

Collaborative Online Development of Modular Intelligent Agents

by Ciarán O’Leary, Mark Humphrys and Ray Walshe

There is a growing consensus that natural intelligence is likely to be composed of multiple diverse algorithms and knowledge representation formats, with hundreds (or thousands) of specialised subsystems collaborating in some network (or ‘Society of Mind’). This is in contrast to the ‘one size fits all’ approach of many popular algorithms in Artificial Intelligence. If such diverse, extremely hybrid artificial agents are to be built, many authors will need to be involved, each with a different area of specialization. And if the future of AI is one of multiple authors and multiple laboratories building vast collaborative systems, how is this work to be organized?

The World-Wide-Mind project (w2mind.org) proposes that AI research should be organized online, using an agreed, stripped-down, simplified protocol for communication between different subcomponents of the artificial minds. This is related to large online ‘open source’ collaborative projects, but differs from them in that it is not proposed that the components actually have to be open source. In many cases, users of a third party’s subsystem (or ‘sub-mind’) need not understand how it works, but only how to use it as a component in the larger ‘mind’ they are constructing. By keeping the framework as simple as possible and by populating the Web with multiple algorithms, the task of the agent developer is reduced to selecting the algorithms and plugging them into the agent mind. What results are new and novel agent minds composed of many diverse components, designed and implemented by multiple, loosely coupled collaborators.

The project was started in 2001 by Dr Mark Humphrys at Dublin City University, as an attempt to facilitate the integration of the many diverse components of agent minds into whole minds. Though much fascinating research is performed in focused subdomains of AI, such as perception, vision, navigation and language processing, there has traditionally been less effort directed towards the problem of integrating all these components into single, whole agents. Within the past two decades the animat and behaviour-based communities have contributed several modular agent architectures (Brooks’ subsumption architecture

being the best known). These are naturally inclined towards dividing the agent mind into components and integrating the components to create whole minds. Such approaches, generally referred to as ‘Society of Mind’ approaches after Minsky’s famous book, recognize the importance of masking the implementation details of individual components from other components, while retaining a public interface through which they can interact. This reflects the modern approach to software engineering, where modular development is seen to support reuse, maintainability, extensibility and in general the creation of software systems where no individual is required to understand the entire system.

A natural extension of the modular approach in software engineering is the distributed modular approach. Middleware standards such as CORBA and Web Services have provided developers with the means to connect modules over the Internet without knowing anything about remote modules apart from their URL and interface. Although middleware platforms such as JADE (Java Agent Development Environment) provide support for multiple interacting autonomous agents, there has been little or no attention directed towards how the distributed modular approach to software engineering can support the modular implementation of whole agent minds, or how the integration of independent remote components can further our understanding of the diversity required for artificially intelligent agents.

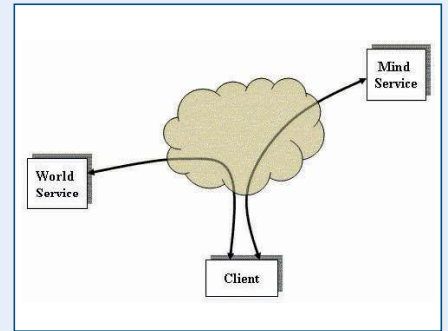


Figure 1: The basic World-Wide-Mind architecture. Client queries both world and mind services for state and action data, passing the required data between the services.

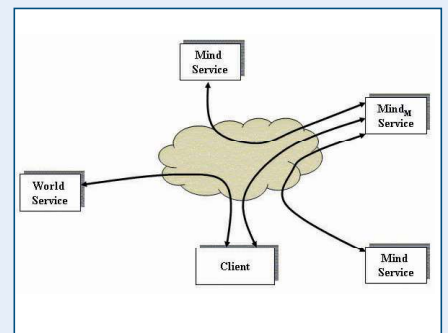


Figure 2: A Society of Mind on the World-Wide-Mind. The MindM service developer creates a switch that selects the appropriate mind service for a given state, delegating responsibility to that service.

