Agent Models for Dynamic 3D Virtual Worlds

Mary-Lou Maher
National ICT Australia, Key Centre of Design Computing and Cognition, University of Sydney
mary@arch.usyd.edu..au

Kathryn Merrick
National ICT Australia, School of Information Technologies, University of Sydney
kkas0686@it.usyd.edu.au

Abstract

Agents are systems capable of perceiving their environment through sensors, reasoning about their sensory input using some characteristic reasoning process and acting in their environment using effectors. When one or more agents control the objects that comprise a 3D virtual world, the result is a dynamic, adaptive environment that changes in response to users' actions. We have experimented with three different agent models for this purpose: a swarm model, a cognitive model and a motivated agent model. Each of these models differs in the complexity of its implementation and can thus be used to produce dynamic virtual environments of differing behavioural complexity. This paper introduces a schema for characterising the implementation and behavioural complexity of agent models for dynamic virtual environments. We apply this schema to the agent models we have studied to reveal their advantages and disadvantages and identify directions for future work.

1. Introduction

A sense of place can be achieved in virtual environments by the use of rooms, buildings and other artefacts associated with physical places. When the virtual environment is networked and multi-user, these places can be used to support a broad range of activities including communication, collaboration and education.

The focus of many virtual world design platforms has been on the visual and interactive aspects of the world, resulting in environments that are largely static. Dynamic behaviour of the 3D objects in virtual worlds is typically achieved with simple, scripted behaviours triggered by events. The effort required to implement these simple behaviours over wide areas of a virtual world is high as each object must be scripted individually. A major issue in the development of virtual worlds is the behavioural complexity of the objects in the world and the complexity of the implementation that is required to achieve the behaviours.

Maher and Gero [3] proposed a way to increase the behavioural complexity of dynamic virtual worlds by giving agency to each persistent 3D object in the virtual world. Agents are systems capable of perceiving their environment through sensors, reasoning about their sensory input using some characteristic reasoning process and acting in the world using their effectors. Agents are generally implemented using a programming language such as Java or C/C++ providing them with the potential for complex reasoning processes that produce dynamic virtual environments of greater behavioural complexity than those with scripted behaviours. Maher and Gero achieved an increase in behavioural complexity by using a cognitive agent model [3, 4, 6] for the behaviour of the 3D objects in a virtual meeting room. Their cognitive model includes processes for reasoning about the world at different levels of abstraction and a direct communication structure. It requires preprogrammed domain specific rules giving it a high implementation complexity.

This paper presents two additional models that can be used to create dynamic virtual worlds: a swarm model and a motivated agent model [2]. The swarm model is an agent based alternative to scripted behaviours that reduces implementation complexity. The motivated agent model also reduces implementation complexity but offers the behavioural complexity of the cognitive model.

Figure 1 illustrates our schema for characterising the implementation and behavioural complexity of agent models for dynamic virtual environments and shows broadly where each model fits into the schema. In the following sections of this paper we define the terms behavioural and implementation complexity in more detail then use them to evaluate each model. These discussions provide the basis for a more detailed classification of each model later in the paper. We demonstrate the performance of each model when implemented in a virtual meeting room scenario. We consider the advantages and disadvantages of each approach as well as possible future directions for agent models in 3D virtual worlds.

2. Behavioural Complexity

Behavioural complexity measures the richness of the reasoning process that produces the dynamics of a virtual environment. Behavioural complexity can be measured in five

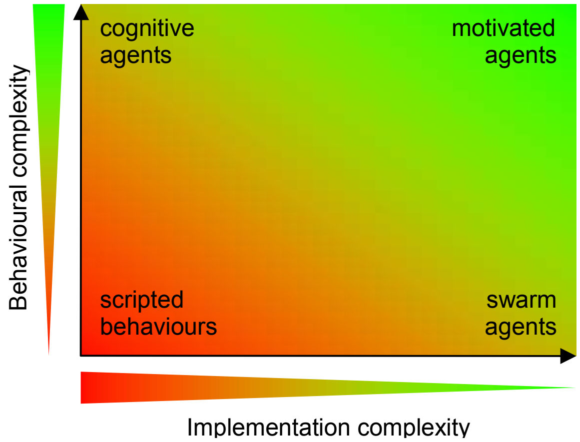

Figure 1. Broad classification schema for techniques used to create dynamic 3D virtual worlds.

modes: reflexive, reactive, reflective [3], autonomous [7] and proactive [8]. Each mode requires increasingly sophisticated reasoning, with reflexive being the simplest.

