Building a hybrid Society of Mind using components from ten different authors

Ciarán O'Leary[1] and Mark Humphrys[2]

[1]Dublin Institute of Technology, School of Computing, Kevin St., Dublin 8, Ireland
Ciaran.OLeary@comp.dit.ie, www.comp.dit.ie/coleary
[2]Dublin City University, School of Computing, Glasnevin, Dublin 9, Ireland
humphrys@computing.dcu.ie, computing.dcu.ie/~humphrys

Abstract. Building large complex minds is difficult because we do not understand what the necessary components are or how they should interact. Even if the components were known it is difficult to see a situation where a single lab would have all the necessary expertise and manpower to be able to build the mind. One possible answer is to distribute the various components of the mind throughout the Internet, by letting different groups or individuals build parts of the mind. What would be needed then is a protocol that can be used to allow the components to interact, and a mechanism for managing the independent failure of components, and the switching between the required parts of the mind. This paper describes the first attempt to build such a mind. Using components that were developed independently by ten different authors, we have built a mind to solve a specific problem. Although the problem is a simple blocks world implementation, the issues dealt with, as well as the technology used, are relevant for larger and more complex minds. Robust minds require duplication and distribution. The Internet provides the ideal way to build in just such features into artificial minds.

1 Introduction

As writers such as Brooks [2] have argued, building large complex minds is difficult because we do not understand what the necessary components of the mind are or how they should interact. Even if the components (e.g. vision, navigation, planning, learning, etc.) and their interactions were known, it is difficult to see a situation where a single lab would have all the necessary expertise and manpower to be able to build the mind. A solution is to distribute the problem between all interested parties, each of whom can solve part of the problem, and then provide a way for the independently developed components to interact with each-other. The World-Wide-Mind project [8] suggests such a scheme and this paper describes the first implementation where a *Society of Mind* was built using the work of multiple (in this case ten) authors.

In other words, what we are building is a truly *hybrid* system. Many authors have argued that the mind is composed of many different types of algorithms [11], or that hybrid systems produce better solutions to problems [4], [9], [13]. The future of AI then, it seems involves hybrid symbolic/sub-symbolic systems built from quite different components.

1.1 Society of Mind

A Society of Mind, as described by Minsky [11] is a mind built from a multitude of agencies. Each agency is comprised of a set of agents and a master agent, which compete hierarchically for control of the body. Once a particular agency wins at one level, its sub-agents compete and so on. Duplication of functionality in multiple agents, as well as distribution of agents over the available hardware means that Societies of Mind are robust, and able to survive localised failure.

There is much agreement with Minsky that minds probably look like this, but such complex minds rarely get built. Modular, distributed minds have been built [1], [3], [6], [10], but in these cases the minds were all built by a single author/group, and all ran on localised hardware. Consequently, these minds could not incorporate the type of diversity that is necessary in a truly hybrid Society of Mind.

There are two requirements for a Society of Mind:

1. Agents: In order to build a Society of Mind, we require a diverse set of agents. To provide the necessary diversity an approach would be to allow different authors develop the components independently.
2. Arbitration: An algorithm is required for managing the competition between agents, as only one agent should have control of the body at any point in time. Several such algorithms already exist for distributed models of mind, some of which rely on the explicit design of the arbitration policy [1], [3], and others which allow this to be learned [6].

1.2 Contributions of this paper

Requirement 1 above is met by the World-Wide-Mind. This architecture provides a simple way for an author to make an agent, or sub-mind available for others to use remotely, without having to know exactly how the sub-mind is going to be used. This is described further is section 2 below.

Once these sub-minds are online, Societies of Mind can be built by independent authors, not involved in the construction of the individual components. Authors of these sorts of minds are required to satisfy requirement 2. Although the World-Wide-Mind does not provide any instructions on how this problem should be solved, this paper describes two simple arbitration mechanisms for a specific Society of Mind.

Section 3 below introduces the problem domain for which the Society of Mind was built. Section 4 describes a hand designed Society of Mind, where the arbitration policy follows a set of heuristics developed from observing the relative performance of each of the sub-minds. Section 5 shows how such a society could be developed dynamically. Section 6 provides a set of results for the dynamically created Society of Mind, which are then discussed in full in section 7.

