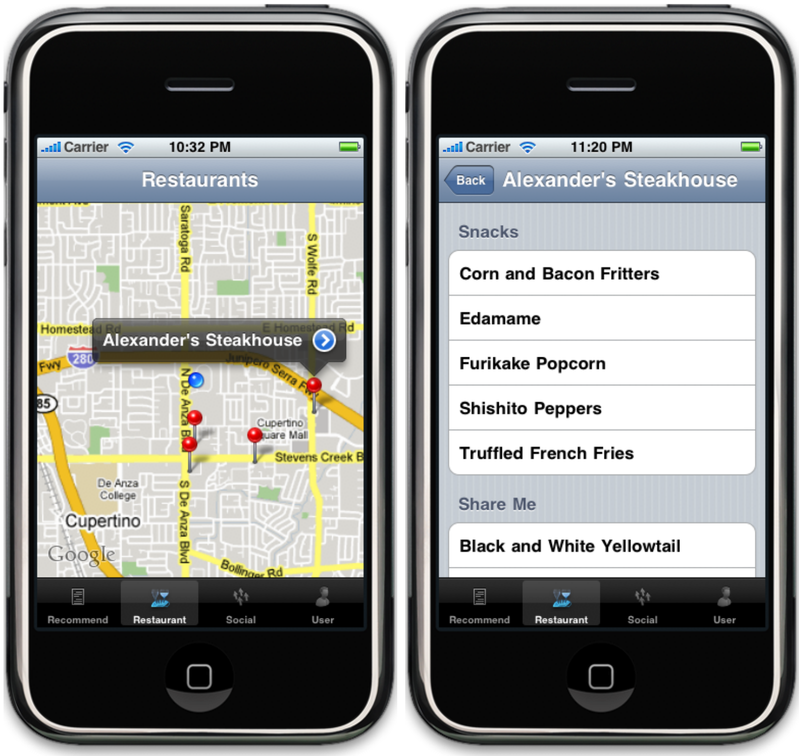


Fig. 2. Map view displaying restaurants near the user

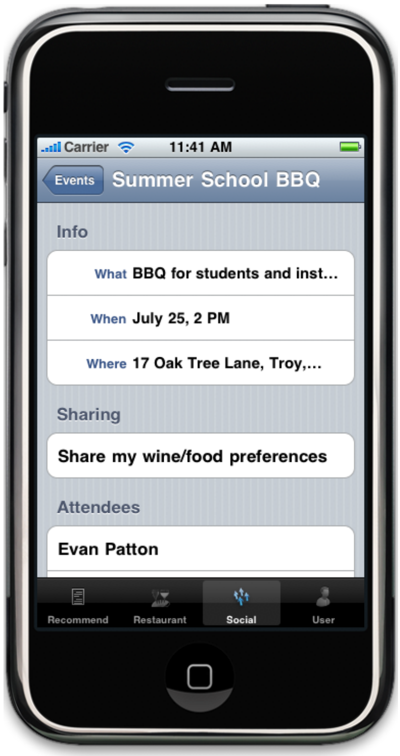
6 Social Interaction

The Wine Agent provides a variety of ways for users to interact with one another, both internally and through external applications. Interaction data is stored as RDF and as such can be queried by external applications. When users select a recommendation, for example, they can indicate whether they approve of the recommendation or not. The agent stores this information as an RDF triple and in the positive case prioritizes the recommendation and in the negative case removes it from future results for this individual. Recommendations and preferences can also be shared with other users via Facebook and Twitter.

6.1 Facebook

Facebook is the largest social networking community and its API is extremely flexible. We have integrated access to Facebook into the Wine Agent to provide a number of features. For users to enable Facebook for the Wine Agent, they must first add the Tetherless World Wine Agent application in Facebook. Once they grant permission to use their information in the application, they can log into Facebook through the Wine Agent. This will enable a number of features for obtaining and sharing content, such as sharing preferences and posting to their news feed.

First, users can allow others to read their personal preferences by granting permissions to friends or friend lists. The client application uploads the user's preferences to be stored on the server[7] and issues a command to the server to allow permission to the indicated users to access these new triples. When a friend wants to reason about food and wine using these shared preferences, they can choose to import them into their instance of the agent through the appropriate menu items. For privacy reasons, users can also revoke permissions to view personal preferences at any time.