Reflexive behaviour is a pre-programmed response to the state of the environment – a reflex without reasoning. Only recognised states will produce a response. This mode of behaviour is typically achieved using a scripting language to implement behaviours associated with the 3D objects in a virtual world. These scripts define behaviours which are triggered by predefined patterns of events.

Reactive behaviour manifests itself as reasoning about responses within a fixed set of goals. This mode of behaviour is achieved by using agents to control one or more objects in a virtual world. Agents make changes to the objects in the world to work towards achieving goals in response to changes in the state of the environment. Reactive behaviour is a consequence not only of the state of the environment but also of how that state is perceived by each agent. Perception may vary as a consequence of experience.

Reflective behaviour also has a fixed set of goals. In addition, it does not simply react but hypothesises possible desired states of the environment and proposes alternate actions that will achieve those states. This type of behaviour is also achieved using agents. Reflective agents are able to reason about the world at different levels of abstraction.

Autonomous behaviour does not simply select goals from a fixed set but includes reasoning processes to create new goals in response to new situations.

Pro-active behaviour goes beyond reasoning about goals to be achieved and hypothesises possible undesirable future states of the environment and proposes alternate actions to avoid those states.

3. Implementation Complexity

Implementation complexity is a measure of the level of programming effort that is required to produce a certain level of behavioural complexity. Implementation complexity depends on properties of the system architecture such as domain dependence and social ability.

3.1. Domain Dependence

Domain specific systems incorporate modules that tie them to a particular environment or problem within an environment. Examples of such modules are scripts defining the behaviour of a specific object or a set of goals relevant to a particular environment.

Domain independent systems are comprised of general modules that are relevant to a wide range of environments and problems. Domain independent systems have less implementation complexity than domain specific systems because domain specific systems must be modified each time they are to be applied to a new problem or environment.

3.2. Social Ability

When the objects in a 3D virtual world are controlled by more than one script or agent there may be a requirement for these scripts or agents to communicate. Implementation complexity is affected by the level of communication or social ability of the system, with no communication being the simplest.

No communication implies that every script or agent functions independently of every other script or agent and does not receive any information about their behaviour or the region of the environment they modify.

Indirect communication or *stigmergy* is communication through the environment. There are two types of stigmergy, discrete stigmergy and continuous stigmergy. When a script or agent responds to a structural change in the environment made by another script or agent it is responding to discrete stigmergy. Scripts or agents can communicate via continuous stigmergy by depositing pheromones. In nature a pheromone is a chemical substance deposited by one individual that triggers some behaviour in another individual. In a virtual world, a pheromone can be modelled as an invisible 3D object with a type and strength. The type of pheromone determines the behaviour that is triggered. The strength of the pheromone determines the time it will take to decay and thus the amount of time it will be present to trigger behaviour from other scripts or agents.

Direct communication implies that messages are sent directly from one script or agent to another script or agent. This form of communication has the greatest implementation complexity. Smith et al [6] define an agent society as an aggregation of agents that share a common connection with a virtual world and have some ontological connection with each other. For example, a *Floor* agent and a set of *Wall* agents might form a society that represents a virtual meeting room. When a group of agents form a society there is frequently a need for them to communicate. Direct

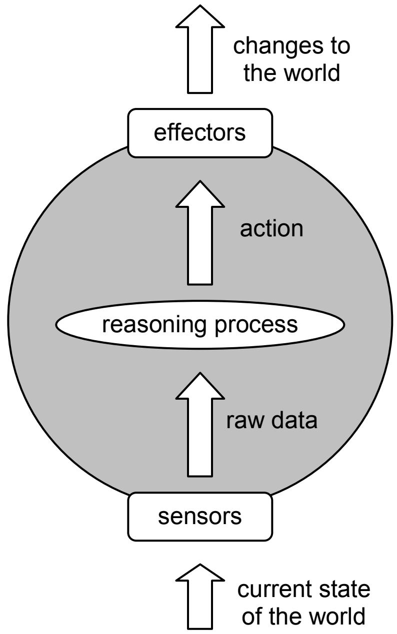

Figure 2. The general agent model.

