Technology for automated assessment: The World-Wide-Mind

Ciarán O'Leary

Dublin Institute of Technology, School of Computing, Kevin St., Dublin 8, Ireland
Ciaran.OLeary@comp.dit.ie, www.comp.dit.ie/coleary

Abstract. Technology recently developed for a large scale computing project offers what we feel is the first attempt to provide a platform where students can compare and contrast their skills and performance with students world-wide.

The World-Wide-Mind [w2mind.org] is an artificial intelligence (AI) project that aims to make various tools, algorithms, artificial worlds and artificial minds available online as web services that can be interacted with remotely. The core idea of the project is that anyone who has an interesting assignment can make this available to others who can run it remotely. Similarly anyone who has written an algorithm to solve the problem posed by the assignment can make it available online. A freely available program can then be used by third parties to run the algorithm (or mind) in the artificial world and observe its performance.

A score board maintained by the artificial world service can record the best performing minds. This architecture was recently used by a group of final year computing students at the Dublin Institute of Technology, to solve a relatively simple AI problem. The performance of their algorithms was used in their grading, but unlike traditional assignments, their software is still available to interact with online, as indeed is the problem world they used.

One of the goals of the World-Wide-Mind project is to get researchers/lecturers/teachers to put problem worlds online as well as the artificial minds that are used to solve the problems. Other institutions can then easily reuse the world in their own assignments, and compare the results of their students with the results of others where relative performance could easily be used for automated marking. Students can also control runs of existing minds with existing worlds.

The technology behind the project is deliberately simple and does not require the purchase of any tools.

Keywords: Remote re-use, Performance based evaluation, World-Wide-Mind

1 Introduction

The World-Wide-Mind (Humphrys, 2001, Humphrys and O'Leary, 2002) is a distributed artificial intelligence project that aims to make it easier for researchers to build large, hybrid, massively diverse artificial minds. The potential success of the project is dependent on the large scale co-operation of communities of researchers. In order to facilitate this co-operation a protocol and architecture was developed which aims to make it as simple as possible to create a software component that can be re-used by another researcher in the development of their software.

The simplicity of the technology required made this architecture ideal for usage in a learning environment, where students could build software components that could be made available online. Similarly, any components used in the instruction or assessment of a course could also be made available online for others to use. When used for assessment, it is possible to foresee a situation where an instructor on one course could remotely reuse an online assessment given by another instructor.

This paper will describe the development of one such component for assessment of coursework for a final year class of Computer Science undergraduates at the Dublin Institute of Technology. The component is still available online to be interacted with by others, and could be used for assessment by another instructor teaching a related subject. Interaction with the component takes place using the simple network protocol of the World-Wide-Mind.

It is felt that the usage of the protocol/architecture of the World-Wide-Mind offers a number of opportunities and benefits for instructors, in particular

1. No requirement for the purchase of tools or software packages.
2. No requirement for local installation of software.
3. Possibility for large repository of interactive materials available online.

This paper will proceed as follows. Section 2 will discuss the World-Wide-Mind in as much detail as is required to understand the motivation behind the project as well as the overall goals of the project. This section will demonstrate the architecture that has been developed, as well as the protocol that is in use on the World-Wide-Mind. Section 3 will describe how the architecture has been used for the assessment of a class of undergraduates, and how the software developed here could be reused remotely by others. Section 4 will analyse the protocol and architecture and place it in context with some related projects such as PEARL and the Grid. Section 5 will summarise, and discuss possible future development of the World-Wide-Mind in an e-Learning context.

2 The World-Wide-Mind

"Since, according to faculty psychologists, the mental causation of behaviour typically involves the simultaneous activity of a variety of distinct psychological

mechanisms*, the best research strategy would seem to be* divide and conquer*: first study the intrinsic characteristics of each of the presumed faculties, then study the ways in which they interact"*(Fodor, 1983)

The engineering of artificial minds is made difficult by virtue of the fact that most parts of these minds are very different from all other parts. For example, in a complete mind, one would expect to find modules responsible for vision, language, navigation etc. Any one research group that attempts to build intelligent agents, robots or artificial virtual creatures is limited by the skills, expertise and resources available in its laboratory. Also, researchers have long pointed to the fact that we cannot know all the modules that are required for an artificial mind, if this mind reaches any level of complexity (Brooks, 1991).