Summer School BBQ

Info

What BBQ for students and inst...

When July 25, 2 PM

Where 17 Oak Tree Lane, Troy,...

Sharing

Share my wine/food preferences

Attendees

Evan Patton

Fig. 3. A Facebook event as seen by the Wine Agent

[7] Unlike recommendations and the core ontologies, these triples are not publicly available for privacy reasons.

Users can join events via Facebook and share their preferences with these events (see Fig. 3). Any attendees to the event using the Wine Agent can register a set of triples so that the event organizer can load all of the preferences of those attending into an instance of the Wine Agent for reasoning. These permissions are only granted for the particular event and thus are more restricted than the general sharing mechanism. In the future, this option will also allow for creating group recommendations via iPhone discovery (see Future Work for details).

To facilitate sharing of recommendations, dishes, and wines, the Wine Agent also ties into the Facebook feed, allowing users to post a message to their friends regarding any entry in the agent they find notable. These posts are accompanied with the appropriate URI so that users can launch the Wine Agent directly from Facebook.com.

6.2 Twitter

In addition to the integration with Facebook, the Wine Agent is begin integrated with Twitter so that users with Twitter accounts can post updates to their Twitter feeds about the dishes, wines, and recommendations they discover through the agent. The Wine Agent presents a text box for the user to enter a message and uses URL shortening tools to provide links to any resource the user identifies for the message (see Fig. 4). The text is then sent to the Twitter API using the OAuth security protocol. This allows users to share Semantic Web data with friends without having to understand the complexities of RDF and OWL.

7 From Web to Wine Agent

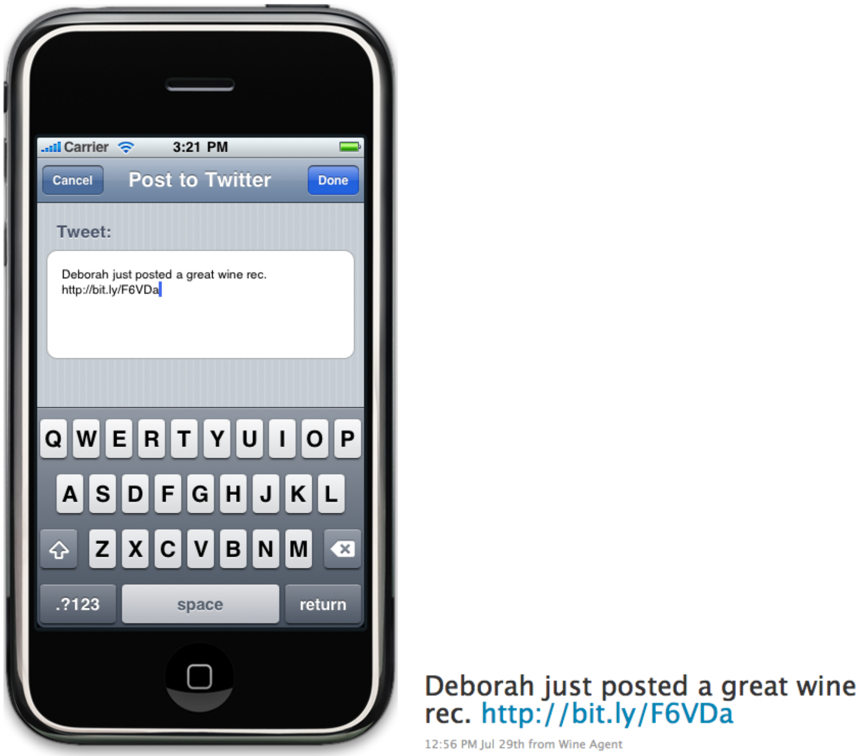
In order to facilitate interaction on the web and the sharing of data generated by the Mobile Wine Agent, the agent responds to a custom URI scheme and all recommendations and dish/wine classes generated by users are given unique URIs starting with this scheme. These URIs can be used as part of an anchor HTML tag:

```
<p>Check out this <a href="twwa://rec/dlm-5/">
recommendation</a> from the Wine Agent.
</p>
```

or with RDFa [19]:

```
<p about="http://tw.rpi.edu/wiki/Deborah_L._McGuinness">
Deborah posted an <a rel="agent:recommends"
href="twwa://rec/dlm-5/">amazing recommendation</a>
for dinner last night. </p>
```

