Assembling Hybrid Minds

Oisín Mac Fhearaí, Ray Walshe and Mark Humphrys

School of Computing, Dublin City University, Dublin, Ireland
{omacfh, Ray.Walshe, humphrys}@computing.dcu.ie

Abstract. An important topic in artificial intelligence is the decomposition of larger problems into smaller ones that can be solved by more granular solutions. Modular, "hybrid" solutions tend to be built from components mostly created by the same programmer or the same team. In the World-Wide Mind project, many users submit their individual programs to solve problems in a collaborative and competitive environment. In this paper a method of combining some of these solution programs together to create a higher-level program which performs better than the individual "subminds" is presented. The auto-construction of these hybrid solutions is also described and the results discussed.

1 Introduction

It is commonly presumed that biological intelligence consists of a multitude of specialised components co-operating with each other, rather than a general algorithm which solves all problems. The authors are determining whether artificial intelligence can be scaled up by constructing larger scale solutions to complex problems from the combined work of multiple authors. Although development, testing and evaluation of solutions is faster on a local machine, as solutions grow to be composed of more and more subprograms it is inevitable that a distribution of computing resources will be required.

An examination of the literature on this topic has found collaborative and competitive efforts in A.I. research and problem-solving, such as the DARPA Challenge [1] and RoboCup [2], but similar work facilitating the building of larger solutions from the programs of multiple authors has not been found in the literature search.

The World-Wide Mind (W2M) project was started in 2001 to facilitate the scale up of artificial intelligence, by distributing action-selecting agents (which we refer to as "*minds*") and problem environments (which we call "*worlds*") on the internet, and by allowing minds to call other minds and thus facilitate building *hybrid* minds from multiple programs which may have been written by many different authors [3].

This paper gives a general overview of the W2M architecture and ongoing research into the automatic construction of hybrid minds, as well as some experimental results.

2 Related Work

The concept of combining many weaker subprograms to create a strong hybrid mind bears some similarity with ensemble learning methods [4] in machine learning. In ensemble learning, a set of weak learners are combined and make better predictions collectively. In the work presented in this paper, however, the subminds are programs written by humans which usually aim to solve the target problem on their own. The learning element is a three-phase process consisting of *1)* an analysis of the individual performances of each mind, *2)* ranking the minds using a variety of metrics, and *3)* selecting a small number of minds to use in the construction of higher-level hybrid minds.

A large body of work exists on modular cognitive architectures, such as the Society of Mind [5], the subsumption architecture [6] and CogAff [7]. These architectures often make assumptions about how information should be represented and communicated, how systems should be organised and what their responsibilities are. The authors believe that with fewer requirements and less restrictions placed on mind authors, a greater level of participation and contribution will be facilitated.

Our approach is largely architecture- and implementation-agnostic. One major goal of the W2M project is to minimise the technology barrier to entry for researchers, students and casual programmers to create and test their own minds and worlds freely [8]. Consideration for the inclusion of advanced mechanisms present in existing modular architectures (e.g. k-lines in Society of Mind [5], alarms and perceptual filters in CogAff [7]) and decisions about knowledge representation is reserved for future work. Rather, world and mind designers are free to define the world state and action representations, as well as the interactions between subminds, in almost any way they see fit.

In the current implementation of the World-Wide Mind we have selected a scheme defined by the the XML-based Society of Mind Markup Language (SOML [9]) where minds return a suggested action to perform in response to a *getaction* message containing the current state observable by the agent in the world, and worlds respond to *getstate* and *takeaction* methods as outlined in figure 2.

Some collaboration and sharing exists within artificial intelligence research; for example, a number of websites serve as repositories for machine learning code [10] and training datasets [11]. These repositories are useful, but the steps required to install or adapt an existing solution differ in each instance and there is little consistency in the types of programs and interfaces provided. Potential users must download the code and in some instances modify it to compile on their own machine, and must adapt the program to suit the interface and/or problem structure they wish to solve, if indeed the program is suitable for addressing the chosen problem.

RoboCup [2] and the DARPA Grand Challenge [1] are closer to what we wish to achieve, but the problem domains are specific and there is no clear way to build and share hybrid minds. These projects focus on competition rather than explicitly on collaboration and re-use.

The ideas behind this work are described further in [3] and [12].

3 Overview

3.1 Action-Selection

The action selection problem in artificial intelligence consists of an agent situated in a real or simulated environment. The agent periodically polls the state of the environment through its sensors and attempts to perform an action which will lead towards a problem being solved or a more beneficial state being reached. Some problems consist of multiple conflicting goals, where the problem-solving agent must prioritise the available options and act opportunistically where possible. In some situations, multiple actions may satisfy the same goal.