The Society of Mind Markup Language (SOML) is the protocol of the World-Wide-Mind (WWM), in much the same way as HTTP is the protocol of the World-Wide-Web. The WWM defines three types of entities, each of which interacts with the others using this protocol.

A world service on the WWM represents some sort of problem, eg a robot in a complex environment, or a creature in some virtual reality. The world service presents a public interface that can be queried for the state of the world (such as the sensory inputs of the creature) and can also be sent an action that the creature should execute.

A mind service represents some action-producing algorithm specifically

designed for a given world service. The mind service will return an action when it is presented with the state of the world. The implementation details of the algorithm are hidden behind the interface, so any state maintained by the algorithm is transparent to the calling entity.

A client is used to plug a mind service into a world service. When launched, the client requests the URLs of the world service and mind service from the user. It then executes a run by querying the world for its state and the mind for its action based on this state, before querying the world again to execute the action. The run iterates through these steps until its natural conclusion (death of the virtual creature, expiration of time permitted for robot), at which point a score is recorded on a publicly accessible scoreboard at the world service.

With multiple online mind services, a third-party developer could select a set of useful mind services and provide a

high-level switch that acts as a mind service, but simply decides at each timestep to which of the multiple mind services it will delegate responsibility. Such switches, named MindM services, facilitate the creation of Societies of Mind, where the many mind services are authored by a diverse set of independent researchers.

The technology required to implement mind services, world services and clients is kept deliberately simple, well below the threshold of knowledge required for complex distributed systems development. SOML entities interact over HTTP, meaning that the creation of a service requires only basic knowledge of Web programming (CGI, Java Servlets, ASP etc).

The framework has been tested with several classes of undergraduate students at Dublin City University and Dublin Institute of Technology. Currently, two world services are online, although since

the protocols are in the public domain, anyone can contribute to the project by developing their own world service, or implementing a mind or MindM service for existing worlds.

A key benefit of the framework is that it is open to all. Complex minds can be built by integrating the product of advanced research by professionals with the novel algorithms of interested amateurs. The result will be complex arrangements of never-before-seen algorithms, possibly bringing us closer to understanding how our own minds and the minds of other animals are made so much richer in the presence of widely diverse components.

Link:

<http://w2mind.org>

Please contact:

Ciarán O'Leary

Dublin Institute of Technology, Ireland

Tel: +353 1 402 4718

E-mail: info@w2mind.org

CE-Ants: Ant-Like Agents for Path Management in the Next-Generation Internet

by Otto Wittner and Bjarne E. Helvik

In general, today's Internet can provide only 'best effort' connections, with no quality of service (QoS) guarantees. Both new and popular existing applications and services would benefit from connections with QoS guarantees, eg minimum capacity, maximum delay, continuity of service and availability. How can connections in the Next-Generation Internet (NGI) be managed such that desired QoS guarantees can be provided while the robustness of today's network is maintained? We pursue an approach inspired by the robustness and emergent behaviour seen in swarms combined with stochastic optimization techniques.

One of the fundamental design objectives of today's Internet was connectivity robustness. First of all, two terminals connected to the network should be able to communicate. Secondly, the network should stay operative and provide reliable communication even when a significant number of network nodes and transmission links fail. Hence packet forwarding is controlled in a fully distributed manner and routing units are more or less independent of each other. In this way, the Internet provides connectivity to everybody with a high probability. Unfortunately, no QoS guarantees or differentiations can be given.

Resource Control and Path Management

It is common for network operators to use over-provisioning combined with virtual connections realized by multiprotocol label switching (MPLS), as a way of providing sufficiently low packet losses and delays, load distribution and capacity control. Continuity of service, short downtimes and high availability are obtained by supplementing the traditional Internet restoration techniques with protection switching based on MPLS and/or in the underlying optical layer. These techniques require central-

ized planning of resource utilization and path management, which significantly weakens the inherent robustness and autonomy of the traditional Internet.

Furthermore, when multiple paths exist between two communicating terminals, today's routing systems by default forward all packets along the path regarded as having the lowest cost. The exact definition of the lowest cost is generally decided by the network operator, and is not influenced by applications and users. Hence path management does not handle service differentiation. As for MPLS