When an iPhone user clicks on this link in Safari, the application will quit and launch the Wine Agent, passing it the URI which the agent can then use to look



Deborah just posted a great wine rec. http://bit.ly/F6VDa

Deborah just posted a great wine rec. http://bit.ly/F6VDa

12:56 PM Jul 29th from Wine Agent

Fig. 4. Twitter interface with resulting web post.

up dishes and wines in restaurants and across the web. For situations where tools auto-generate links based on an http:// prefix, the Wine Agent website hosts a translator tool which will redirect from an http:// URI to a twwa:// URI. This translation utility is integral to the agent's Twitter interface as it allows for URL shortening services to be used on custom URIs generated by the application. Restaurants can also take advantage of this custom URI scheme by using the twwa:// prefix with the path to the RDF versions of their menus on any website. Clicking on such a link will launch the agent, which will automatically retrieve and import the restaurant instance data for the user's convenience.

8 Using the Wine Agent as a Web Service

Many approaches have been taken to providing and describing web services. The Web Services Description Language [14] for HTTP or SOAP [15] provides a common framework for interacting with services. A number of ontologies have evolved for semantically describing these services, including DAML-S [16], the Web Service Modeling Framework (WSMF) [18], the Semantic Web Services

Language [20] and OWL-S [17]. We instead choose to provide an ontology which describes, as RDF, all of the methods one can call against the Wine Agent. This provides us the benefit that both the input and the output of the Wine Agent are in RDF, rather than providing one as WSDL+SOAP and another as RDF. A semantically enabled service may send the following RDF via HTTP POST to request:

