ACHIEVING ADAPTABLE BEHAVIOUR IN INTELLIGENT ROOMS USING CURIOUS SUPERVISED LEARNING AGENTS

KATHRYN MERRICK, MARY LOU MAHER, ROB SAUNDERS
Key Centre of Design Computing and Cognition, University of Sydney
[kathryn|mary|rob]@arch.usyd.edu.au

Abstract. Multiple devices, both hardware and software, may come and go at any time in a given room. Software controlling the behaviour of these devices must be able to adapt to encompass new devices or the removal of existing devices. This paper presents a model for curious, supervised learning agents that address the issue of adaptability at a behavioural level in an intelligent room. Curious, supervised learning agents comprise a curiosity module and a supervised learning algorithm. The curiosity module identifies interesting devices on which to focus the agent's learning. The supervised learning component realises behaviours by observing, modelling and mimicking human actions. Our framework is demonstrated in a virtual meeting room in *Second Life.* We show that the curious learning agent can adapt its behaviour to identify new learning goals in response to new devices and activities.

Keywords. Curiosity, Supervised learning, Agent, Intelligent room.

1. Introduction

Adaptability is recognised as a key concern when developing intelligent environments. In his vision for the computer in the twenty-first century, Weiser (1991) describes how multiple devices, both hardware and software, may come and go at any time in a given room. Computational processes controlling these devices must be able to adapt their behaviour to encompass new devices or the removal of existing devices. This paper presents a model for curious, supervised learning (CSL) agents that address the issue of adaptability at a behavioural level.

Previously, system architectures for intelligent rooms have had two main levels of focus: middleware architectures and behavioural architectures. Middleware architectures have considered adaptability in terms of the need for reconfigurable environments at a hardware level. These approaches support resource management, communication between devices and dynamic reconfiguration. Adaptability is facilitated via components such as real-time interaction modules (Brooks et al., 1997), Metaglue modules that support modification of a running system (Coen et al., 1999; Phillips, 1999), Gaia presence services (Roman et al., 2002) and ad hoc networking (BLIP Systems, 2007). However, while this level of adaptability is necessary for achieving adaptive behaviour in intelligent environments, it is not sufficient alone. Adaptability at the behavioural level is required for the actions and responses of an intelligent room to react to the addition or removal of devices, and the changing behaviour of people in the room.

Existing approaches to behavioural architectures for intelligent rooms have demonstrated success as techniques for allowing intelligent environments to adapt to changing human activities (Brdiczka et al., 2005; Mozer, 1998). In contrast, the issue of achieving autonomous, adaptive behaviour when the hardware devices in the room change, has not been widely considered. Previous behavioural architectures have tended to assume a fixed set of devices to be monitored or controlled. Reprogramming or addition of applications or agents is required to respond to new devices at the behavioural level. Responding to new devices without reprogramming or otherwise modifying behavioural architectures requires new approaches that can explore the potential of a new system configuration and identify appropriate new behaviours.

This paper presents a CSL agent model as a behavioural architecture that can adapt the responses of an intelligent room to the addition or removal of devices from the space and changes in the activities in the room. This model is designed to draw on the potential of adaptive middleware technologies – such as ad hoc networking or Metaglue – but extend those technologies with an adaptive behavioural level.

Our CSL agent is demonstrated in a virtual meeting room in *Second Life,* which is modelled on a real world university meeting room. We show that the curious learning agent can identify new learning goals in response to new devices and human actions and use these goals to focus the agent to learn new behaviours.

2. A Behavioural Architecture using Curiosity and Supervised Learning

CSL agents comprise a curiosity module and a supervised learning algorithm, as shown in Figure 1. The role of the curiosity module is twofold. First, the curiosity module identifies interesting tasks on which to focus the agent's learning. Secondly, the curiosity module determines when learned tasks should be acted upon. The curiosity module functions as a filter to focus attention on relevant data.

Relevant data selected by the curiosity module is input to the learning and activation processes of the supervised learning algorithm. The learning process constructs behaviours as sequences of actions by observing, modelling and mimicking curious human actions. The activation process executes these behaviours in the environment.