3.2 Mind and World Services

We define a *world* to mean a type of problem such as chess, poker or a predator-prey simulation. A *mind* is a program designed to control an agent operating in the world and solve the underlying problem (for example, winning a chess or poker game, or evading the predator for as long as possible).

At each discrete timestep in the world, a *state* is provided (a snapshot of the current state of the world) to the mind instance, representing the set of sensory percepts available at that moment. The state seen by the mind may contain all the information about the current state of the world - in chess, for example - or may include only partial information, as human sensory inputs do. The mind may use the state information to decide which *action* to choose (from an array of valid actions available to the mind). Taking this action will alter the world and thus causes the world instance to generate a new state. Eventually, the run (the sequence of state-action stimulus response steps) in the world will terminate, perhaps when a game is won or lost, or when a simulated creature dies, or when some fixed number of timesteps have occurred. At the end of a run, a *score* will be produced by the world to describe how well the mind performed, possibly over a number of aspects - for example, whether a chess game was won or lost, how many moves were played and how many times the opponent's king was in check.

Such an approach facilitates the creation of a *hybrid* mind (Mind-M) which queries one or more *subminds* for their suggested actions in any given state. A hybrid mind can itself be used as a submind in another hybrid mind, thus allowing for a hierarchy of mind services.

4 Architecture of the World-Wide Mind Framework

A general architecture diagram of the platform is shown in figure 1, highlighting the most important interactions between instances of world and distributed mind services. The *instance manager* in the figure represents the interface which allows the user to carry out runs and to upload new mind and world services; in our case this functionality is provided by a web interface on the World-Wide Mind server.

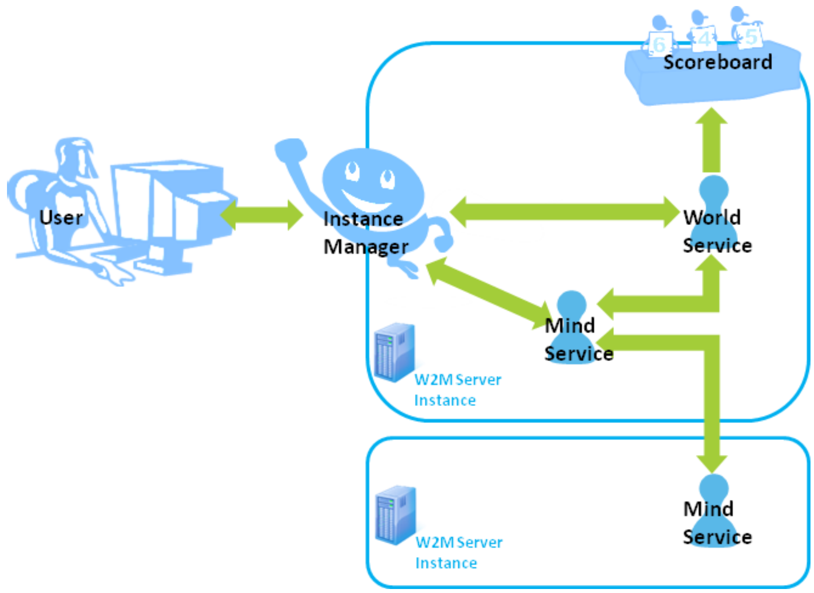

Fig. 1. Architecture diagram representing the main interactions between components.

To facilitate the creation of minds and worlds, and especially the construction of large hybrid minds which consult other minds as part of their decision-making process, we introduced a simple, uniform interface which minds and worlds must follow, representing the messages that may be sent to minds (getaction) and worlds (getstate and takeaction), explained in section 5.

This interface, coupled with the ability to upload a mind to a server and have it immediately appear as a service online, makes fast experimentation and composition of minds simpler (and will encourage people to try out the platform themselves).

Although we consider the W2M platform to be useful for teaching and exploration of problems, its main advantage over other solutions is that large hybrid minds will be created which query other subminds for suggested actions (which may themselves query other minds for actions and so on), and thus large-scale hierarchical problem-solving programs can be built from the work of many authors who do not need to know how and may not even understand how the other sub-components work.

However, since each level in the hybrid mind's hierarchy may have a number of branches, it is easy to see how the computational demands can rise exponentially. To cope with this problem, it must be possible to distribute the minds across different machines and networks.