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:agent="http://wineagent.tw.rpi.edu/data/agent.rdf#"
   xmlns="http://wineagent.tw.rpi.edu/data/agent.rdf#"
   >
   <CompoundCommand rdf:ID="Request">
      <rdf:_1>
         <LoadRestaurantMenuCommand>
            <username>jsmith</username>
            <sessionID></sessionID>
            <descriptor>
               <Menu rdf:about="http://wineagent.tw.rpi.edu/ \
               restaurants/www.loportos.com/menu.rdf"/>
            </descriptor>
         </LoadRestaurantMenuCommand>
      </rdf:_1>
      <rdf:_2>
         <RecommendWineForDish>
            <username>jsmith</username>
            <sessionID></sessionID>
            <descriptor>
               <Dish rdf:about="http://wineagent.tw.rpi.edu/ \
               restaurants/www.loportos.com/menu.rdf \
               #LinguiniConGamberetti">
            </descriptor>
         </RecommendWineForDish>
      </rdf:_2>
   </CompoundCommand>
</rdf:RDF>
```

Such a request would generate an `agent:Response` which would contain a number of recommendations which match the instance LinguiniConGamberetti. Using the XML Stylesheet Language Transformations (XSLT), this output could be translated into new input which might match the recommendation against LoPorto's wine list. This system is discussed in more detail on the Wine Agent's website[8].

[8] http://wineagent.tw.rpi.edu/

9 Conclusion

The Mobile Wine Agent is a complex Semantic Web application that integrates with a number of social platforms to enable sharing of user-generated semantic content. It provides a common user interface to generate descriptions of wines, dishes, and recommendations that is sensitive to the underlying ontology driving the application. The ontology and instance data can be extended by anyone, with a particular focus on restauranteurs and winery proprietors, to reflect the large variety of wines and dishes. Custom URIs can be linked to from anywhere on the web that are recognized by Safari as being associated with the Wine Agent, thus allowing links on blogs, Facebook, and Twitter to directly launch the agent and let users start working with the data immediately. The Wine Agent server can also act as a web service, accepting commands and returning results all described in RDF. While many more features could be added to the Wine Agent to extend its capabilities, the core structure, both the ontologies and the application itself, provide a rich Social Semantic Web experience.

10 Future Work

To further expand on the social aspects of the Wine Agent, a full OWL-DL capable reasoner is being developed to run on the iPhone. This will allow the phone to perform all of the necessary reasoning locally but will also facilitate the investigation of parallel reasoning across multiple devices using the iPhone's Bluetooth® interface. With this new capability, along with the Facebook Events integration as mentioned in Sec. 6.1, a group of friends could go out to dinner, import their dinner event, and have each of the devices talk to the others, sharing preferences and meal choices to pick the most appropriate wines for dinner.

Supporting provenance in the Wine Agent is another future goal. For example, if a wine critic such as Robert Parker began making recommendations using the Wine Agent, many individuals might want to know which recommendations are his and prefer those recommendations over others. Similarly, friends who are less wine literate may want to rely on recommendations from friends who are more wine literate when going out to dinner. Using data from Facebook, for example, could be one such way of building provenance into the Wine Agent.

References

1. Berners-Lee, T.: TED talks: Tim Berners-Lee on the next web (February 2009)
2. Brachman, R.J., McGuinness, D.L., Patel-Schneider, P.F., Resnick, L.A., Borgida, A.: Living with CLASSIC: When and how to use a KL-ONE-Like language. Principles of Semantic Networks: Explorations in the Representation of Knowledge (1991) 401–456
3. Hsu, E., McGuinness, D.L.: KSL Wine Agent: Semantic Web testbed application. In: Proceedings of the 16th International Workshop on Description Logics, International Workshop on Description Logics (September 2003)

4. Michaelis, J., Ding, L., McGuinness, D.L.: The TW Wine Agent: A social semantic web demo. In: Poster and Demo track of the 7th International Semantic Web Conference. (October 2008)
5. Miller, B.N., Albert, I., Lam, S.K., Konstan, J.A., Riedl, J.: MovieLens unplugged: experiences with an occasionally connected recommender system. In: Proceedings of the 8th International Conference on Intelligent User Interfaces. (2003) 263–266
6. van Setten, M., Pokraev, S., Koolwaaij, J. In: Context-aware Recommendations in the Mobile Tourist Application COMPASS. Volume 3137 of Lecture Notes in Computer Science. Springer Berlin (2004) 235–244
7. Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology based context modeling and reasoning using OWL. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops. (March 2004) 18–22
8. Luther, M., Fukazawa, Y., Wagner, M., Kurakake, S.: Situational reasoning for task-oriented mobile service recommendation. The Knowledge Engineering Review **23(1)** (March 2008) 7–19
9. Smith, M.K., Welty, C., McGuinness, D.L.: Web Ontology Language (OWL) guide version 1.0 (2004)
10. McGuinness, D.L., Abrahams, M.K., Resnick, L.A., Patel-Schneider, P.F., Thomason, R., Cavalli-Sforza, V., Conati, C.: CLASSIC Knowledge Representation System Tutorial. Artificial Intelligence Principles Research Department, AT&T Labs Research. (1994)
11. McBride, B.: Jena: A semantic web toolkit. IEEE Internet Computing **6(6)** (2002) 55–59
12. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Journal of Web Semantics **5(2)** (April 2007) 51–53
13. Parsia, B., Halaschek-Wiener, C., Sirin, E.: Towards incremental reasoning through updates in OWL-DL. In: Proceedings of the Reasoning on the Web Workshop. (2006)
14. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web services description language (WSDL) 1.1
15. Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H.F., Thatte, S., Winer, D.: Simple object access protocol (SOAP) 1.1
16. Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila, O., McDermott, D., Martin, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne, T., Sycara, K.: DAML-S: Web service description for the semantic web. In: Proceedings of the 1st International Semantic Web Conference. (2002)
17. Martin, D., Burstein, M., McDermott, D., McGuinness, D.L., McIlrath, S., Paolucci, M., Sirin, E., Srinivasan, N., Sycara, K.: Bringing semantics to web services with OWL-S. World Wide Web Journal **10 (3)** (September 2007) 243–277
18. Fensel, D., Bussler, C.: The web service modeling framework WSMF. Electronic Commerce Research and Applications **1(2)** (2002) 113–137
19. Adida, B., Birbeck, M., McCarron, S., Pemberton, S.: RDFa in XHTML: Syntax and processing (October 2008)
20. Battle, S., Bernstein, A., Boley, H., Grosof, B., Gruninger, M., Hull, R., Kifer, M., Martin, D., McIlraith, S., McGuinness, D.L., Su, J., Tabet, S.: Semantic web services language (SWSL) (2005)