Semantic Web Interaction on Internet Relay Chat

Jennifer Golbeck
University of Maryland, College Park
Department of Computer Science
A.V. Williams Building
College Park, MD 20740
Tel: 1-301-498-7476
E-mail: golbeck@cs.umd.edu

Paul Mutton
Computing Laboratory
University of Kent
Canterbury, Kent, CT2 7NF
United Kingdom
Tel: +44 (0)1227 764000 ext. 7754
E-mail: pjm2@kent.ac.uk

ABSTRACT
Internet Relay Chat (IRC) is a chat system that has millions of users. IRC robots (bots) are programs that sit in chat rooms and provide different services to users. The IRC bot as a mechanism for human interaction with the Semantic Web – specifically with web services and knowledge bases – is simple to program, has an intuitive, conversational interface for human users, and fits well with the inputs and outputs of Semantic Web queries. This paper presents implementations of bots for interacting with web services and FOAF/Trust data models. We outline tools for implementation and present directions for future work to generalize the function of these bots.

KEYWORDS: Internet Relay Chat (IRC), Semantic Web, FOAF, bots

INTRODUCTION
The World Wide Web grew out of a design with the goal of presenting information to users. HTML evolved features to allow users to layout and present content to human readers in a variety of ways. The Semantic Web extends the capabilities of the web to make content understandable by computers. This has gone far beyond the markup of existing web content so that a machine-readable version exists parallel to the HTML version. The Semantic Web comprises large, distributed collections of data that are not accessible to human web users. Semantic Web services, designed to be easily composed software-powered resources for use in applications, provide simple interfaces to functionality that can be very useful for human users. Unfortunately, without writing code or going through a web page interface, users do not have easy access to web service functionality.

Human interaction with web services and Semantic Web data models must take place in an interactive, online medium. Because most Semantic Web data and functionality is not accessible through web pages, in this paper we look at alternatives to interaction through websites. We will focus on *Internet Relay Chat* (IRC), which is a chat system accessible anywhere on the internet, allowing users to join in with live discussions. IRC is used by millions of people from all around the world and can even be used to facilitate collaboration with researchers in other countries. While IRC is commonly accessed by humans using IRC client software, some client programs are totally autonomous. Such programs run without direct human input, but are designed to react to input from other human users on IRC. To distinguish them from their human counterparts, these autonomous clients are dubbed *IRC robots*; a term which is commonly contracted to *IRC bots*.

IRC AND IRC BOTS

Internet Relay Chat (IRC) is a chat system that lets a user connect to a network of IRC servers and participate in live discussions. Once connected to an IRC network, a user can join a set of *channels*. Each channel is rather analogous to a room, containing a set of users interested in the same topic. Each user can send a message to a channel and expect it to be received by everyone else in that channel, much like somebody talking in a room, who can be overheard by others in the same room. Unlike some centralized systems, IRC does not require a sign up or registration, so any user with an IRC client can connect under any name at any time. There are hundreds of IRC networks, and some of the larger ones see over one million users each week.

