Reconciling Ontological Differences by Assistant Agents

Kendall Lister
The University of Melbourne
Department of Computer Science and Software Engineering
Victoria, 3010, Australia
+61 3 8344 1418
krl@cs.mu.oz.au

Leon Sterling
The University of Melbourne
Department of Computer Science and Software Engineering
Victoria, 3010, Australia
+61 3 8344 1404
leon@cs.mu.oz.au

Kuldar Taveter
The University of Melbourne
Department of Computer Science and Software Engineering
Victoria, 3010, Australia
+61 3 8344 1403
kuldar@cs.mu.oz.au

ABSTRACT
In this paper, we propose an alternative solution for approaching the problem of semantic interoperability, based on recent projects to develop software agents and systems that attempt to reconcile ontological differences without explicit formal ontologies. An application capable of reconciling ontologies based on example data is first described. The results from one of the case studies where the proposed solution has been applied are reported. Finally, the design of an agent embedding the solution is sketched. The solution is aimed at the creation of assistant agents helping the users to cope with heterogeneity in both information sources and contexts of their usage.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *coherence and coordination, intelligent agents*; D.2.12 [**Software Engineering**]: Interoperability – *data mapping*

General Terms
Experimentation

Keywords
Ontology, agent, ontology reconciliation, Web service, context, information retrieval

1. INTRODUCTION
It can be envisioned that in the near future the users of the Internet will make more use of different services, which are commonly termed as Web services [1], instead of just passive browsing. Increasingly important is the delivery of customised Web services which return information based on the profile and context information, for example the location and connection mode, of the user. With the growing diversity of access terminals other than computers, such as mobile phones and portable digital assistants (PDAs), mobile information access and remote transactions are becoming commonplace. Already today, wireless carriers offer services that allow information such as weather, stock quotes, news, traffic situation and sports updates to be accessed by a mobile phone. Mobile phones are also capable of proactive invocation of useful Web services depending on the user's context. For example, business or leisure travellers by car or train may interact with different systems in order to find and utilize context-based information or services offered by various content and service providers.

The data returned by a Web service is structured by using tags (labels) of the Extensible Markup Language (XML) [2]. An information source represented by a Web service can been seen as having an implicit view of the world – *ontology*[1] – that is defined by the tags it uses to present its data and the structure and representation it gives to that data. In a similar manner, the user can be seen as having his/her own ontology which is the user's view of the world. However, the ontologies of information sources are usually different from each other and from users' ontologies. Further, tags used for data structuring may be cryptic or represented in a language that the user doesn't master. Therefore we propose to interpret the data returned by a Web service by mapping the data elements of an unfamiliar information source to the data elements of another information source the tags of which the user understands. This can be done by means of a software agent, commonly understood as an *active* entity, possessing the features of *autonomy*, *proactiveness*, *responsiveness*, and *social behaviour* [4]. A user's access terminal would then be equipped with such an agent, termed an *assistant agent*, which helps the user to make sense of the information gathered from the Internet.

2. AReXS
AReXS (Automatic Reconciliation of XML Structures) is an application built at the Intelligent Agent Lab of the University of Melbourne that reconciles heterogenous information sources by aligning them according to their implicit ontological structure. It is able to reconcile differences of expression and representation across data streams structured by means of XML tags from heterogeneous sources without any predefined knowledge or human intervention [5, 6]. It achieves this by identifying XML elements whose meanings are similar enough to be considered equivalent. AReXS requires no knowledge or experience of the domain in which it works, and indeed is completely domain independent. It uses Example-Based Frame Matching (EBFM) [7] and is able to achieve very high recall with modest precision on real world data collected from commercial Web services and

[1] A *problem-oriented ontology* is a description by truth values of the concepts and relationships of the problem domain [3].

sites. By requiring no domain knowledge, AReXS is suitable for application to any field; its success relies on its ability to identify and resolve the differences in representation that result from sourcing data from a multi-cultural environment.

AReXS does not fit into the traditional mould of an ontology management tool, as it does not use any type of formal ontology representation. Instead, it works in much the same way that two people who do not share a common language might teach each other by pointing at objects and saying the names that each person's language gives to that object. In formal terms, AReXS attempts to automatically resolve the problems of synonymy and polysemy that are significant hurdles to semantic interoperability [6, 8].

As a result of experiments conducted at the Intelligent Agent Lab, we modified the original EBFM algorithm described in [7] in two important ways. Whereas, in its original form, the algorithm would discard hypotheses that are based on a single element pair, we deliberately retain these hypotheses. Further, we made the reward for matching contents of data elements independent of the uniqueness of the match. The original EBFM algorithm devalues matches if they are common, whereas we found that this reduced the effectiveness of the reconciliation process.