The existence of the World-Wide-Mind alone does not solve the problem of building large complex minds. The core part of this problem is the arbitration between the independent modules. However, as demonstrated here, the World-Wide-Mind provides a way to collect the large, diverse set of agents/sub-minds required for complex minds. This was not previously possible.

2 The World-Wide-Mind

Humphrys, [8] introduced the World-Wide-Mind project to the Artificial Life community. This is an open, online scheme where individuals, whether AI researchers, ordinary programmers or even complete novices can build components of minds and make them available for others to use.

The World-Wide-Mind provides two things; an *architecture* and a *protocol*. The architecture defines the types of entities that are required, the protocol defines how these entities interact.

2.1 Architecture

The World-Wide-Mind specification [7] defines three types of entity, a *world service*, a *mind service* and a *client*. A *world service* represents a given problem, such as a micro-world implementation of an artificial creature's habitat (for example Tyrrell's world [14]) or an embodied robot in a real world environment. The function of the world service is to provide the user with a representation of the current state of the world (for example the creature's perceptions in the micro-world, or the robot's perceptions in the real world). The *mind service* represents an implementation of (a part of) a solution to the problem faced by the creature in the world. When provided with the state of the world, the mind service will return a suggested action.

The client software is used to manage the interaction of the mind service and the world service in a *run*. A run consists of the following:

1. Using the client software, the user selects a mind service and a world service.
2. The client software contacts both services to tell them they are involved in a run.
3. The client queries the world service for its state.
4. The client forwards this state to the mind service, requesting it to suggest an action.
5. The client instructs the world service to execute this action.
6. Loop to step 3 until completion of the task at hand.

2.2 Protocol

The protocol of the World-Wide-Mind is named SOML (Society of Mind Markup Language) [12]. This protocol defines how a service is created, and once it is created, how the client (or any other entity) can communicate with it. The services built for the World-Wide-Mind are termed *Lightweight Web Services*, since they are inspired by the emerging web service standards in distributed computing [16], but lack the high level of complexity considered unnecessary for this domain. Since our services are web services, they are interacted with over HTTP, the protocol of the World-Wide-Web. In order to build a service, an author need only have access to some web server software, and have a limited knowledge of web programming using, say, CGI (a simple way of writing online programs that read HTTP messages, instead of console input). The service must support up to five SOML messages, e.g. a mind service must support a `getaction` message.

3 Blocks World

A class of final year undergraduates studying for an honours degree in Computer Science were given the task of building a solution for the well-known *blocks world* problem [15]. The world consists of a set of labelled blocks that are presented in an unordered fashion, and need to be stacked in ascending order.

This problem was delivered to the class using World-Wide-Mind technology. A world service was created that represented a randomly initialised blocks world. This world service presented the state of the world as a text string reflecting the current ordering of the blocks. Each student was then required to write a mind service that would take the state of the blocks world as input and return an action (e.g. move block X to column Y).

Each of the ten minds developed was tested ten times in the following three different worlds:

Test 1: Three blocks, three columns
Test 2: Fifteen blocks, five columns
Test 3: Five blocks, fifteen columns

The performance of the mind is measured as the number of queries (actions) that had to be sent to the world service in order to solve the problem (lower scoring minds are obviously better). A test is passed if the problem is solved.

Table 1: Performance of each mind service in problem world

	Percentage of tests passed				**Average Score**		
Mind	***Test 1***	***Test 2***	***Test 3***	***Total***	***Test 1***	***Test 2***	***Test 3***
I	80%	30%	100%	70%	7	49.33	10.6
II	90%	100%	100%	97%	5.4	57.8	8.6
III	100%	100%	90%	97%	5.8	55.2	8.4
IV	100%	90%	80%	90%	7.75	57.6	10.6
V	90%	100%	100%	97%	5.2	48.6	8.4
VI	70%	100%	100%	90%	8.29	67.4	10.8
VII	10%	0%	40%	17%	0		10
VIII	10%	0%	0%	3%	0		
IX	0%	0%	80%	27%			20.5
X	20%	0%	80%	33%	0		13
Overall	57%	52%	77%	62%	6.05	56.85	11.07

As can be seen, minds performed better when free columns were available (Test 3). Failures to pass tests were due to both coding errors by the students, and communication errors on the network.

