Distributing a Mind on the Internet: The World-Wide-Mind

Mark Humphrys

Dublin City University, School of Computer Applications,
Glasnevin, Dublin 9, Ireland.
humphrys@compapp.dcu.ie
www.compapp.dcu.ie/~humphrys

To appear in *Proc. 6th European Conf. on Artificial Life (ECAL-01)*.

Abstract. It is proposed that researchers in AI and ALife construct their agent minds and agent worlds as *servers* on the Internet. Under this scheme, not only will 3rd parties be able to re-use agent worlds in their own projects (a long-standing aim of other schemes), but 3rd parties will be able to re-use agent minds as components in larger, multiple-mind, cognitive systems. Under this scheme, any 3rd party user on the Internet may select multiple minds from different remote "mind servers", select a remote "Action Selection server" to resolve the conflicts between them, and run the resulting "society of mind" in the *world* provided on another "world server". Re-use is done not by installing the software, but rather by using a remote service. Hence the term, the "World-Wide-Mind" (WWM), referring to the fact that the mind may be physically distributed across the world. This model addresses the possibility that the AI project may be too big for any single laboratory to complete, so it will be necessary both to decentralise the work and to allow a massive and ongoing experiment with different schemes of decentralisation. We expect that researchers will *not* agree on how to divide up the AI work, so components will overlap and be duplicated and we need multiple-conflicting-minds models [21]. We define the *set of queries and responses* that the servers should implement. Initially we consider schemes of *low-bandwidth* communication, e.g. schemes using numeric weights to resolve competition. This protocol may initially be more suitable to *sub-symbolic* AI. The first prototype implementation is described in [47]. It may be *premature* in some areas of AI to formulate a "mind network protocol", but in the sub-symbolic domain it could be attempted now.

Note: *The use of the term "AI" may cause confusion to an ALife audience. Logically, it should be the case [22] that artificial intelligence is a subfield of artificial animals which is a subfield of artificial life. However, this taxonomy has not caught on, so throughout this paper I use "AI" to refer to all artificial modelling of life, animals and humans. In the sense in which we use it, classic symbolic AI, sub-symbolic AI, Animats, Agents and ALife are all subfields of "AI". To summarise, this paper applies to all types of artificial minds, whether the types popular in ALife, or not. To illustrate this further, see implementation of various ALife models in the section:* ***"How to implement existing agent architectures as networks of WWM servers"*** *below.*

1 Introduction

The starting point for our motivation is the argument that the AI project is too big for any single laboratory to do it all. Many authors have argued along these lines, and a number of different approaches have evolved:

- The traditional AI approach has been to work on *subsections* of the postulated mind. The criticism of this approach [4, 5] is that the "whole mind" never actually gets built, and each subsection can avoid hard problems as "someone else's problem", so that *no-one* addresses them. For a symbolic AI call to build whole systems see [30].
- The *Animats* approach [50] is to start with *simple whole creatures* and work up gradually to more complex whole creatures. But as the complexity scales up, it cannot avoid the question of whether one lab can really do it all. Perhaps the Cog project [6, 7] is now beginning to hit those limits.
- The *evolutionary* approach is to say that control systems are too hard to design and must be evolved [17]. In practice this has also usually seemed to share with the animat approach an implicit assumption that one lab can do it all.

It seems to me that all these approaches still avoid the basic question: **If the AI project is too big for any single laboratory to do it all, then as we scale up, how will we link the work of multiple laboratories?** Who decides who works on which piece? What if people can't agree on how to divide up the work, or indeed what the pieces are? [5] Will this scheme force everyone to use a common programming language? Will it enforce a common AI methodology and exclude others?

This paper proposes a scheme for decentralising the work in AI *without* having to agree on any of the above issues. Briefly (for a fuller discussion see [24]):

1. Researchers will *never* agree on how to divide up the work, so we need models in which this is not a problem, i.e. *models of multiple overlapping, conflicting and duplicated minds* [28, 29], and conflict-resolution mechanisms to generate winning actions from such a collection [21].
2. Researchers do not re-use each others' work for the same reasons that software re-use in general [20] has not been as easy as promised - complex installation, and incompatibility of libraries, versions, languages and operating systems. Therefore it is suggested that we look to the most successful recent model of re-use - the using of other people's documents and programs off remote Web servers. We suggest a model where the agent minds and agent worlds *stay* at the remote server and are used from there, instead of being installed.
3. We need *total language and platform independence*, so that researchers can concentrate on AI and not on networks.
4. This will be easier for virtual agents, but is not impossible with robotic agents, as the field of *Internet tele-robotics* demonstrates [44].