2.1. MODELLING CURIOUS SUPERVISED LEARNING

Standard supervised learning is defined by a set **S** of states describing the agent's environment and a set **A** of actions that the agent can perform. A task to be learned is defined by a set **X** of examples. The agent learns a policy • mapping states to actions to perform the task. This formulation assumes a set of examples for a small number of tasks (often just one task). In intelligent environments, states describe the status of devices, such as whether a light is on or the current temperature. Actions are ways that this state can be changed. However, examples typically represent multiple tasks as multiple humans use multiple different devices at different times. In CSL, the curiosity module acts as a filter for states and examples, to focus learning and action on certain tasks. We denote the set of all states, actions and examples experienced by the agent as the experience trajectory **Y**:

$$\mathbf{Y}_{(t)} = \{S_{(1)}, X_{(1)}, A_{(1)}, S_{(2)}, X_{(2)}, A_{(2)}, \ldots S_{(t)}, X_{(t)}, A_{(t)}, \quad \} \tag{1}$$

Curiosity is modelled as a function of the agent's experiences at time t:

$$C(\mathbf{Y}_{(t)}) = \begin{cases} \{S_{(t)}\} \text{ to motivate action} \\ \{X_{(t)}\} \text{ to motivate learning} \\ \{S_{(t)}, X_{(t)}\} \text{ to motivate learning and action} \\ \{\} \text{ otherwise} \end{cases} \tag{2}$$

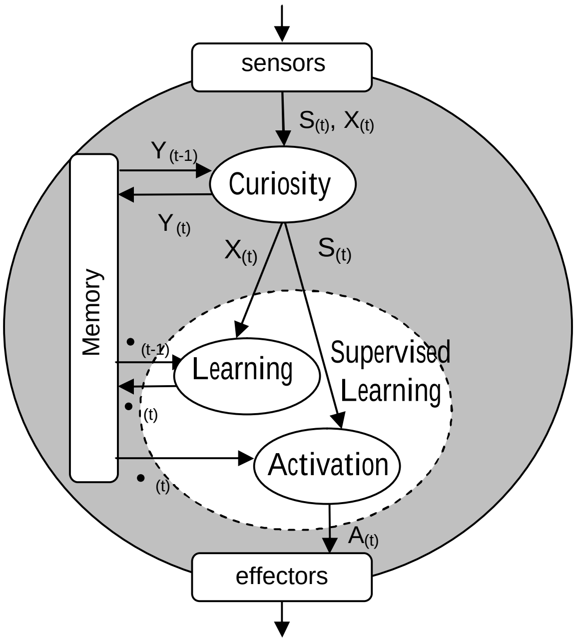

Figure 1. Model for a curious supervised learning agent.

2.2. META-SENSORS AND META-EFFECTORS

Maher et al. (2006) proposed the use of context-free grammars (CFGs) to model dynamic state and action spaces in intelligent environments. Here we describe two specific CFGs for representing the states and actions of an intelligent room to an agent in terms of the devices D_i that comprise its sensors and effectors:

```
S(t)              à   <sensations>
<sensations>      à   <DiSensations><sensations>|•
<DiSensations>    à   <sj><DiSensations> | •
<sj>              à   <number> | <string>
<number>          à   ...
<string>          à   ...
```

(3)

```
A(t)            à   <actions>
<actions>       à   <D_iActions><actions> | •
<D_iActions>    à   <A_j><D_iActions> | •
<A_j>           à   ...                          (4)
```

We assume that this adaptable CFG representation of the changing state and action spaces of the agent is constructed by middleware resource management tools. The agent itself uses meta-sensors, or ‘sensors of sensors’ to monitor the changing state and action space. Likewise, the agent uses meta-effectors to communicate action commands to the middleware layer. In this way, the agent has a fixed set of sensors and effectors but can monitor and affect a dynamic set of devices.

3. Experiments in an Adaptive Virtual Room

We experimented with CSL agents in a virtual meeting room in *Second Life* (www.secondlife.com: Oct 2007), shown in Figure 2. This room is modelled on a real world meeting room in a university. The physical room can be used for seminars, video conferences, staff meetings or project work by students. The virtual room is used for tutorial exercises and research.

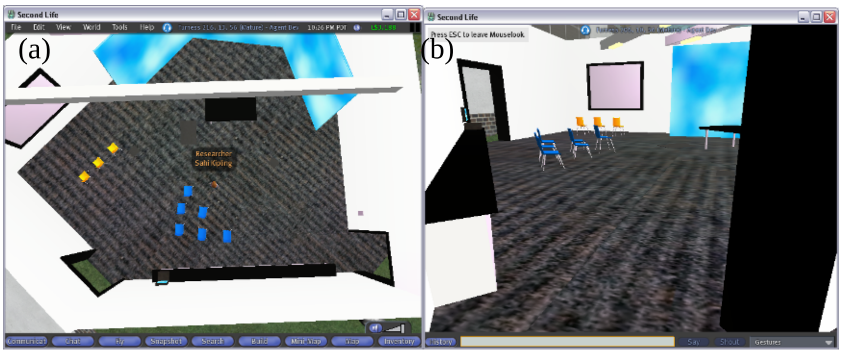

Figure 2. A virtual meeting room in *Second Life.* (a) From above. (b) From the main entrance. Sensors monitor lights, avatar presence, seats and the SmartBoard. Effectors can modify lights and the SmartBoard and its applications.