4 Blocks World Society of Mind – Hand designed

A Society of Mind, as discussed, is a mind constructed by arbitrating between a number of sub-minds, among which functionality is duplicated and distributed. The developer of the Society of Mind does not need to know anything about the internal operation of the sub-mind, but simply needs to be aware of its interface, in order to use it remotely.

The first Society of Mind developed for the blocks world problem was hand designed to use the best sub-mind. When this mind failed, the second best sub-mind (according to the results in table 1) was chosen and so on. The pseudocode for this simple Society of Mind is shown below:

```
order := { V, III, II, IV, VI, I, X, IX, VII, VIII }
mindIndex = 0
currentMind := order[mindIndex]
state := get state of world
while (problem not solved)
      select action from currentMind
      if (currentMind failed)
            mindIndex = mindIndex + 1 % 10;
            currentMind = order[mindIndex]
      else
            execute action in world and return state
end while
```

This Society of Mind achieved the results shown in Table 2 below.

Table 2: Results for Society of Mind – Hand designed

	Percentage of tests passed				**Average Score**		
Mind	***Test 1***	***Test 2***	***Test 3***	***Total***	***Test 1***	***Test 2***	***Test 3***
SOM 1	100%	100%	100%	100%	5.68	51.2	8.92

97% of the time, the best mind was used. Whenever this failed the Society of Mind was able to use an alternative mind, chosen from the ordered set of minds.

5 Blocks World Society of Mind – Dynamically created

The mind described in section 4 above proved to be robust since it was always able to solve the problem regardless of the failure of one of the sub-minds. However, it would be preferable if we could create a Society of Mind without having to explicitly hand code which minds to call.

In order to do so, a new mind service was created, which would take as parameters the list of sub-minds that should be integrated into the Society on Mind. Once these parameters were received, each one was *automatically* subjected to the tests outlined

in section 3 above. However, this time, for each test that was passed, the following information was saved:

1. Whether there was any free space at the beginning.
2. Whether there were more than three free columns at the beginning.
3. Whether all the blocks were clear at the beginning.

These were identified as three reasons why sub-minds failed to solve the problem. The information above was saved into an `Entry` object, along with the URL of the corresponding mind. Once all the tests were completed, we were left with a collection of `Entry` objects, one for each test that was passed. Obviously, a better sub-mind would have more entries than a weaker one. The mind service saved this information as a *mind configuration* and returned the ID back to the user.

In order to use the mind configuration, the user could select the mind service using the client and pass the ID of the mind configuration as a start-up parameter. Sub-minds were then given control according to the following algorithm:

1. Create a template `Entry` object, to match the current state.
2. Randomly select an `Entry` object that matches the template. The more `Entry` objects associated with a sub-mind, the more likely it is to be selected. (If one cannot be found after 20 attempts, a random `Entry` is picked.)
3. Find the sub-mind that corresponds to that particular `Entry`, and give it control.
4. Return to step 1 whenever one of the following conditions is satisfied:
   - The current sub-mind failed
   - The template for the current state changed, meaning that more/less sub-minds are now eligible for selection e.g. a sub-mind that only passed tests where all blocks were clear is now eligible.
5. Continue until the problem is solved.

In this type of Society of Mind:

1. Better minds will get picked more frequently because they will have more entries.
2. A mind will not be selected when it is unsuited to the current state.
3. A number of different sub-minds are used in each run.
4. New sub-minds can be added dynamically, since tests are performed and saved automatically.

6 Results

The dynamically created Society of Mind achieved the results shown in Table 3 below. As can clearly be seen, this Society of Mind was unable to outperform any of the individual minds, or indeed the hand designed mind. However, once again, it never failed to produce an action, and did not need to be hand designed. The better performing minds had more `Entry` objects (table 4), and consequently were chosen more frequently (table 5).