2 The World-Wide-Mind

The proposed scheme to address these issues is called the "World-Wide-Mind" (WWM). In this scheme, it is proposed that researchers construct their agent minds and worlds as *servers* on the Internet.

2.1 Types of servers

In the basic scheme, there are the following types of server:

1. A **World and Body server** together. This server can be queried for the current state of the world: `x` as detected by the body, and can be sent actions: `a` for the body to perform in the world.
2. A **Mind server**, which is a behavior-producing system, capable of suggesting an action: `a` given a particular input state: `x`. A **$\mathbf{Mind_M}$ server** is a Mind server that calls other Mind servers. For example:
   1. An **Action Selection or AS or $\mathbf{Mind_{AS}}$ server**, which resolves competition among multiple Mind servers. Each Mind server i suggests an action a_i to execute. The AS server queries them and somehow produces a winning action a_k. To the outside world, the AS server looks like just another Mind server producing `a` given `x`.

2.2 Types of Societies

By allowing Mind servers call each other we can incrementally build up more and more complex hierarchies, networks or societies of mind. We will call any collection of more than one Mind server acting together a *Society*. A Society is built up in stages. At each stage, there is a single Mind server that serves as the interface to the whole Society, to which we send the state and receive back an action.

1. A $Mind_M$ server calls other Mind servers. To run this Society you talk to the $Mind_M$ server.
2. A $Mind_{AS}$ server adjudicates among multiple Mind servers. To run this Society you talk to the $Mind_{AS}$ server.

2.3 Types of users

1. **A non-technical client user** - essentially any user on the Internet. Basically, the client user will run *other people's minds* in *other people's worlds*. Without needing any technical ability, the client should be able to do the following:
   1. Pick one Mind server to run in one World. Even this apparently simple choice may be the product of a lot of hard work - in picking 2 suitable servers that work together. So it is suggested that the client can present the results of this work for others to use at some URL. No new server is created, but rather a "link" to 2 existing servers with particular arguments.

2. Even a non-technical client may be able to construct a Society. For instance: Select a combination of remote Mind servers, a remote AS server to resolve the competition between these, and a World server to run this Society in. To be precise: Pick a $Mind_{AS}$ server, pass it a list of Mind servers as a startup argument, and then just pick a World to run the $Mind_{AS}$ server in. Again the client can present the results of his work for others to use. Again, no new server is created, but rather a "link" to 2 existing servers with particular arguments.

2. **A technically-proficient server author** - again any user on the Internet, if they have the ability. They will need to understand how to construct a server, but their understanding of AI does not *necessarily* have to be profound. For example:
    1. Write a *wrapper* around an existing, working Mind server, i.e. Write a new $Mind_M$ server. The most simple type of wrapper would not provide any actions itself, but just selectively call other servers: *"If input is x then do whatever Mind server M1 does - otherwise do whatever Mind server M2 does."*
    2. An AI-proficient server author might try writing a $Mind_M$ server that attempts to provide some actions itself: *"If input is x then take this action a - otherwise do whatever the old server does."* The author may need little understanding of how the existing Mind server works. If overriding it in one area of the input space doesn't work (doesn't perform better) he may try overriding it in a different area.
    3. At the most advanced level, AI researchers would write their own servers from scratch. But it is envisaged that even AI researchers will make use of the techniques above.

2.4 Low-bandwidth communication

If Mind servers come from diverse sources, written according to different AI methodologies in different languages, and do not understand each other's goals, there is a limit to how much information they can usefully communicate to each other. A central question is: **What information does the AS server need from the Mind servers to resolve the competition?** For example, if it just gets a simple list of their suggested actions: a_i it seems it could do little more than just pick the most popular one. For more sophisticated Action Selection, the Mind server needs to provide *more information*. We first consider schemes where competition is resolved using numeric weights rather than symbolic reasoning. For example, Mind server i tells the AS server what action a_i it wants to take, plus a weight W_i expressing how much it is willing to "pay" to win this competition. We define the following weights:

1. The "Q-value" defines how good this action is in pursuit of the Mind server's goal, Mind server i might build up a table $Q_i(x,a)$ showing the expected value for each action in each state.