A possible step in the direction of providing all the required functionality would be to distribute all the modules of the mind throughout the Internet, and provide some mechanism whereby components of artificial minds could be integrated with each-other by third party researchers. For example, should *research group A* have an interest in vision they could make their software available online to be interacted with remotely. *Research group B* could do likewise with their say, language modules. *Research group C* could then create a mind that would use the modules made available by research groups A and B.

2.1 Why has this not happened already?

It would seem that the idea of using other's research is not a new one, since research by its very nature requires that we build on the work of others. However, it is rare that we, as computer scientists, see the software that is used in the experiments of others, particularly in the area of artificial intelligence. The published works of researchers will often include detailed descriptions of algorithms used and the results of experiments conducted using these algorithms, but if these are to be reused they must be rewritten. Even then, it is difficult and time consuming to achieve the same results without becoming an expert on the topic at hand. Should the researcher have made their software available for download and local installation, it is then often the case that platform, version, library or programming language differences introduce a whole set of new difficulties. The possible solution that is proposed by the World-Wide-Mind project is the solution of the World-Wide-Web, using simple protocols and standards, allow the software to be used remotely. An early goal of the World-Wide-Mind was to define as simple as possible an architecture and protocol to allow, and even encourage researchers to make their software available as what we term a *Lightweight Web Service* for the World-Wide-Mind.

2.2 World-Wide-Mind Architecture

The architecture of the World-Wide-Mind is server based. Researchers create servers to represent either a problem world, or an artificial mind. The problem world could represent any problem or sub-problem. The mind represents an algorithm

written to solve the problem. Both the mind and the world are put online as web services, or more accurately, as lightweight web services, of which more later. Client software can then be used to let the mind and world interact with each other, as shown in figure 1 below.

The manner in which the mind and world interact through the client is as follows:

1. The client is given the URL of the world service.
2. The client is given the URL of the mind service.
3. The client sends a (`newrun`) message to world service to inform it that it wishes to start a run. The client in then returned a `run id`, which uniquely identifies it to the server, and should accompany all subsequent requests.
4. The client starts a run in the mind service in the same fashion.
5. The client queries the world for its state, which is returned in any format chosen by the world, but wrapped inside a protocol message. It is impossible to standardise the format for the state of problem worlds since this will vary significantly between different problems.
6. The client presents this state to the mind service which will return an action to be executed in the world. Obviously the mind service will have to have been written specifically for the given problem world so that it can interpret the state correctly.
7. The client then presents the action to the world service, where it is executed and the new state is returned. We then loop back to point six above until the client decides to end the run. (It is also possible for the world service to inform the client that the run has ended, if the problem has been solved.)

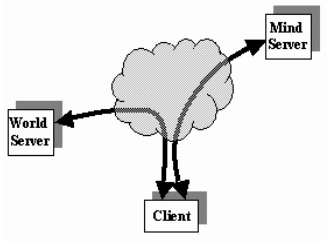

Fig. 1. Client, world server and mind server architecture of the World-Wide-Mind

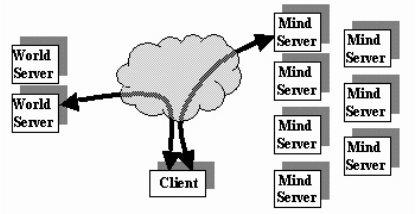

Fig. 2. Client can select from an array of different mind and world servers.

Figure 2 above shows the Internet populated with numerous mind and world services. The client can select any world service and attempt to run any mind service in this world. Obviously the mind will not solve the problem if it does not understand the state of the world. However, it is not the responsibility of the World-Wide-Mind project to ensure that such problems do not arise (no more so than the World-Wide-Web Consortium [w3.org] don't prevent you viewing web sites with unfamiliar characters). What is needed, however, is a simple protocol that can be used for the client to talk to the services. This protocol is SOML and is central to what we define as *Lightweight Web Services*.