5 The World-Wide Mind Server

In the first implementation of the W2M server, worlds and minds were embodied as web services and were assigned a URL by which they could be accessed over HTTP. These web services were hosted using the Apache Tomcat application server, so that messages between services consisted of a web request and response, with the content of the messages represented as an XML document. While this enabled connectivity across the internet, running a mind - and especially a hybrid mind composed of many remote minds - in a world was very slow due to overheads in the underlying Java Servlet technology and the use of HTTP to wrap messages.

To avoid this bottleneck, the authors have created a new server design which sends the same XML messages over a much simpler TCP protocol with very little overhead. This approach also takes advantage of the common cases where mind or world services are located on the same physical machine, by avoiding the network stack completely (and automatically). Despite these performance improvements, a distributed computation approach will still be necessary to scale up to bigger hybrid minds, but where network access can be avoided, it should be.

The initial messaging scheme involved a conversation where at each timestep:

1. **getstate**: the user requests the current state from the world,
2. **getaction**: the user asks the mind for an action based on this state, and
3. **takeaction**: the user forwards the suggested action to the world and receives a new state in response.

This message sequence leads to potentially long delays, especially if there is a significant round-trip network latency between the user and either the mind or world service. The newly designed scheme is further streamlined and performance improved by the world returning a new state immediately in response to a **takeaction** message, making the **getstate** message unnecessary after the first timestep.

Another enhancement was implemented whereby the user asks the mind to carry out a run with the world (via a **continuerun** message), and receives an asynchronous stream of messages from the mind containing the states seen and corresponding actions taken, as well as a score object representing how well the mind performed in this run.

The new optimisations implemented in this work, as well as new graphical display features and a generic scoreboarding system have been tested with undergraduate students and the results so far are encouraging.

6 Constructing Hybrid Minds

At the core of this research is the possibility of creating hybrid minds which query a number of subminds for their suggested actions and, hopefully, outperform

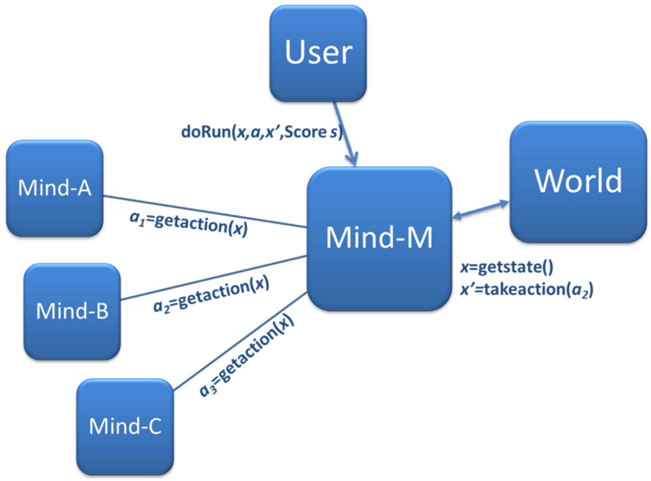

Fig. 2. An example of communication between the user and instances of a hybrid mind (Mind-M), three subminds and a world.

them in the problem world. In this section we describe the problem environment and a method the authors developed for creating a hybrid mind by ranking and selecting a set of subminds.

6.1 The Tyrrell Animal World

In the Tyrrell animal world, a mind controls a simulated animal in a two-dimensional grid world. The animal's goal is to mate as many times as possible, with a number of subgoals which support the primary goal, for example living for as many timesteps as possible, courting mates often, drinking water, avoiding predators and so on. Some of these goals conflict with others, and the animal's senses are often noisy and incomplete.

6.2 Student Assignments

A World-Wide Mind server was used to host several hundred minds for the Tyrrell animal world, as part of an assignment for undergraduate computer science students taking an artificial intelligence module. Minds were automatically ranked by their performance in this world, which prioritises mating and surviving as long as possible in a simulated environment.

For one assignment, a requirement was added that every student must submit at least one hybrid mind which delegates to one or more subminds. A call graph feature was implemented to keep track of calls between minds, and the scoreboard (see Table 1) at the end of the assignment period showed that nine of the top ten minds called at least one other submind during a run.