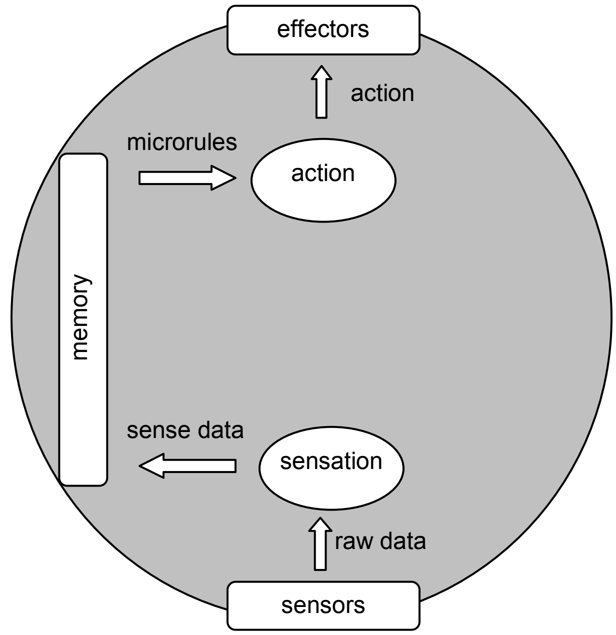

Figure 3. The swarm agent model.

communication allows agents to interact with other agents by sending messages from the effector of one agent to the sensor of another. This allows agents within a society to self-organise without flooding the virtual world with additional objects or events.

4. Agent Models

An agent is a system that perceives its environment through sensors, reasons about its sensory input using some characteristic reasoning process and acts upon the environment through effectors. This general model is shown in Figure 2. Agent models describe ways to implement the characteristic reasoning process of one or more agents. Agents using different models may differ in their behavioural or implementation complexity and thus result in dynamic 3D virtual worlds of differing complexity.

4.1. A Swarm Agent Model

Swarm intelligence is the property of a system whereby the collective behaviours of unsophisticated agents interacting locally with their environment cause coherent functional global patterns to emerge [5]. We investigated swarm intelligence as the basis for an agent model in a virtual world with the purpose of determining whether it is possible for the objects in 3D virtual worlds to achieve globally coherent behaviour without the complexity of a structured communication protocol and multiple levels of reasoning.

In our swarm based model, shown in Figure 3, each agent has two internal reasoning sub-processes, sensation and action. These processes are facilitated by three structures, sensors, memory and effectors. Sensors sense the local state of the environment, that is, the environment within some small radius of the agent. This raw data is transformed by the sensation process into sense-data structures more appropriate for reasoning. Sense-data structures incorporate both the most recent raw data and the raw data sensed immediately prior to the most recent data. This allows a swarm agent to reason about changes in its environment as well as static states. The action process uses a reflexive mode to respond to sense-data by triggering effectors to make changes to the environment. Behaviours are selected by consulting a pre-programmed lookup-table of domain specific microrules that is stored in the agent's memory. Microrules are boolean conditions about sense-data. When events in a virtual environment cause a microrule to evaluate to true a behaviour is triggered.

We implemented a suite of microrules for reproduction, growth, clustering, stigmergic interaction and death. We illustrate the performance of swarm agents using these rules with the example of a virtual meeting room implemented in the SecondLife (www.secondlife.com) virtual environment. In this implementation, agents function as members of a society in which each agent controls exactly one object in the 3D virtual environment. There are two *Floor* agents, four *Wall* agents, four *Column* agents, four *Beam* agents, two *Roof* agents, four *Chair* agents and a *Table* agent as shown in Figure 4.

Each agent can sense other agents and avatars that fall within a radius one and a half times the size of the radius of the object that it controls. Suppose that a small meeting room has reached its capacity of three avatars. One of the *Wall* agents senses the presence of three avatars and constructs sense-data corresponding to "the number of people

Figure 4. A virtual meeting room in SecondLife.

Figure 5. Global behaviour of the swarm the number of avatars present increases: *Floor, Beam* and *Roof* agents expand and *Chair* agents multiply.

in the room is three". The agent then consults its list of microrules and identifies that the current sense-data makes true the condition on the "grow when crowded" microrule. The rule is fired and the *Wall* agent expands. Following this expansion, one of the *Beam* agents senses the presence of the newly enlarged *Wall* agent and constructs the sense-data corresponding to "*Wall* has a new scale of 1.5". The agent then consults its list of microrules and identifies that the current sense-data makes true the condition on the