3.1. SENSORS AND EFFECTORS

A number of working devices have been programmed for the virtual meeting room using the Linden Scripting Languages, including a virtual ad hoc network based on the idea of a BLIP System (2007), lighting, floor sensors, a virtual SMART Board (http://www2.smarttech.com/st/en-US/Products/SMART+Boards/: Oct 2007) and smart chairs.

The virtual BLIP System comprises a BLIP Server (on a virtual PC) and two BLIP Nodes. BLIP Nodes detect other scripted devices in the room and communicate their *Second Life* identifier (UUID) to the BLIP Server. The BLIP Server maintains a list of all devices in the room. The BLIP Server PC also hosts the CSL agent. In our experiments, the virtual BLIP System acts as an adaptive middleware layer. We assume some custom BLIP application software to create and communicate the CFG representation.

The virtual room also has a lighting system comprising one switch that activates seven ceiling lights. The switch and lights, once detected by a

BLIP Node, can communicate with the BLIP Server. The switch communicates an example of human behaviour each time it is touched. The lights communicate their current state each time it changes to on or off.

A single floor sensor, located in the middle of the virtual room, has a 9 metre radius for detecting avatars. This extends to the walls of the room but leaves un-monitored areas near the doors. This allows the sensor to monitor the transition as an avatar enters the room. The sensor communicates the names of detected avatars to the BLIP Server each second.

A virtual SMART Board simulates running Skype for video conferencing. The SMART Board turns on and activates Skype when it is touched and sends its state and an example to the BLIP Server. Three smart chairs are arranged in front of the SMART Board. The smart chairs can sense when an avatar sits on them, or stands up and communicate this state to the BLIP Server[1].

In CFG form, an example state of the room when the lights are on is:

$$<1_{light1}><1_{light2}><1_{light3}><1_{light4}><1_{light5}><1_{light6}><1_{light7}>$$

If the avatar, Sahi Kipling, enters the room, this becomes:

$$<1_{light1}><1_{light2}><1_{light3}><1_{light4}><1_{light5}><1_{light6}><1_{light7}><1_{SahiKipling}>$$

3.2. THE CURIOUS ROOM AGENT

For the experiments in this paper, we used a simplified version of Stanley's (1976) model of habituation to represent curiosity for learning tasks, a model of competence based on learning error for curiosity towards acting, and table-based supervised learning using associations (Steels, 1996) for the learning module. Table 1 shows the parameters of these models and the values used in our experiments.

TABLE 1. Agent parameters and their values.

Parameter	Value	Description
HABITUATION_RATE	0.5	Rate of novelty decrease
NOVELTY_THRESHOLD	0.5	Threshold for learning
COMPETENCE_THRESHOLD	0.8	Threshold for action
LEARNING_RATE	0.5	Rate of supervised learning

3.3. RESULTS AND DISCUSSION

Three preliminary experiments were conducted as a proof-of-concept for CSL agents. The results are discussed here, along with the implications for further development of the system.

The first experiment, illustrated in Figure 3, demonstrates the ability of a CSL agent to learn a simple behaviour. An avatar, Sahi Kipling, walks into the room and turns on the light. Sahi then walks to the door, turns off the light and walks out of the room. Sahi repeats this several times. After six repetitions, Sahi walks into the room and turns towards the light switch. However, before she presses it, the CSL agent turns on the light. Likewise, when Sahi walks to the door to leave the room the agent turns the light off.

---

1 The idea of a smart chair is easily implemented in *Second Life* using built in functions to detect 'sit actions' by avatars. In the real world, a smart chair could have a pressure sensor in the seat, or be implemented using cameras to detect when people are seated in the chairs.

In this experiment it required six repetitions of the action for the agent to learn the behaviour. This could be decreased (or increased) by modifying the LEARNING_RATE or NOVELTY_THRESHOLD parameters. The advantage of a higher learning rate is that fewer repetitions are required for the agent to learn a behaviour. However, learning is less tolerant to mistakes or random actions performed by avatars. Likewise, higher novelty threshold would mean that examples become curious more quickly. However this would mean that the agent may try to learn about random occurrences, when there is very little to be learned in such scenarios.