2.3 Lightweight Web Services

"The term Web services describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available"

-www.webopaedia.com

Web services are interfaces to objects that can be accessed over the ubiquitous HTTP protocol. They were originally seen as a potential replacement for, but are not considered as complementary to other distributed object system protocols such as J2EE [http://java.sun.com/j2ee/] and CORBA [www.corba.org].

By using Web Services developers can make their object/component available to the world by placing it on a web server, and building a special XML interface to it. Web services provide XML based standards for description of a service, discovery of a service and invocation of a service

Web services would appear to be the ideal way to allow remote invocation of our mind components. However following analysis it was decided that the Web Service protocols provide far too much functionality, and their complexity would pose a barrier to entry to those who would have little interest in learning the family of protocols.

Lightweight web services provide only that functionality that is required for our purposes. We only need a simple messaging system to invoke methods on the services, and a simple way to describe the service, in terms of the parameters it expects etc. This is required so that the client will know what information to ask the user for before starting a run in the world or mind.

The SOML (Society of Mind Markup Language) protocol defines the messages and the description mechanism in a single specification. It is an XML-like protocol without being actually XML based. For our requirements we saw XML as overkill. We only require that authors of services make their services available for invocation by supporting simple messages such as the one shown below in figure 3. We do not enforce well-formedness or strong validity. This point is expanded upon elsewhere (O'Leary and Humphrys, 2002) and the SOML protocol is described in full in (O'Leary, 2002).

```
1. <soml version="0.9">
2.     <request type="getaction" runid="123456">
3.         <param name="state">
4.             (0, 0, 0, 9, 7, 8, 2, 0, 4)
5.         </param>
6.         <argument name="usemem" value="true"/>
7.     </request>
8. </soml>
```

Fig. 3. An SOML message

2.4 Building a service

In order to build a service, whether a world or a mind service, all that is required is that the author have access to a web server, and have some limited knowledge of some web technology such as Java Servlets [java.sun.com/products/servlet/], Java Server Pages http://java.sun.com/products/jsp/] or Active Server Pages [www.activeserverpages.com]. Even without this knowledge it is possible to use the Common Gateway Interface of the web server to read and write from the World-Wide-Web using any programming language.

Detailed instructions on how to create a service will be made available at the World-Wide-Mind portal site [w2mind.org] in the near future.

2.5 Client

The client software is required, as we have seen, in order to start a run in the world and mind services, and control them until completion. So far, two clients have been built. One is a HTML form that can be used for testing, as shown in Figure 4 below. The second is a Java applet. Figure 5a shows the front page of the applet, where the user enters the URLs of the two services. Figure 5b shows the applet during a run. The applet has a great deal more functionality (e.g. entering start-up parameters) as it supports the full SOML protocol.

Both the applet and the web page were designed to be as simple as possible, and require that the user only need to know the URLs of the two services to view a run in progress.

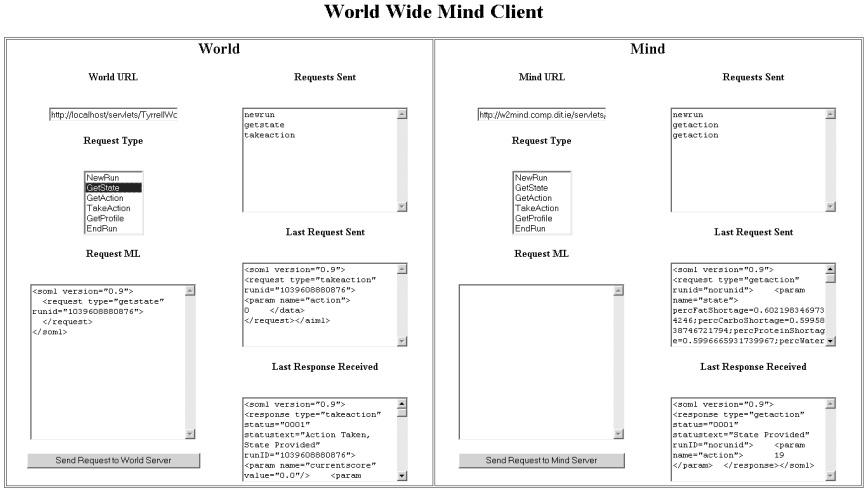

Fig. 4. HTML form used as a client for the World-Wide-Mind.