Rank	Mind	Mated	Steps	Runs	Minds called	Author
1	w2m.Exp9Standalone	75	4553	180	3	oisin
2	RobMindM	74	4567	10	2	rblestr
3	CowardlyMindFinal	63	3840	4	3	hands3
4	Mater	55	3380	2	1	murpha74
5	Bavaria	54	4306	1	1	dan
6	RTesterMind2	54	3836	0	1	rosshaugh
7	NeverMind	53	3294	6	3	rosshaugh
8	SFINALMIND2	52	4256	0	1	lawa3
9	CraigMind	52	3625	1	1	craig1928
10	TimiMind1	51	4086	0	None	milansatala

Table 1. The top ten highest-scoring minds for a modified Tyrrell animal world. The score components specified by the world author are "Mated" and "Steps", and the scoreboard automatically sorts entries in descending order. Links in the "Minds called" column display the set of subminds called (if any) by each mind.

6.3 Building the Hybrids

The selection of appropriate subminds is critical when constructing a hybrid mind. There are two subproblems here: firstly, how can we break down the problem into a number of distinct goals against which the existing Minds can be evaluated and ranked? And second, how can we design a hybrid Mind by making use of this ranking information?

6.4 Metrics and Ranking Minds

As described above, a collection of over 100 minds was uploaded by undergraduate students. This set of minds was used to perform a profiling analysis of the world state and score data. A large number of runs were performed with the submitted minds, and a record kept of all states seen and actions taken.

A series of metrics were created corresponding to important goals, and these metrics were used to rank minds by their performance on each goal. These metrics were constructed using a number of criteria, namely *1)* information from the score data only, *2)* information from the states seen at each timestep, and *3)* information from the state and the aggregate score data.

Some of the metrics and the three top-ranked minds for each metric were:

- Maximise number of times mated (*RobMindM, CowardlyMindFinal, CoalatronTheSecond*)
- Maximise number of mate encounters (*RobMindM, CowardlyMindFinal, mindafro*)
- Maximise average health of the animal (*Coalatron, MindUrBusiness, AmaAS3*)
- Minimise average level of thirst (*MindWater2, ThirstMind1, LOLMind*)

Once a set of subminds is chosen from a combination of rankings, a hybrid mind can be created which consults the best mind for each subgoal in a series of simple case-action tests, ordered by priority. For example, the logic for the hybrid mind created in experiment #5 in the table below can be expressed as follows:

```
if mate is nearby then return mating_mind.getaction()
if thirst > thirst_threshold then return drinking_mind.getaction()
if health < health_threshold then return survivor_mind.getaction()
return mate_finder.getaction()
```

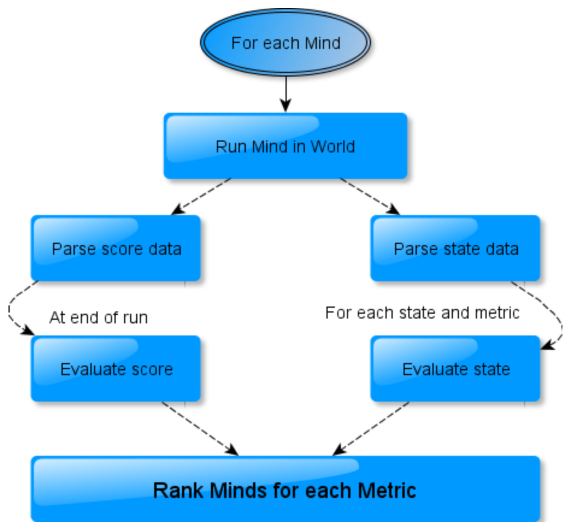

Fig. 3. The process of collecting data and ranking the minds on different metrics

7 Empirical Results

A number of experiments were carried out for hybrids composed of 2, 3 and 4 subminds, using different combinations of metrics, some focused solely on score information, some on state information, and some evaluating both the score and states seen. Table 2 summarises results over 100 runs for each test condition.

These tests produced a hybrid mind (experiment #5) which scored 10% better than any of the subminds and the best existing mind on the scoreboard. While the selection of appropriate metrics in these experiments required the designer to have some knowledge about the world, it is interesting to note that the best hybrid mind generated no actions itself; every action taken was provided by one of its four subminds. This shows that an evolved strategy using the existing available intelligent solutions at appropriate timesteps during the solving of the problem can result in a significant improvement in the overall result.

Exp#	Subminds	Type of analysis	Max mates	Max life	Median mates	Median life
1	-	Best existing mind	74	4567	-	-
2	4	Score	52	3762	20	2264
3	2	State	70	4840	48	4358
4	3	State	74	4877	48	4212
5	4	State	82	4797	49	4226
6	4	State (avg./step)	36	3269	23	2512
7	4	State+Score (avg.)	16	724	6	419
8	4	State+Score	72	4873	49	4213

Table 2. Results of the experiments in constructing hybrids. Experiment #1 is the existing best mind on the scoreboard, included as a control.