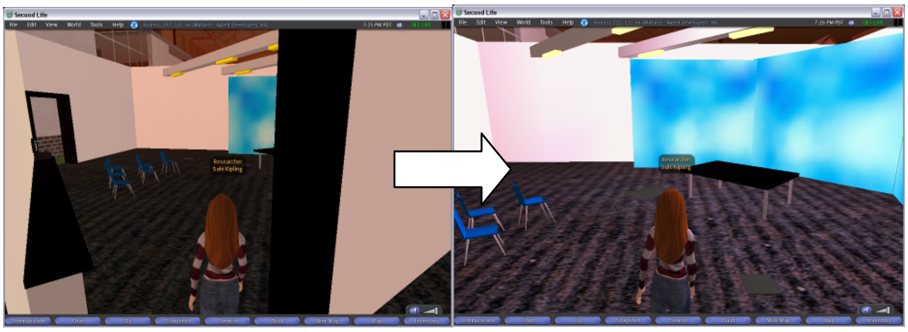

Figure 3. The curious supervised learning agent learns to turn on the lights when an avatar enters the room.

In the second experiment, Sahi brings a SMART Board and three smart chairs into the virtual room. They are detected by the BLIP System and the new state and action space communicated to the CSL agent in a CFG representation. Sahi then sits down, and turns on the SMART Board, which launches Skype as shown in Figure 4. After some time, she gets up and turns off the SMART Board. She repeats this sequence several times. Again, after six trials, the agent is able to predict Sahi's needs and turn on the SMART Board when she sits down. Likewise, the agent turns the board off when Sahi stands up. This experiment demonstrates the ability of the CSL, in combination with an adaptive middleware layer, to adapt the behaviour of the room to a new device and new human activities.

One limitation to emerge from this example, is that, because the introduction of the SMART Board has changed the state space of the room, the agent will no longer turn the lights on and off automatically. If the SMART Board were removed, the light switching behaviour would be remembered, but if the light switching behaviour is required while the SMART Board is in the room, the agent must be taught the sequence again. This limitation is apparent because we use a table-based learner in this prototype. This means that every state is treated as unique and an action must be learned for that state. The agent is very accurate in learning but unable to generalise behaviours learned in one state to other similar states. A neural network supervised learner may be more appropriate for this.

A similar issue is that the agent will only respond by turning on the lights when Sahi Kipling enters the room. If another avatar enters the room and wants the lights on, they will have to teach the agent that behaviour themselves. If the agent could generalise, then it could apply the same behaviour for Sahi to other avatars. However, the disadvantage of this is that the other user may want different behaviours. There is a trade off between the ability to generalise behaviours and the ability to recognise different contexts.

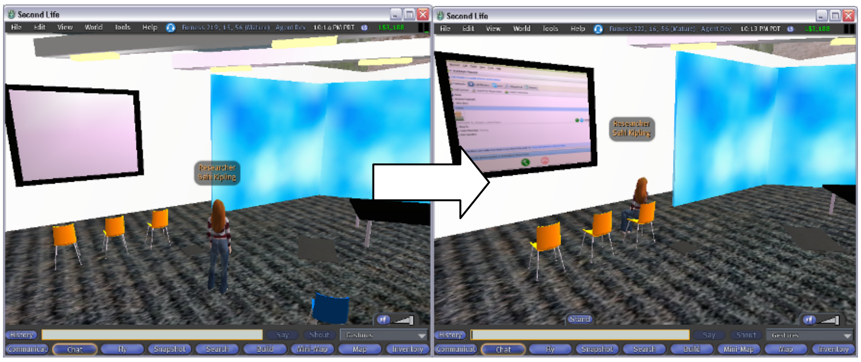

Figure 4. After a SMART Board is brought into to the room, the same curious supervised learning agent learns to turn on the SMART Board and launch Skype when an avatar sits on the smart chairs in front of the board.

The third experiment shows that the agent can adapt longer action sequences. No changes are made to the devices in the room or the CSL agent, following the previous experiment. Sahi sits down and the CSL agent responds by turning on the smart board. Sahi then turns off the lights[2]. After some time, Sahi stands up, turns off the SMART Board and turns the lights back on. She repeats this several times. After six repetitions, when Sahi sits down, the curious agent responds by, not only turning on the SMART Board, but also turning off the lights. This experiment illustrates two properties of the CSL agent. First, the agent can learn sequences of actions to form a behaviour. Secondly, the agent can adapt a previously learned sequence to changes in human activities.