3. PRACTICAL RECONCILIATION

Suppose that a user with a mobile access terminal requests a weather forecast for a particular geographical location. A currently available real Web service returned the following data:

```
<AYPY>
  <1>Moresby</1>
  <2>Papua New Guinea</2>
  <3>-09.26</3>
  <4>147.13</4>
  <5>Moresby, Papua New Guinea</5>
  <6>Sep 08, 2005 - 07:00 PM EDT</6>
  <7>from the SE (140 degrees) at 15 MPH (13 KT)</7>
  <8>greater than 7 mile(s)</8>
  <9>mostly cloudy</9>
  <10>78 F (26 C)</10>
  <11>68 F (20 C)</11>
  <12>69%</12>
  <13>29.85 in. Hg (1011 hPa)</13>
  <14>AYPY 082300Z 14013KT 9999 FEW018 BKN040 BKN140
26/20 Q1011</14>
</AYPY>
```

As one can see, the data returned makes use of cryptic tags and is therefore hard to comprehend at a glance, nor should the user have to consult the documentation page of the service. Here a personal software agent can come to the user's help. We propose that an agent faced with a situation described would retrieve data or use cached data from other information source(s) of the same purpose that the user is familiar with in order to interpret the data such as presented in the example above. For instance, another similar Web service returns the following record reporting the weather conditions of the same geographical location:

```
<AYPY>
  <1 Location>Moresby, Papua New Guinea</1 Location>
  <2 Humidity>69%</2 Humidity>
  <3 Pressure>29.85 in. Hg (1011 hPa)</3 Pressure>
  <4 Sky>mostly cloudy</4 Sky>
  <5 Temperature>78 F (26 C)</5 Temperature>
  <6 Visibility>greater than 7 mile(s)</6 Visibility>
  <7 Wind>from the SE (140 degrees) at 15 MPH (13 KT)
</7 Wind>
</AYPY>
```

In order to demonstrate how "understanding by reconciliation" could be achieved, we tested our proposed approach to ontology reconciliation on two sets of Web services. One of the sets consisted of two public Web services returning weather forecasts. The first of these services was the "World Weather" by InnerGears [9]. It retrieves the weather report for all airports in the world registered by the International Civil Aviation Organization (ICAO) based on a specified ICAO identifier. The second service "Airport Weather" by CapeScience [10] is similar to the first one but returns elements with different tag names. We devised a series of experiments where both services were invoked with the same ICAO identifiers which had been selected so that the weather forecasts returned were on the airports from all over the world. After applying AReXS to the data streams returned by the Web services, the application reported the candidate element pairs and its confidence in each match:

```
Slot 0 [1] <-> Slot 0 [1 Location]: 0.95012
Slot 1 [2] <-> Slot 0 [1 Location]: 0.95575
Slot 4 [5] <-> Slot 0 [1 Location]: 0.97643
Slot 6 [7] <-> Slot 6 [7 Wind]: 0.99999
Slot 9 [10] <-> Slot 4 [5 Temperature]: 0.97845
Slot 10 [11] <-> Slot 4 [5 Temperature]: 0.95376
Slot 11 [12] <-> Slot 1 [2 Humidity]: 0.99923
Slot 12 [13] <-> Slot 2 [3 Pressure]: 0.99968
```

The results of automatically reconciling the data streams returned by two Web services are that the elements labelled <1>, <2> and <5> of the first stream have unsurprisingly been matched with the element <1 Location> of the second stream. Analogously, the elements of the first data stream labelled <7>, <12> and <13> have been matched with the elements <7 Wind>, <2 Humidity> and <3 Pressure> of the second stream, respectively, with the highest possible degree of confidence. The degree of matching confidence between the elements <10> and <11> of the first data stream with the respective element <5 Temperature> of the second stream is somewhat lower because the first service returns both minimum and maximum temperatures, while the second service returns only the latter. Note that the elements of the two data streams could not have been matched by only considering the element names, as would generally occur in an explicit ontology alignment algorithm, such as, for example, Chimaera [11]. Also, it is worth noting that AReXS is able to cope with blank data tags.

One could argue that the matching reported between the elements of the two data streams is self-evident. It is really not impossible for the user to achieve with some effort the same results, but the user with a mobile access terminal usually lacks the time and an appropriate interface required for it.

4. DESIGNING AN ASSISTANT AGENT

In the Internet, it is inherently difficult to predict the information needs of individual agents, let alone which particular data representations they require. Although many proposed multi-agent systems enforce system-wide global ontologies, this is not always desirable or feasible, especially when the agents are embedded in mobile access terminals.

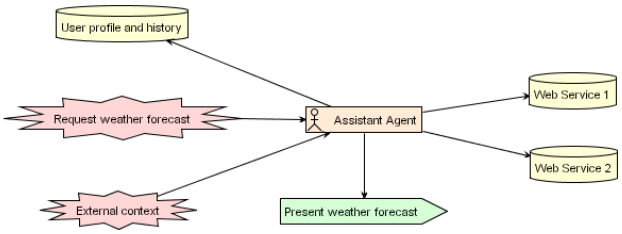

Figure 1. A system overview diagram.