8 Discussion and Future Work

A more automated, statistical approach would identify the important subgoals in a more robust and fairer way than by relying on human intuition and domain knowledge to decompose the problem, so future work will be carried out to create a correlation analysis which will iterate through each available metric, trying to establish a ranking of metrics which most strongly influence the aggregate score (in this case, the number of times the virtual animal mates).

This can be iteratively tested by selecting metrics contained in the final score for each run only, such as how often the animal drank water or was injured, and then by searching the matrix for significant correlation between each of these attributes and the aggregate score. Alternative rankings can then be constructed based on individual performance on those metrics which appear to correlate most strongly (in either direction) with overall success. The best individual for each ranking can be selected as a “specialist” submind as before, and the performance of the resulting hybrid mind containing these “specialists” can be evaluated in the world.

Another approach is to perform further data-mining on the state information at every timestep to record more detailed metrics. This will be necessary for worlds which do not generate rich score information (for example, chess, which might only indicate whether the mind won, lost or drew a game). The stepwise mining of state data could produce equivalent metrics to facilitate the construction of hybrid minds at least as successful as those described in the previous section.

The work described so far requires that the delegation logic in the hybrid mind be written by hand. To produce a policy automatically by examining the chosen metrics would be a significant improvement. If, for example, an influential metric is found to be one that minimises thirst, then we could add a clause that queries the water-drinking submind when the current perceived thirst is greater than a given threshold. An autonomic controller would provide such flexibility and facilitate auto-construction of hybrid solutions.

Considering the success achieved by students building hybrids by hand, using each others’ work as subminds, we expect to see increased development of specialist “hybrid builders” in future. Some will be machine guided as in our work, and some will be assembled by manually using human observation and both would be a step toward the inevitable and in our view necessary scaling up of artificially intelligent programs.

This research was funded by IRCSET's Embark Initiative.

References

1. Buehler, M., Iagnemma, K., Singh, S. (eds.): The DARPA Urban Challenge: Autonomous Vehicles in City Traffic, George Air Force Base, Victorville, California, USA, Springer Tracts in Advanced Robotics, vol. 56. Springer (2009), http://dx.doi.org/10.1007/978-3-642-03991-1
2. Visser, U., Burkhard, H.D.: RoboCup: 10 years of achievements and future challenges. The AI Magazine 28(2), 115–132 (2007)
3. Walshe, R., Humphrys, M.: First implementation of the world-wide-mind. In: Kelemen, J., Sosík, P. (eds.) ECAL. Lecture Notes in Computer Science, vol. 2159, pp. 714–718. Springer (2001), http://link.springer.de/link/service/series/0558/bibs/2159/21590714.htm
4. Sollich, P., Krogh, A.: Learning with ensembles: how over-fitting can be useful (1995), http://preprints.cern.ch/cgi-bin/setlink?base=preprint&categ=scan&id=SCAN-9510301
5. Minsky, M.: The Society of Mind. Simon and Schuster (1985)
6. Brooks, R.A.: Elephants don't play chess. In: Maes, P. (ed.) Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, pp. 3–15. The MIT Press: Cambridge, MA, USA (1990), http://www.ai.mit.edu/people/brooks/papers/elephants.ps.Z
7. Sloman, A.: How many separately evolved emotional beasties live within us? (May 27 2002), http://citeseer.ist.psu.edu/429656.html; http://www.cs.bham.ac.uk/research/cogaff/sloman.vienna99.ps
8. Walshe, R., Humphrys, M., O'Leary, C.: Constructing complex brains: Building minds using sub-minds from biotechnology authors. In: 1st IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI-04), Toulouse, France, August 22-27, 2004, Proceedings (2004)
9. O'Leary, C., Humphrys, M., Walshe, R.: Constructing an animat mind using 505 sub-minds from 234 different authors. In: Schaal, S., Ijspeert, A., Billard, A., Vijayakumar, S., Hallam, J., Meyer, J. (eds.) From Animals to Animats 8. Proceedings of the Eighth International Conference on the Simulation of Adaptive Behavior (SAB'04). pp. 39–48. MIT Press (2004)
10. Kantrowitz, M.: CMU Artificial Intelligence Repository. http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/0.html (2009)
11. Asuncion, A., Newman, D.: UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/ (2009)
12. Mac Fhearaí, O., Humphrys, M., Walshe, R.: A high-speed architecture for building hybrid minds. In: 3rd International Conference on Agents and Artificial Intelligence (ICAART 2011). pp. 659–663. INSTICC