2. The "W-value" defines how *bad* it is for the Mind server to lose this competition. This depends on what action will be taken if it loses. Mind server i may maintain a table $W_i(x)$ defining how bad it is to lose in each state. It may judge the badness of a *specific* action a by the quantity: $Q_i(x,a_i) - Q_i(x,a)$.

The usage of Q and W comes from [21]. For the differences between Q and W see [21, §5.5, §6.1, §16.2]. Higher-bandwidth communications than numeric weights would seem difficult if we are not to impose some structure on what is inside each Mind server. So I begin the WWM implementation with a *sub-symbolic Society of Mind*, rather than a symbolic one.

2.5 What is the definition of state and action?

We have so far avoided the question of what is the exact data structure that is being passed back and forth as the state or action. It seems that this definition will be different in different domains. This scheme proposes that we allow different definitions to co-exist. Each server will explain the format of the state and action they generate and expect (most of the time this will just involve linking to another server's definition).

2.6 The name "The World-Wide-Mind"

The name "The World-Wide-Mind" makes a number of points:

1. **The mind stays at the server:** The mind will be *literally* decentralised across the world, with parts of the mind at different remote servers.
2. **Parts of the mind are separate from each other:** The important thing is not the separation of *mind from world*, but the separation of different parts of the mind *from each other*, so they can be maintained by different authors.
3. **This is separate from the Web:** This is a different thing to the *World-Wide-Web*. During the recent rise of the Internet, many people have talked about seeing some sort of "global intelligence" emerge [3]. But these writers are talking about the intelligence being embodied in the *humans* using the network, plus the pages they create [32], or at most intelligence being embodied implicitly in the hyperlinks from page to page [18, 14]. Claims that the network *itself* might be intelligent are at best vague and unconvincing analogies between the network and the brain [37]. For a *real* society of mind or network mind, we need a network of *AI programs* rather than a network of pages and links.
4. **This may not even *interact* with the Web:** This is separate from existing work that might go under the name of "AI on the Web", namely, AI systems learning from the Web. A WWM system is not *necessarily* interested in learning from or interacting with the current Web or its users.

3 How the WWM will be used in AI

We imagine that a scheme such as this is inevitable in AI - that the days of isolated experiments are numbered. For a detailed discussion see [24]. Briefly, it addresses these issues:

1. **Duplication of Effort** - Until now, sharing work has been so difficult that researchers tend to build their own agent minds and worlds from scratch, duplicating work that has been done elsewhere. There have been a number of attempts to re-use agent minds [40, 43] and worlds [11], but the model of re-use often requires installation, or even a particular programming language. Here we propose a language-independent model of remote re-use. [31] is probably the closest previous work to this philosophy.
2. **Unused agents and worlds** - Having invented a robotic or agent testbed, few experiments are often done with it. For example, I was in fact one of the first people to put an AI mind online [51, 52], an "Eliza"-type chat program in 1989 [19]. Many people talked to it, but soon (by 1990), it had ceased to interact with the world. A *brief*, finite interaction with the world, seen by only a few people, is the *norm* in autonomous agents research. In this field it has become acceptable not to have direct access to many of the major systems under discussion. How many action selection researchers have ever *seen* Tyrrell's world running, for example? [45] How many robotics researchers have ever *seen* Cog move (not in a movie)? [7] Due to incompatibilities of software and expense of hardware, we accept that we will never experiment with, or even *see*, many of these things ourselves, but only read papers on them.
3. **Making AI Science - 3rd party experimentation** - Building your own agent world also means your new algorithms are not tested in the same world as previous algorithms. How to *prove* one agent architecture is better than another has become an important issue. [8] points out that, essentially, no one uses each other's architectures, because they are not convinced by each other's tests. In any branch of AI, the existence of objective tests that cannot be argued with tends to provide a major impetus to research. This is one of the reasons for the popularity of *rule-based games* in AI, and, more recently, *robotic soccer* [31]. [9] suggests the setting up of a website repository of standard tests for adaptive behavior. The WWM goes further than that, where the standard test worlds need not even be installed, but are run remotely.