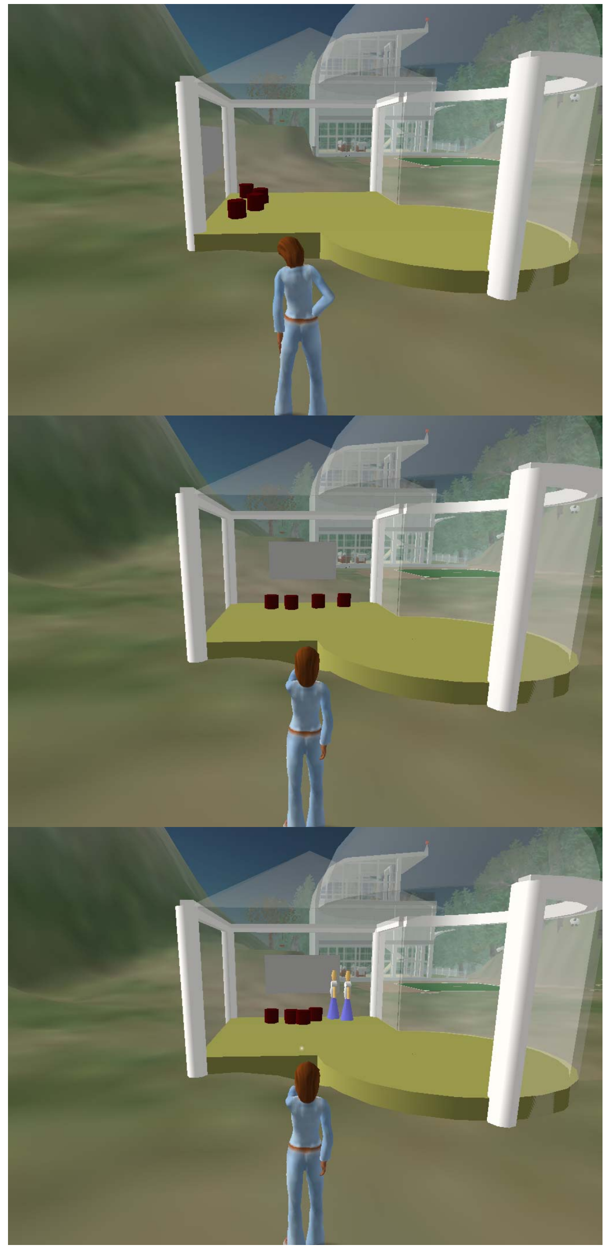

Figure 6. A swarm can adapt to different situations without specific programming.

"grow when neighbour grows" microrule. The rule is fired and the *Beam* agent expands. This expansion is gradually communicated to all agents in the society via this process of discrete stigmergy until the entire meeting room has adjusted to the avatars' presence in the room either by expanding (*Wall* agents) or multiplying (*Chair* agents) as shown in Figure 5.

While individual swarm agents display only simple, reflexive behaviour, the swarm as a whole is able to adapt to situations not pre-programmed in microrules. For example, the clustering microrules in our implementation are general enough to produce the three situations in Figure 6 where

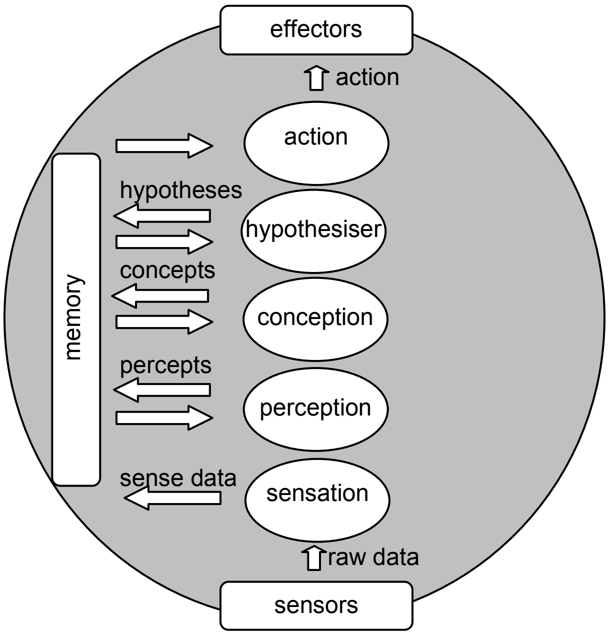

Figure 7. The cognitive agent model.

chairs have adapted to different arrangements of the projector screen, even when there are avatars in the way.

4.2. A Cognitive Agent Model

Maher and Gero's original agent model [3, 4, 6] is shown in Figure 7. Agents using this model are capable of constructing increasingly complex interpretations of their environment. Their interpretations influence their selection of goals and actions. Interpretations are constructed using five successive reasoning sub-processes, sensation, perception, conception, hypothesising and action. These processes are facilitated by three structures, sensors, memory and effectors.