Fig. 5a. World-Wide-Mind Java applet client front page

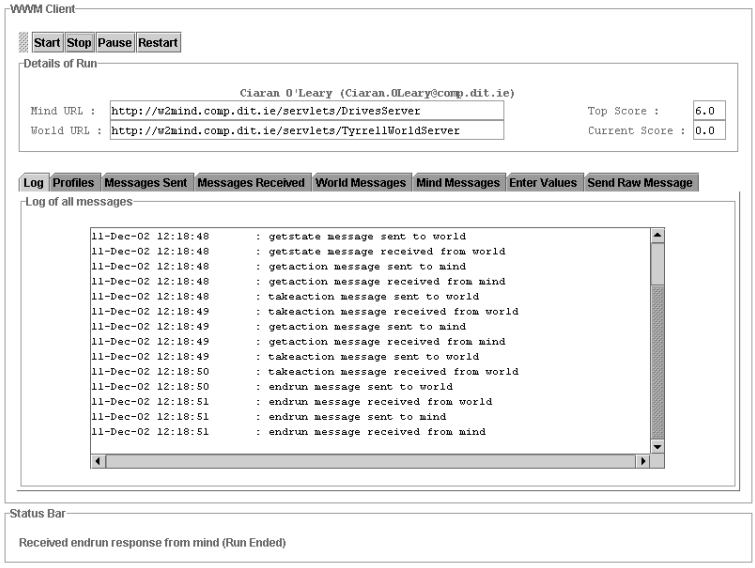

Fig. 5b. World-Wide-Mind client during a run

3. World-Wide-Mind in a learning environment

World-Wide-Mind technology was originally developed for the artificial intelligence project described above. The characteristics of the architecture and protocol, in particular their simplicity, lend themselves nicely to use in a learning environment. In such an environment, students, in particular Computer Science students should have sufficient knowledge to be able to master the simple requirements for putting a service online. Most students already have access to web servers and maintain their own web sites. The majority of third level courses teach modules on web development where students are exposed to a number of development technologies.

3.1 Blocks World example

Recently a class of final year Computer Science students at the Dublin Institute of Technology were given an assignment where they were required to solve the well known *blocks world* problem (Slaney and Thiébaux, 2001).

The problem is as follows. The student is presented with a number of columns, each of which contain a certain number of blocks. Each block is identified by a letter of the alphabet. The student has to write control software for a virtual robot arm, where the robot arm can *grip* a named block and then *ungrip*, or drop the block into any column. The problem is solved when all the blocks are stacked in ascending alphabetical order. The score that is achieved in the number of moves that were needed to be made to solve the problem, where lower scores are obviously superior to higher scores.

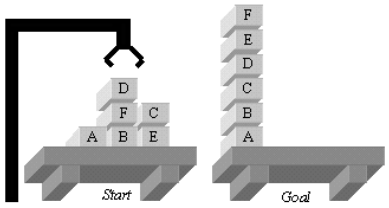

Fig. 6. Sample blocks world problem, with start and goal states.

This problem was presented to the class of undergraduates as a World-Wide-Mind service. Each student was required to write a mind service to solve the problem, and then run the mind in the world. The score was recorded by a scoreboard at the world service.

The world service presented the state of the world in the format shown in figure 7a below. The mind service had to choose an action from the list shown in figure 7b, where a `grip` action gripped a named block only if it was clear. An `ungrip` action dropped the currently gripped block into the column provided as a parameter.

```
empty(1)&ontable(A,2)&clear(A)&
ontable(B,3)&on(F,B)&on(D,F)&
clear(D)&ontable(E,4)&on(C,E)&
clear(C)&empty(5)&gripper()
```

Fig. 7a. Initial state of the blocks world problem shown above

- `grip(A)`
- `ungrip(3)`

Fig. 7b. Sample actions that could be chosen by the mind.