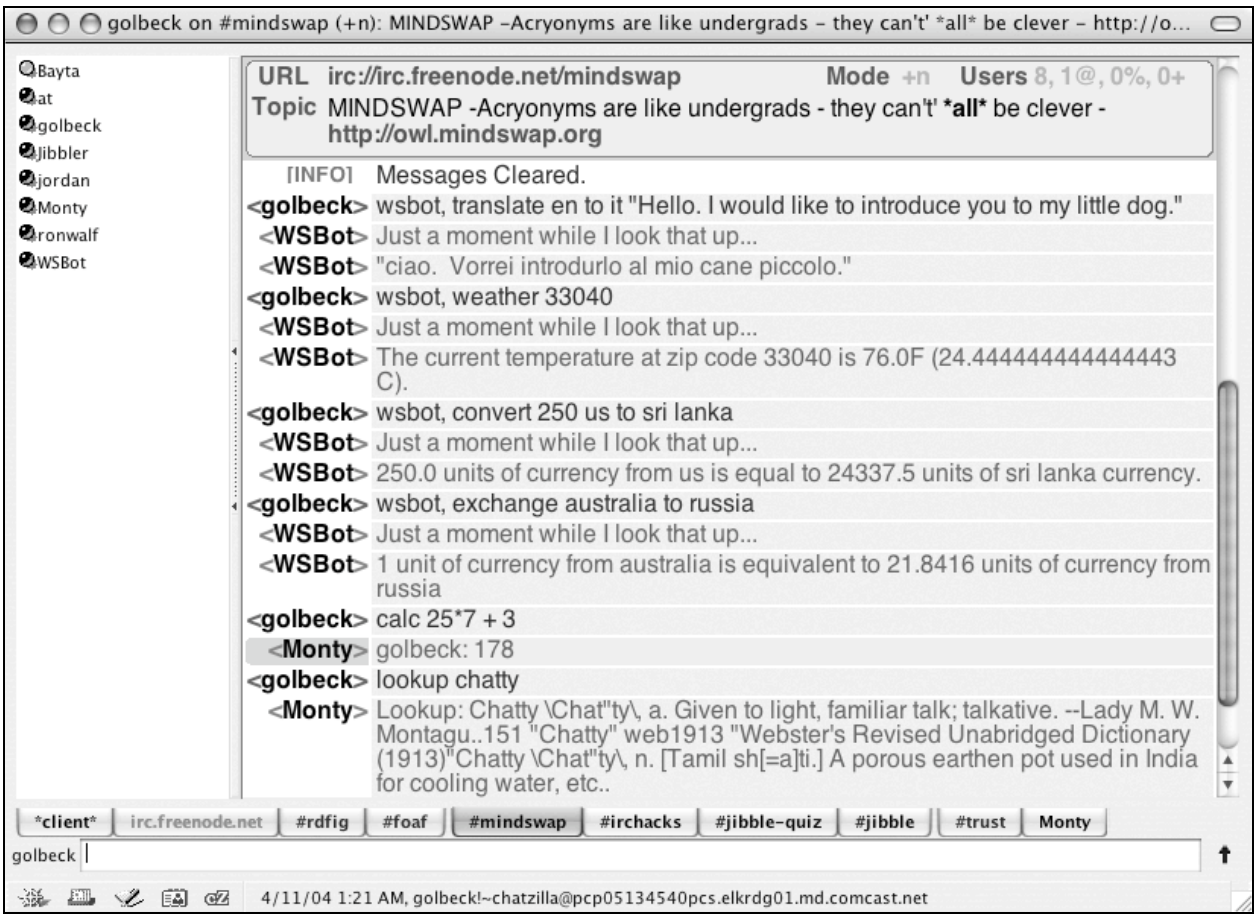

Figure 1: an example of interaction between a user (golbeck) and two IRC bots (WSBot and Monty) in the #mindswap channel on the irc.freenode.net IRC network.

As far as casual observers are concerned, there is no distinction between a human-driven IRC client and an IRC bot. An IRC bot looks much like any other user in a channel, and can usually only be distinguished if it is obvious that it exhibits autonomous behavior. For this reason, it is polite (and indeed common practice) to make other users aware of the bot's presence and to let them know what it does, particularly if it is not obvious from its behavior. Bots can be used to record public logs of conversations on channels; others are merely fun AI programs that attempt to hold conversations, and many provide simple services like calculators, date and time lookups, alarms, and web searches

.

SEMANTIC WEB INTERACTION WITH BOTS

Interaction with IRC bots is usually conversational or done with simple commands, as shown in figure 1. This simple type of interaction is ideal for interacting with web services or making queries to semantic data models. The fact that queries and web services will generally return simple responses that can be reported in one line of text also means that the IRC medium is well suited to serving as an interface between human user and Semantic Web.

Web Service Bots

Web services neatly package many functions that are typical of the functionality users expect from IRC bots. Weather forecasts, stock quotes and translation services have certainly been incorporated into bots without the use of web services. The source of that information, however, usually comes from parsing web pages. This has the disadvantage that if the format of the web site changes, even slightly, the data extraction code no longer works. Web services do not suffer from this problem. Programmers only need to know the URI of a web services WSDL description. From there, they can extract the name of a web service's operations, the names of the input message parts, and the name of the output message. These pieces of information are all that is required to enable a bot to use a web service. For example, when creating a Java based bot using the PircBot IRC API

[4] and Apache Axis for accessing web services, the actual process of programming a bot to read a user command in an IRC channel, invoke the corresponding web service, and send the result message over IRC can be done in under ten lines of code. This ease of programming means that it is trivial to bring Semantic Web services to users through bots. The friendly interface of bots also means that using the services is intuitive for IRC users. WSBot, shown in Figure 1, uses Semantic Web services in this way to provide a translator, temperature lookup, and currency converter. The conversational tone is maintained in the bot's responses, by wrapping the service output in a sentence.

Bots with Backend Knowledge Bases

A more complex issue on the Semantic Web is aggregating data into knowledge bases or unified data models. The nature of the Semantic Web as a distributed open system means that files containing interrelated data can be spread across any number of servers. To look at an entire data model requires aggregating all of the data contained in distributed files into a single model. To interact with the model, a user needs an interface to pose queries and view results. IRC bots have been used for this purpose in several contexts. The most popular bots work in the context of the Friend-Of-A-Friend (FOAF) project [2], and the Trust Project[7].

Figure 2: interactions between user (golbeck) and bots (TrustBot and foafbot). Both bots maintain a model built from Semantic Web data and use the IRC interface for receiving queries to those models.