Like swarm agents, cognitive agents function as members of a society in which each agent controls exactly one 3D object from the virtual world. Unlike swarm agents, cognitive agents sense the global state of their environment, that is, all agents and avatars that fall within the bounds of the society. Sensation transforms raw data from sensors into sense-data structures more appropriate for reasoning. The perception, conception and hypothesising processes each build upon the output of the former to build more abstract interpretations of the environment. Perception transforms sense-data into patterns called percepts that are used as the building blocks for concepts about recurring situations. Conception is the process of recognising concepts and is the basis for hypothesising desired situations. The hypothesiser identifies mismatches between the current and desired situations and reasons about which goals should be pursued to eliminate or reduce the mismatch.

The action process reasons about sense-data, percepts, concepts and goals and selects a behavioural mode corresponding to the complexity of the current interpretation. Reflexive mode responds to sense-data with pre-programmed actions. Reactive mode reasons about both sense-data and percepts to produce actions. Reflective mode uses the concepts understood by the agent to hypothesise possible external states and propose alternate goals to achieve them.

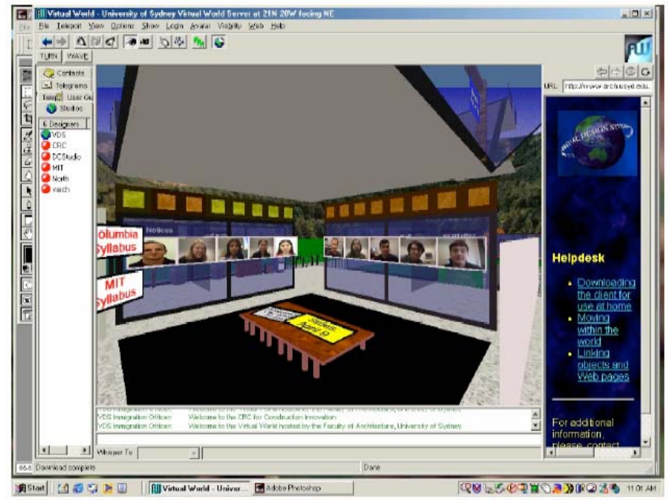

Figure 8. A virtual meeting room in Active Worlds. (Picture from Smith et al [6]).

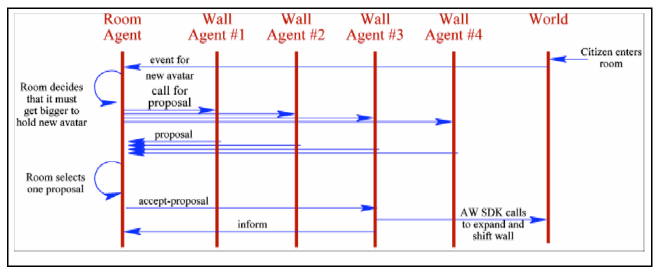

Figure 9. Interaction diagram showing communication within the virtual meeting room society when a new avatar enters a full room. (Picture from Smith et al [6]).

Communication within the cognitive agent society uses the Contract Net negotiation protocol, a form of direct communication, in contrast to the indirect communication used by swarm agents. Agents with problems to solve broadcast a call for solution proposals to all agents in their society. Agents that believe they can satisfy the call answer with proposals. One agent is then awarded a contract to initiate their proposal.

We illustrate the performance of this agent model using the example of a virtual meeting room implemented in the Active Worlds (www.activeworlds.com) virtual environment. This meeting room consists of a *Room* agent plus a set of *Wall* agents as shown in Figure 8. The *Room* agent has a sensor that can sense the presence and location of avatars. Suppose that the meeting room has reached its capacity of 20 avatars when Greg enters. The room's sensor constructs sense-data corresponding to "Greg is in the room" and the perception process interprets this with the percept "the number of people in the room is 21". The conception process uses forward chaining on percepts and expectations to recognise that the room is too small. The hypothesiser then identifies two goals that can reduce the mismatch between the current state and the desired state of the world from a fixed set of domain specific goals, such as "make the room bigger" and "eject one citizen". It selects the goal to make the room bigger. The action process recognises that it does not have an effector that can make the room bigger so it sends a message to the society to call for proposals to achieve the "make the room bigger" goal. An interaction diagram that illustrates how this inter-agent communication propagates through the society is shown in Figure 9.