3.2 Results

As a result of this assignment, the following are now available online:

1. The blocks world problem as a Lightweight Web Service.
2. Ten solutions for the blocks world problem available as Lightweight Web Services.
3. A scoreboard listing the best performing minds in the problem.

Obviously, the client software described earlier is also available online for others to use in starting a run in the world and a selected mind, and observing its performance.

4. Analysis

It would seem clear that this model provides a simple, efficient method for reusability, and a possibility for automated correction based on performance. The fact that no tools are required in order to create a service is an advantage, and the freely available protocol means that anyone can make a service available. Also, the fact that there is no requirement for local installation of any software means that there are no issues with versions, platforms or programming languages.

Should a number of different instructors make their assignments available online as services, and accompany these services with scoreboards, then these assignments could be reused for other assignments, and give an indication of the performance of one set of students against another. Also, if the mind services are kept online, students could integrate the better solutions with their own (once credit is given) in order to evolve superior solutions for well known problems.

This model of remote re-use is similar in a lot of regards to the PEARL project (Cooper *et. al.*, 2002) in e-Learning and the Grid [www.gridforum.org] project where instruments are given a web interface for others to interact with. The difference here, we feel is that the entry level is sufficiently low for students to create their own services. Also, clearly, there is no risk of any damage being caused since students will be interacting with virtual worlds and problems, unlike the Grid and PEARL where the interaction is with physical devices.

The future may see the integration of this protocol with other standards in e-Learning. While this would be welcomed, we would be eager to maintain the low entry level. It is better to get a lot of simple services we feel, than to get a small number of highly complex services.

Further analysis of the issues surrounding the correction of assignments based on the scores achieved will be required, as well as the issues regarding the re-use of one student's work in other student's work through online integration.

5 Summary and Conclusion

The World-Wide-Mind project has been responsible for the development of an architecture and protocol that is intended to make it as easy as possible for researchers to build software components that are made available online for remote re-use. A by product of this project has been the potential for the use of the architecture and protocol in a learning environment. We presented above a discussion of the first successful attempt to make use of the World-Wide-Mind in a learning environment where ten final year undergraduate students made their assignments available online as lightweight web services for others to interact with. The assignment problem is also available along with score-board for others to use, where they can be encouraged by the possibility for automated assignment correction using the scores achieved in the world.

This paper has outlined the potential for the use of the technology discussed in a learning environment. Development will continue on both the technology and on higher levels for the protocol.

References

Brooks, 1991 — Brooks, R.A. (1991), *Intelligence without Representation*, Artificial Intelligence 47:139-160

Cooper *et. al.*, 2002 — Cooper, Martyn, Donnelly, Alexis, Ferriera, Jose Martins (2002), *Remote controlled experiments for teaching over the Internet: A comparison of approaches developed in the PEARL Project*, ASCILITE 2002, Auckland, New Zealand

Fodor, 1983 — Fodor, J. A. (1983). The Modularity of Mind. Bradford Books. MIT Press, Cambridge, MA.

Humphrys, 2001 — Humphrys, Mark (2001), *The World-Wide-Mind: Draft Proposal*, Dublin City University, School of Computer Applications, Technical Report CA-0301, February 2001 computing.dcu.ie/~humphrys/WWM/

Humphrys and O'Leary, 2002 — Humphrys, Mark and O'Leary, Ciarán (2002), *Constructing complex minds through multiple authors*, in From Animals To Animats 7: The 7th International Conference on the Simulation of Adaptive Behavior (SAB-02), August 2002, Edinburgh, Scotland.

O'Leary and Humphrys, 2002 — O'Leary, Ciarán and Humphrys, Mark (2002), *Lowering the entry level: Lessons from the Web and the Semantic Web for the World-Wide-Mind*, poster at 1st Int. Semantic Web Conf. (ISWC-02), June 2002, Sardinia, Italy

O'Leary, 2002 — O'Leary, Ciarán (2002), *SOML(0.9) Society of Mind Markup Language 0.9*, Available online at w2mind.comp.dit.ie

Slaney and Thiébaux, 2001 — Slaney, John and Thiébaux, Sylvie (2001), *Blocks World Revisited*, Artificial Intelligence 125:119-153