Table 3: Results for Society of Mind – dynamically created

	Percentage of tests passed				Average Score		
Mind	***Test 1***	***Test 2***	***Test 3***	***Total***	***Test 1***	***Test 2***	***Test 3***
SOM 2	100%	100%	100%	100%	9.84	200.3	13.23

Table 4: Percentage of entries for each sub-mind

Mind	I	II	III	IV	V	VI	VII	VIII	IX	X
Percentage	11.3	15.6	15.6	14.5	15.6	14.5	2.7	0.5	4.3	5.4

Table 5: Percentage of times each sub-mind was called on for Society of Mind

Mind	I	II	III	IV	V	VI	VII	VIII	IX	X
Percentage	16.5	14.0	20.1	16.8	16.8	6.7	1.4	1.4	3.3	3.0

A Society of Mind will only provide a better score to an individual mind if the individual mind is non-optimal, and its deficiencies are catered for in other minds. Clearly, this was not the case here. Several of the minds provided results that could not be improved upon by their combination with other minds. The best option for our Society of Mind would have been to pick one of the better minds e.g Mind V, and stick with it, as we did in the hand designed mind.

What has been demonstrated however, is a mechanism for handling a distributed Society of Mind, composed of independently authored components, which is robust when faced with failure of its components, and has a facility to grow with the addition of new components.

7 Discussion

This paper discusses a simple Society of Mind, developed to solve a simple blocks world problem. This particular problem has faced some criticism [5], but serves our purposes here, as our goal was to show that using independently developed components, it is possible to build robust Societies of Mind. Both of the Societies of Mind built for the purposes of this paper can be considered robust since they will always be able to produce a result, regardless of the strengths or weaknesses of the sub-minds. Each of the sub-minds failed at least one of the tests put to it, either because of coding errors, or communication errors over the Internet.

While accepting that the problem and solution were relatively simple, we expect to be able to scale up. We deliberately kept the specification for the protocol and architecture as simple as possible, and not tied to any specific platform (apart from the web), to ensure that willing authors can easily construct services. We hope to have results for more complex, multi-goal problems, such as Tyrrell's Simulated Environment [14] in the near future. We also intend to show Societies of Mind developed using authors from different institutions.

Building hybrid systems has always been difficult because of the requirement, in practice, that a single research team or single author have the necessary skills to build each of the different sub-systems. This work is a step towards a future where AI/ALife researchers will build hybrid systems out of large numbers of components they did not write.

Acknowledgements

The authors would like to thank two anonymous reviewers for their helpful comments

References

1. Brooks, R. A.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, RA-2:14–23, April (1986)
2. Brooks, R.A.: Intelligence without Reason, Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91) (1991)
3. Bryson, J.: Intelligence by Design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agents, PhD Thesis, Massachusetts Institute of Technology (2001)
4. Connell, J.: SSS: A Hybrid Architecture Applied to Robot Navigation, Proceedings of the IEEE International Conference on Robotics and Automation, Nice, France, pp. 2719-24. (1992)
5. Dreyfus, H.: What Computers Still Can't Do: A Critique of Artificial Reason . Cambridge: MIT Press. (1993)
6. Humphrys, M.: Action Selection methods using Reinforcement Learning, PhD Thesis, University of Cambridge (1997)
7. Humphrys, M.: The World-Wide-Mind: Draft Proposal, Dublin City University, School of Computer Applications, Tech Report CA-0301 (2001) computing.dcu.ie/~humphrys/WWM/
8. Humphrys, M.: Distributing a Mind on the Internet: The World-Wide-Mind, Proc. 6th European Conf. on Artificial Life (ECAL-01), (2001).
9. Isla, D., Burke, R., Downie, M., and Blumberg, B.: A Layered Brain Architecture for Synthetic Creatures. Presented at The International Joint Conference on Artificial Intelligence, Seattle, USA, (2001).
10. Maes. P.: How to do the right thing. A.I. Memo 1180, MIT, Cambridge, MA, December, (1989)
11. Minsky, M.: The Society of Mind, Simon and Schuster (1985)
12. O'Leary, C., SOML (0.9) Society of Mind Markup Language 0.9, Available online at w2mind.comp.dit.ie
13. Skinner, J. and Luger, G.: Contributions of CBR to an Integrated Reasoning System, Journal of Intelligent Systems, 5:1, 19-47 (1995)
14. Tyrrell, T., Computational Mechanisms for Action Selection, PhD Thesis, University of Edinburgh, Centre for Cognitive Science (1993)
15. Winograd, T.: Understanding Natural Language. Academic Press. New York. (1972)
16. World-Wide-Web Consortium: Web Services Activity, online at w3.org/2002/ws/