   And the WWM goes further to support testing. By its emphasis on *3rd party* experimentation, algorithms will be subjected to constant examination by populations of testers with no vested interest in any outcome. 3rd parties will test servers in environments their authors never thought of, and combine them with other servers that their authors did not write. The servers should get a much more thorough testing than their own authors could ever have given them.
4. **Minds will be too complex to be fully understood** - Finally, there is definitely something in the evolutionary criticism that advanced artificial minds may be too complex to be *understood*. In the system we propose, of a vast network of

servers calling other servers, each individual *link* in the network will make sense to the person who set it up, but there is no need for the system as a whole to be grasped by any one individual.

4 Objections to the model

For a discussion of possible objections see [24]. Here we mention a few points:

1. **Models of Broken links and Brain Damage** - The WWM will need well-understood models of *fault-tolerant* artificial minds. e.g. In [21] I explicitly addressed the issue of brain damage in a large society of mind [21, §17.2.2, §18]. The reader might have wondered what is the point of a model of AI that can survive brain damage. Here is the point - a model of AI that can survive *broken links*.
2. **It is premature at symbolic level to attempt to define mind network protocols.** - Probably true. Researchers have long debated symbolic-AI knowledge-sharing protocols, with [13] arguing that it is premature to define such protocols. Recently this debate has continued in the Agents community as the debate over *agent communication languages* [27] and, to some extent, XML [2]. Agreement is weak, and it may be that the whole endeavour is still premature. For example [35] attempts to implement a Society of Mind on the Internet, but they insist on a symbolic model, with which they make limited progress. We argue, though, that it is not premature to start defining mind network protocols at the sub-symbolic level.
3. **The network is not up to this yet.** - Possibly true. But that will change.
4. **"Agents" researchers have already done this.** - Apparently not. There are some major differences between this and the Agents approach:
    1. **Agents researchers imagine that agents should be *installed*.** I disagree. Agents should be *servers*.
    2. **Agents have 1 localised (installed) mind per body. I have multiple remote minds (servers) per body.** Consider that *Distributed AI* (DAI) [42, 33] has split into two camps:
        1. Distributed Problem Solving (DPS) - where the Minds are cooperating to solve the *same problem* in one Body.
        2. Multi-Agent Systems (MAS) - where the Minds are in different Bodies. We have 1 mind - 1 body actors, and coordination of multiple actors. This is what the field of *"Agents"* has come to mean. Indeed [33, §4.3] makes clear that our servers here are not Agents.

        This is neither of these, but is multiple minds solving multiple problems in one body. It is closer to *Adaptive Behavior* and its interest in whole, multi-goal creatures whose goals may simply *conflict*. Previous work in ALife has been more towards the MAS approach, e.g. in [36] it is a *society of agents* that is distributed across the network, not a *single agent mind*.
    3. In short, Agents researchers simply *aren't* trying to solve the problem of **how to divide up the work in AI, and link the work of multiple laboratories**, that we addressed at the start of this paper.

5 Implementation

We now define a way of implementing this idea on today's network. We suggest the standard client-server model of short, limited-length transactions. The server responds to a short query with a response within a limited time. The server does not know when, if ever, it will receive the next query. The client software, which is driving a single top-level Mind and World, implements a program like:

1. For each server:
   - Connect to server - Send "New run" command - Receive unique run ID
2. Repeat:
   1. Connect to World - Send ID - Query state - Get state `x`
   2. Connect to Mind - Send ID - Send state `x` - Get action `a`
   3. Connect to World - Send ID - Send action `a` - Get new state `y`
   4. Connect to Mind - Send ID - Tell it new state `y` - Receive confirm
3. For each server:
   - Connect to server - Send ID - Send "End run" - Receive confirm

The unique run ID is because the server may be simultaneously involved in many other runs with other clients. We will not lay down what the client algorithm should be (for example, should it implement time-outs). It may implement any general-purpose algorithm using the server queries. Similarly, a server may implement any algorithm it likes provided it responds to the set of queries expected of it (e.g. it may *itself* be a client to yet another server). So the definition of the WWM really comes down to just the definition of the possible queries and responses of WWM servers. For a detailed list of queries see [23].