An issue arises from this example if Sahi no longer wants the lights off when using the SMART Board. When she turns the board on, the agent responds by turning the lights off. So Sahi has to turn the lights on again. Eventually the agent will learn this sequence, but it will be inefficient – it will first turn the lights off then turn them on again. One possible solution is a user interface by which users can manually edit, override or delete learned behaviours.

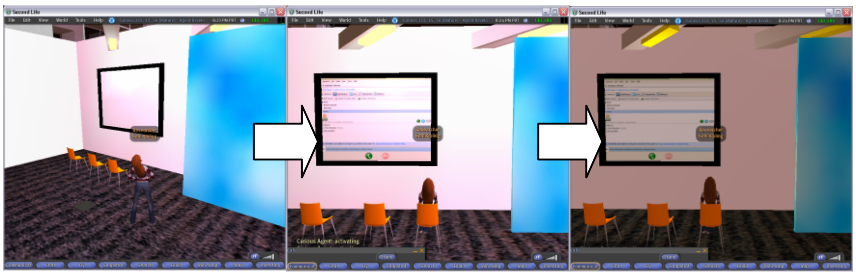

Figure 5. The same curious supervised learning agent adapts its previously learned behaviour and learns to turn on the SMART Board, launch Skype and turn off the lights when an avatar sits down in front of the board.

[2] One useful property of virtual rooms is that switches can be reached from a distance. In a physical room this action sequence would have to be performed in a different order or using a remote control.

4. Conclusions and Future Work

This paper has presented a model of curious supervised learning (CSL) agents for adaptive behaviour in an intelligent virtual room. Experiments demonstrate the ability of agents using this model to adapt to human activities and new devices in the room. While the experiments presented in this paper represent a proof-of-concept for the CSL agent model, further work is required to develop these models. First, we hope to develop CSL agents with more complex curiosity modules to enable more effective attention focus and goal finding behaviour. We will also develop a suite of more significant experimental scenarios to better test the range of functionality of CSL agents. This test suite will include environments with greater numbers of potential learning tasks, dynamic environments and environments with more complex learning tasks. The ultimate aim of this testing is to develop CSL agent technologies from the virtual world to real world scenarios.

Acknowledgements

This work was supported by a grant from the Australian Research Council for the project: "Curious Places: Agent Mediated, Self-Aware Worlds". This paper was written while Mary Lou Maher was working at the National Science Foundation in the United States. Any opinion, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

BLIP Systems: 2007, BLIPNet Technical White Paper, BLIP Systems A/S, Aalborg, Denmark.

Brdiczka, O., Reignier, P. and Crowley, J.: 2005, Supervised learning of an abstract context model for an intelligent environment, *Joint sOc-EUSAI Conference*, Grenoble, pp. 259-264.

Brooks, R. A., Coen, M., Dang, D., DeBonet, J., Kramer, J., Lozano-Perez, T., Mellor, J., Pook, P., Stauffer, C., Stein, L., Torrance, M., Wessler, M.: 1997, The intelligent room project. *The Second International Cognitive Technology Conference*, Aizu, pp. 271-279.

Coen, M., Phillips, B., Warshawsky, N., Weisman, L., Peters, S., Finin, P.: 1999, Meeting the computational needs of intelligent environments: the Metaglue system, *MANSE*.

Maher, M-L., Merrick, K., Macindoe, O.: 2006, Intrinsically motivated intelligent sensed environments. *The Thirteenth International Workshop of the European Group for Intelligent Computing in Engineering*, Ascona, Switzerland, pp 455-475.

Mozer, M.: 1998, The neural network house: an environment that adapts to its inhabitants. *AAAI Spring Symposium on Intelligent Environments*, pp 110-114.

Phillips, B.: 1999, Metaglue: a programming language for multi-agent systems, Masters Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Roman, M., Hess, C., Cerqueira, R., Ranganat, A., Campbell, R., Nahrstedt, K.: Gaia: a middleware infrastructure to enable active spaces. *IEEE Pervasive Computing*, (Oct-Dec 2002), pp 74-83

Stanley, J. C.: 1976, Computer simulation of a model of habituation. *Nature*, **261**: 146-148.

Steels, L.: 1996, Self-organising vocabularies, Technical Report, Artificial Intelligence Laboratory, Vrije Universiteit, Brussels.

Weiser, M.: 1991, The computer for the 21st century. *Scientific America*, **265**(3): 94-104