FOAF is a vocabulary for describing people, their personal information, and their interconnections. The Trust Project extends the FOAF vocabulary, allowing people to express their trust relationships with other people. Both vocabularies are expressed as simple OWL ontologies[2,7]. Any person can create a file with FOAF or Trust information and make it available online. There are thousands of users in the Trust project and millions of FOAF files on the Semantic Web. If it is merged together, all of the data forms a large, connected social network with useful information about the people in it. Because the data is spread out over so many files, a web spider must search for files, parse them, and aggregate the data into a single model in order for users to make queries. The type of information one may want from the

centralized network includes simple personal data like name, email address, photos, and web pages, or more complex data like the number and length of paths connecting people in the network, or recommendations about how much two strangers should trust each other based on the network structure. Fortunately, posing these types of questions are simple for users, and the results are short. That, combined with the relevance of the information to common IRC interactions, makes IRC bots serving the data an ideal interface for interacting with the semantic model.

Two bots currently exist and serve up this information. Foafbot and Trustbot each maintain a large model on its back end and process requests for information over IRC. As shown in figure 2, they have a natural dialogue similar to the bots shown in figure 1. Both bots support a set of queries to access the most common features of their data model and functions over it.

Tools for Implementing Semantic Web Bots

There are many tools available to help in building Semantic Web bots. Although there are toolkits to support Semantic Web interactions in many languages – particularly C and python – Java is by far the most common language used for semantic web development. The bots in this paper are all Java based, and take advantage of the excellent open source toolkits that are available. The popularity of Java on the Semantic Web is certainly due, in large part, to HP's open source Jena toolkit[1]. Among its many capabilities, Jena has an API for reading and creating RDF and OWL files. For programmers, this means it is straightforward to write applications that build knowledge bases from the Semantic Web. The Apache Axis framework for web services is also open source and allows a programmer to invoke web services with only a few lines of code. For the front end, PircBot [4] is an open-source, Java based framework for creating IRC bots. These three tools free the programmer from needing to understand the intricacies of IRC and the Semantic Web. With a general understanding of how to combine elements of each, any Java programmer can assemble these elements with a focus on the user's interaction experience.

CONCLUSIONS AND FUTURE WORK

In this paper we describe how Internet Relay Chat bots can provide a simple and intuitive interface for human interaction with Semantic Web data. We have shown bots that use IRC as an interface for invoking web services, and mechanisms for putting a bot's simple interface on top of queries to a back end knowledge base.

A future step beyond the current web service bot implementation would be to allow users to dynamically add web service capabilities to the bot. Only a small amount of information – the WSDL location of a service and the parameters necessary for an operation – is required to expand a bot. Because the information needed is simple and invoking a web service is an easily parameterizable function, users could program a bot with a new command corresponding to a web service over the IRC interface relatively easily. This would allow the bot to know about arbitrary web services and expand as new services become available.

Just like generalizing bots for web services, bots with access to more general knowledge bases than the Foafbot and Trustbots can also exist. The swBot [6] is backed by an RDF knowledge base. It can contain arbitrary facts using any ontology, and process queries created by the user. However, because these queries are not pre-specified, they use a more complex query. One example offered at [6] is as follows:

Goal: Point me to pictures that show both Dan and Gerald.
Form of Query: Tell me about things that are depictions of both something whose name starts with "Dan C" and something whose name starts with "Gerald".
Query that must be entered over IRC: swBot, { ?d foaf:name [log:startsWith "Dan C"]; foaf:depiction ?pic. ?g foaf:name [log:startsWith "Gerald"]; foaf:depiction ?pic. } => { ?pic a :PhotoOfDanAndGerald } ?

Clearly, this type of query interface is powerful, but also not accessible by most users. A space of future work in this area is to develop a structured but natural method of posing queries to bots with backend knowledge bases such that they will be useful over IRC.

REFERENCES

1. Carroll, Jeremy J., Dickinson, Ian, Dollin, Chris; Reynolds, Dave, Seaborne, Andy, Wilkinson, Kevin, "Jena: Implementing the Semantic Web Recommendations", *HP Technical Report HPL-2003-146*, 24 Dec, 2003.
2. The Friend-Of-A-Friend (FOAF) Ontology: http://xmlns.com/foaf/0.1/

3. Golbeck, Jennifer, Bijan Parsia, James Hendler, “Trust Networks on the Semantic Web,” *Proceedings of Cooperative Information Agents 2003*, August 27-29, Helsinki, Finland.
4. Mutton, Paul. *PircBot 1.2.5 Java IRC API: Have fun with Java*. In *Java Developer’s Journal*, Volume 8, Issue 12. (December 2003) pp. 26-32
5. Mutton, Paul. *IRC Hacks* (O’Reilly). To be published (August 2004).
6. The swBot: http://dev.w3.org/cvsweb/2000/scribe-bot/ [accessed April 10, 2004].
7. The Trust Project Ontology: http://trust.mindswap.org/ont/trust.owl