What technology should we use to implement these queries? I suggest one overriding objective: *That the WWM server authors be required to know as little as possible to get their servers on the network.* The server authors are interested in AI, not necessarily in networks. They may only know AI programming languages. They may have never written a network application, and may not want to learn. As a result, it is proposed that the basic implementation of the WWM be done using CGI across HTTP. Every AI programmer has access to a HTTP server with CGI, and every AI programmer can write a program that receives stdin and writes to stdout. For further justification see [24].

What format should the data transmitted be? We suggest plain text XML [2]. We need *some* format, and XML provides an *extensible* format, where the invention of new queries won't crash old servers. For examples of XML encoding of the server queries see [23]. To summarise, all requests to a WWM server are requests to a CGI program on a Web server: `http://site/path/program`. All arguments (including the type of WWM query being sent) are passed as XML in stdin. The server writes the response as XML to stdout.

6 How to implement some existing agent architectures as networks of WWM servers

6.1 Sub-symbolic Mind servers

[24] shows in more detail than is possible here how many existing agent architectures can be implemented as networks of WWM servers, including:

1. Internet tele-robotics [53, 44]. One issue is if multiple clients can connect to the World at the same time [41, 34, 15, 38, 16].
2. The Subsumption Architecture [4, 5], using a hierarchy of Mind_{M} servers.
3. Serial models [39], using a master Mind_{M} server.
4. Any state-space learner [10], including Reinforcement Learning [25, 48].
5. Hierarchical Q-Learning [26], using a master Mind_{AS} server.
6. Static measures of W-values (e.g. W=Q) [21, §5.3].
7. Dynamic measures of W-values [21, §5.5], including W-learning [21, §5, §6], including where Minds do not share the same suite of actions [21, §11, §13], e.g. Minds from different authors.
8. Strong and Weak Mind servers [21, §8, §C, §D], passing "mind strength" as an argument to the server.
9. Mind servers with different senses in the same Society [21, §6.6, §7, §8, §10]. The top-level Mind server sees the full state.
10. Global Action Selection decisions [21, §14], including Minimize the Worst Unhappiness and others [21, §F]. All such schemes in [45] etc. Some require a Mind_{AS} server that makes *multiple* queries of each Mind server.
11. Nested Q-learning [12], and Nested W-learning [21, §18.1], where the action returned is: "do whatever server `k` wants to do".
12. Feudal Q-learning [48] and Feudal W-learning [21, §18.2], where the Mind server accepts commands: "Take me to state `c`"
13. Economy of Mind [1], where the Mind_{AS} server will redistribute payments.

6.2 Symbolic Mind servers - single

There are a vast number of models of agent mind, whether hand-coded, learnt or evolved, symbolic or non-symbolic, that will repeatedly produce an action given a state. Most of these could be implemented as WWM servers without raising any particular issues apart from having to agree on the format of state and action with the World server. For example, "Eliza"-type chatbots [19, 49].

6.3 Symbolic Mind servers - multiple

The difficulty arises when we consider *competition* between multiple symbolic Minds. So far we only defined a protocol for conflict resolution using numeric weights. Higher-bandwidth communication leads us into the field of *Agents* and its problems with defining *agent communication languages* (formerly symbolic AI knowledge-sharing protocols) that we discussed above.

A lot could be done, though, without having to define symbolic queries. Master Mind_M servers can switch from server to server. The drawback is the Mind_M server needs a lot of intelligence. This relates to the "homunculus" problem, or the need for an intelligent headquarters [21, §5]. Another possibility is the subsidiary Mind servers can be symbolic, while the master Mind_{AS} server is sub-symbolic - e.g. a Hierarchical Q-learner which just *learns* which subsidiary symbolic Mind server to let through.

7 Future work

This is clearly the start of an open-ended program of implementation and testing. The first prototype implementation is now described in [47]. Immediate issues are:

1. **Define the server queries** - Define the full list of server queries, arguments, responses and error conditions [23] and encode them in XML. Is this list sufficient to implement *all* current sub-symbolic agent minds and worlds?
2. **Define the client user view** - The basic question is whether the client user software can be provided through *existing Web browsers* (perhaps through a *WWM portal site*), or whether a *separate client application* needs to be installed.

8 Conclusion

There are two issues here - first, that we need a system of decentralised network AI minds, and second a proposed protocol for it. Even if the protocol here is not adopted, the first part of this paper (the need to decentralise AI) stands on its own. For further reading see [24]. For the first prototype implementation see [47]. The first test of this system may be in the domain of language evolution [46].