We will now provide a sketch of the design of a system that has in its centre an assistant agent embedded in a mobile phone or PDA. For this purpose, we make use of the ROADMAP/Prometheus methodology [12, 13] of agent-based software engineering which is especially suitable for developing open multi-agent systems. Figure 1 represents an overview diagram of the system to be formed around the Assistant Agent. As Figure 1 reflects, the Assistant Agent perceives a request for a weather forecast by its user. In addition to that, the external context of the request, containing, for example, the current location of the user, serves as another input *percept* for the Assistant Agent. The agent also utilizes some previously collected data consisting of the user profile and the history of the previous actions by the user. Based on this information, the Assistant Agent determines a Web service that it queries for a weather forecast. However, the information returned by the Web service that the user is most familiar with may be incomplete or this Web service may not respond. In such a case, some or all parts of the weather forecast are retrieved from a different Web service. In order to decide in which form the information retrieved should be presented to the user, the Assistant Agent utilizes the data about the user profile and history. For example, provided that the Assistant Agent has cached some data from previous invocations of the user's favourite weather forecast Web service, it is able to "translate" the names of data elements used in the information retrieved from "alien" information sources to the data element names known by the user. As a result, the weather forecast is finally presented to the user in the form which is as close as possible to the one preferred by her/him. In Figure 1 this is reflected by the agent's *action* "Present weather forecast". Please note that the whole process described occurs in a dynamic manner without the use of *any* pre-specified domain ontology.

5. CONCLUSIONS

Even though the Internet is in principle global, a lot of information available in it is of only local significance. Context-dependent invocation of Web services and context-dependent retrieval of Web pages are therefore of an increasing importance. This implies that a growing number of different views of the world will be present in the Internet. However, it seems unlikely that any single proposed solution for adding semantics to the Internet, such as Ontology Web Language (OWL) [14], will be widely accepted in the near future. Even if such an event occurred, it is doubtful that many smaller commercial and individual publishers of information would be willing to devote the time and effort required to comply with a standard that requires ontology development. For this reason, we are proposing and developing technologies and methodologies that cope with heterogeneity and change in information sources and in contexts of their usage by performing implicit ontological reconciliation.

6. ACKNOWLEDGEMENTS

The authors would like to thank the Centre for Perceptive and Intelligent Machines in Complex Environments of the Australian Research Council for supporting the work reported.

7. REFERENCES

[1] *Web Services Activity*. The World Wide Web Consortium, http://www.w3.org/2002/ws/, retrieved on 21 Sept. 2005.

[2] *Extensible Markup Language* (XML). The World Wide Web Consortium, http://www.w3.org/XML/, retrieved on 21 Sept. 2005.

[3] Gruber, T. R. A Translation Approach to Portable Ontologies. *Knowledge Acquisition, 5*(2), 199-220, 1993.

[4] Jennings, N. R., Sycara, K., Wooldridge, M. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems, 1*(1), 7-38, 1998.

[5] Hou, D. *Automatic Reconciliation of XML Structures*. Honours thesis, Department of Computer Science and Software Engineering, The University of Melbourne, Australia, 2001.

[6] Lister, K., Sterling, L. Agents in a Multi-Cultural World: Towards Ontological Reconciliation. In *AI 2001: Advances in Artificial Intelligence, 14th Australian Joint Conference on Artificial Intelligence, Adelaide, Australia, December 10-14, 2001, Proceedings*. LNCS 2256, Springer 2001, 321-332.

[7] Ikeda, Y., Itoh, F., Ueda, T. Example-Based Frame Mapping for Heterogeneous Information Agents. In *Proceedings of the 3rd International Conference on Multi-Agent Systems* (ICMAS 1998), 3-7 July 1998, Paris, France. IEEE 1998, 150-157.

[8] Klein, M. Supporting Evolving Ontologies on the Internet. In *EDBT 2002 Ph.D. Workshop, Prague, Czech Republic, March 28, Proceedings*. Matfyzpress 2002, 51-58.

[9] *GetWorldWeatherByICAO Documentation*. InnerGears, http://www.innergears.com/WebServices/WorldWeatherByICAO/DOC/WorldWeatherByICAO.htm, retrieved on 22 Sept. 2005.

[10] *Airport Weather Web Service*. CapeScience, http://capescience.capeclear.com/webservices/airportweather/index.shtml, retrieved on 22 Sept. 2005.

[11] McGuinness, D., Fikes, R., Rice, J., Wilder, S. The Chimaera Ontology Environment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, July 30 - August 3, 2000, Austin, Texas, USA. AAAI Press / The MIT Press 2000, 1123-1124.

[12] Kuan, P. P., Karunasakera, S., Sterling, L. Improving Goal and Role Oriented Analysis for Agent Based Systems. In *Proceedings of the 16th Australian Software Engineering Conference* (ASWEC 2005), 31 March - 1 April 2005, Brisbane, Australia. IEEE 2005, 40-47.

[13] Padgham, L., Winikoff, M. Developing Intelligent Agent Systems: A Practical Guide. John Wiley & Sons 2004.

[14] *OWL Web Ontology Language Reference*. W3C Recommendation 10 February 2004, http://www.w3.org/TR/owl-ref/, retrieved on 1 Feb. 2006.