8.1 Endnote - Showing the world what a mind looks like

If a scheme like the WWM becomes successful, much of the user population of the Internet will gradually become familiar with minds made up of hundreds or even thousands of distributed components; minds that have little identifiable headquarters, but contain crowded collections of sub-minds, duplicating, competing, overlapping, communicating and learning, with *"alternative strategies constantly bubbling up, seeking attention, wanting to be given control of the body"* [21, §18.3]. Such models may be long familiar to AI researchers, but they are not much understood outside AI. The WWM scheme may help large numbers of people expand their imagination to think about what a mind *could* be.

References

1. Baum, E.B. (1996), Toward a Model of Mind as a Laissez-Faire Economy of Idiots, *13th Int. Conf. on Machine Learning*.
2. Bosak, J. and Bray, T. (1999), XML and the Second-Generation Web, *Scientific American*, May 1999.
3. Brooks, M. (2000), Global Brain, *New Scientist*, 24th June 2000.

4. Brooks, R.A. (1986), A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation* 2:14-23.
5. Brooks, R.A. (1991), Intelligence without Representation, *Artificial Intelligence* 47:139-160.
6. Brooks, R.A. (1997), From Earwigs to Humans, *Robotics and Autonomous Systems*, Vol. 20, Nos. 2-4, pp. 291-304.
7. Brooks, R.A. et al. (1998), The Cog Project: Building a Humanoid Robot, *Computation for Metaphors, Analogy and Agents*, Springer-Verlag.
8. Bryson, J. (2000), Cross-Paradigm Analysis of Autonomous Agent Architecture, *JETAI* 12(2):165-89.
9. Bryson, J.; Lowe, W. and Stein, L.A. (2000), Hypothesis Testing for Complex Agents, *NIST Workshop on Performance Metrics for Intelligent Systems*.
10. Clocksin, W.F. and Moore, A.W. (1989), Experiments in Adaptive State-Space Robotics, *AISB-89*.
11. Daniels, M. (1999), Integrating Simulation Technologies With Swarm, *Workshop on Agent Simulation: Applications, Models, and Tools*, Univ. Chicago, Oct 1999.
12. Digney, B.L. (1996), Emergent Hierarchical Control Structures: Learning Reactive/Hierarchical Relationships in Reinforcement Environments, *SAB-96*.
13. Ginsberg, M.L. (1991), Knowledge Interchange Format: The KIF of Death, *AI Magazine*, Vol.5, No.63, 1991.
14. Goertzel, B. (1996), *The WorldWideBrain: Using the WorldWideWeb to Implement Globally Distributed Cognition*, goertzel.org/papers/wwb.html
15. Goldberg, K. et al. (1996), A Tele-Robotic Garden on the World Wide Web, *SPIE Robotics and Machine Perception Newsletter*, 5(1), March 1996.
16. Goldberg, K. et al. (2000), Collaborative Teleoperation via the Internet, *IEEE Int. Conf. on Robotics and Automation (ICRA-00)*.
17. Harvey, I.; Husbands, P. and Cliff, D. (1992), Issues in Evolutionary Robotics, *SAB-92*.
18. Heylighen, F. (1997), Towards a Global Brain, in *Der Sinn der Sinne*, Steidl Verlag, Gottingen, pespmc1.vub.ac.be/papers/GBrain-Bonn.html
19. Humphrys, M. (1989), The MGonz program, www.compapp.dcu.ie/~humphrys/eliza.html
20. Humphrys, M. (1991), *The Objective evidence: A real-life comparison of Procedural and Object-Oriented Programming*, technical report, IBM Ireland.
21. Humphrys, M. (1997), *Action Selection methods using Reinforcement Learning*, PhD thesis, University of Cambridge, www.compapp.dcu.ie/~humphrys/PhD
22. Humphrys, M. (1997a), *AI is possible .. but AI won't happen: The future of Artificial Intelligence*, "Next Generation" symposium, Jesus College, Cambridge, Aug 1997.
23. Humphrys, M. (2001), *The World-Wide-Mind: Draft Proposal*, Dublin City University, School of Computer Applications, Technical Report CA-0301.
24. Humphrys, M. (2001a), The World-Wide-Mind: A protocol for building a client-server sub-symbolic Society of Mind distributed over the network, submitted to *Adaptive Behavior*.
25. Kaelbling, L.P.; Littman, M.L. and Moore, A.W. (1996), Reinforcement Learning: A Survey, *JAIR* 4:237-285.
26. Lin, L-J (1993), Scaling up Reinforcement Learning for robot control, *10th Int. Conf. on Machine Learning*.
27. Martin, F.J.; Plaza, E. and Rodriguez-Aguilar, J.A. (2000), An Infrastructure for Agent-Based Systems: an Interagent Approach, *Int. Journal of Intelligent Systems* 15(3):217-240.
28. Minsky, M. (1986), *The Society of Mind*.
29. Minsky, M. (1991), Society of Mind: a response to four reviews, *Artificial Intelligence* 48:371-96.
30. Nilsson, N.J. (1995), Eye on the Prize, *AI Magazine* 16(2):9-17, Summer 1995.
31. Noda, I. et al. (1998), Soccer Server: A Tool for Research on Multiagent Systems, *Applied Artificial Intelligence* 12:233-50.

32. Numao, M. (2000), Long-term learning in Global Intelligence, *17th Workshop on Machine Intelligence (MI-17)*.
33. Nwana, H.S. (1996), Software agents: an overview, *Knowledge Engineering Review*, 11(3).
34. Paulos, E. and Canny, J. (1996), Delivering Real Reality to the World Wide Web via Telerobotics, *IEEE Int. Conf. on Robotics and Automation (ICRA-96)*.
35. Porter, B.; Rangaswamy, S. and Shalabi, S. (undated), *Collaborative Intelligence - Agents over the Internet*, Undergraduate final year project, MIT Laboratory of Computer Science.
36. Ray, T.S. (1995), *A proposal to create a network-wide biodiversity reserve for digital organisms*, Technical Report TR-H-133, ATR Research Laboratories, Japan.
37. Russell, P. (2000), *The Global Brain Awakens*, Element Books.
38. Simmons, R.G. et al. (1997), Xavier: Experience with a Layered Robot Architecture, ACM SIGART *Intelligence* magazine.
39. Singh, S.P. (1992), Transfer of Learning by Composing Solutions of Elemental Sequential Tasks, *Machine Learning* 8:323-339.
40. Sloman, A. and Logan, B. (1999), Building cognitively rich agents using the SIM_AGENT toolkit, *Communications of the ACM*, 43(2):71-7, March 1999.
41. Stein, M.R. (1998), Painting on the World Wide Web: The PumaPaint Project, *IEEE / RSJ Int. Conf. on Intelligent Robotic Systems (IROS-98)*.
42. Stone, P. and Veloso, M. (2000), Multiagent Systems: A Survey from a Machine Learning Perspective, *Autonomous Robots*, 8(3), July 2000.
43. Sutton, R.S. and Santamaria, J.C. (undated), A Standard Interface for Reinforcement Learning Software, www-anw.cs.umass.edu/~rich/RLinterface/RLinterface.html
44. Taylor, K. and Dalton, B. (1997), Issues in Internet Telerobotics, *Int. Conf. on Field and Service Robotics (FSR-97)*.
45. Tyrrell, T. (1993), *Computational Mechanisms for Action Selection*, PhD thesis, University of Edinburgh.
46. Walshe, R. (2001), The Origin of the Speeches: language evolution through collaborative reinforcement learning, submitted to *3rd Int. Workshop on Intelligent Virtual Agents (IVA-2001)*.
47. Walshe, R. and Humphrys, M. (2001), First Implementation of the World-Wide-Mind, poster to appear in *ECAL-01*.
48. Watkins, C.J.C.H. (1989), *Learning from delayed rewards*, PhD thesis, University of Cambridge.
49. Weizenbaum, J. (1966), ELIZA - A computer program for the study of natural language communication between man and machine, *Communications of the ACM* 9:36-45.
50. Wilson, S.W. (1990), The animat path to AI, *SAB-90*.
51. Yahoo list of AI programs online, yahoo.com/.../Web_Games/Artificial_Intelligence
52. Yahoo list of ALife programs online, yahoo.com/.../Artificial_Life/Online_Examples
53. Yahoo list of robots online, yahoo.com/.../Devices_Connected_to_the_Internet/Robots