While the internal processes and communication requirements of a swarm agent are significantly simpler than that of a cognitive agent, both cognitive agents and swarm agents incorporate domain specific components. Cognitive agents incorporate rules with different levels of abstraction while swarm agents incorporate microrules at a single level of abstraction. However, cognitive agents are more easily applied to new problems by defining a new set of goals. Swarm agents require the definition of carefully chosen microrules to achieve the same behaviour.

4.3. A Motivated Agent Model

A motivated agent model has the potential to achieve behavioural complexity without the need for domain specific rules. Motivation is that which gives purpose and direction to behaviour and motivation is the drive that arouses an organism to action towards a desired goal [1]. Purpose is simply a synonym for goal so we can say that motivation is that which creates goals and that which stimulates action towards goals, the two requirements for autonomy. The motivated agent model [2] is an approach to building autonomous agents that are influenced by a domain independent motivation process rather than pre-programmed domain specific goals or microrules. In place of such influences, these agents are motivated to generate their own goals by identifying interesting events in their environment. In addition, they are motivated to solve their goals and encapsulate the knowledge acquired while solving goals as new behaviours.

The primary reasoning components of this model, shown in Figure 10, are sensation, motivation, learning and action. Sensation transforms raw input from sensors into sense-data and event structures more appropriate for reasoning. Sense-data structures represent static states, while events represent the difference between two states. These structures allow motivated agents to reason about both the changes in their environment and the static states they sense. The motivation process identifies interesting sense-data and events using the domain independent intuition that rare occurrences are interesting. Interesting occurrences are used as the basis for new goals. The motivation process priorities goals so that the agent can direct its action towards the most important goal. Learning encapsulates knowledge learned while solving goals as new effector sequences that can be reused if similar goals are identified in the future. Action chooses effectors to fire to further the agent's progress towards its most important goal. Motivated agents use a reflective mode of reasoning to create goals. They use a reactive mode of reasoning to respond to sense-data and events using reinforcement learning to choose actions that further progress towards their highest priority goal. Unlike agents that use the previous two models, agents that use this motivated model do not function as members of a society. Instead, a single agent controls a

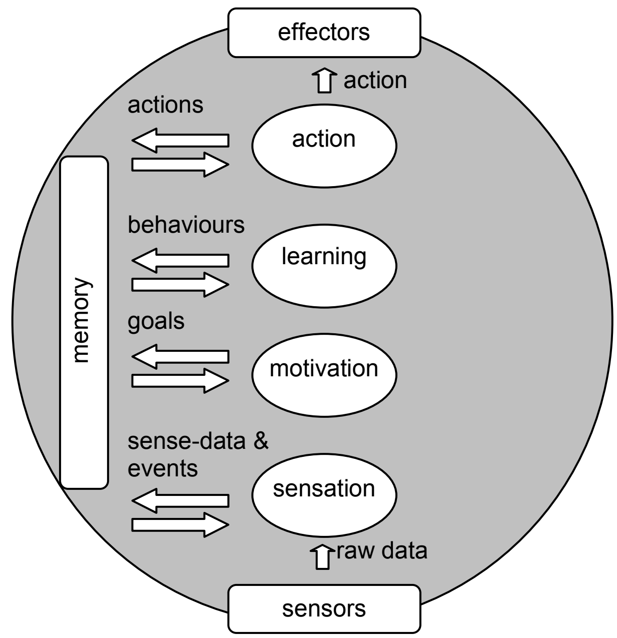

Figure 10. The motivated agent model.

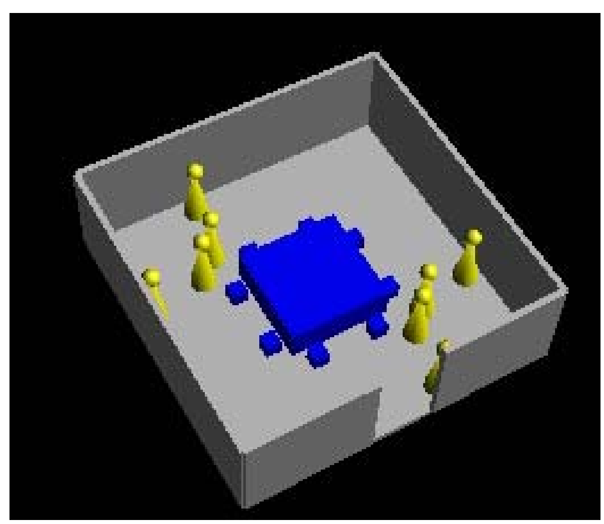

Figure 11. A virtual meeting room simulated in Java 3D.

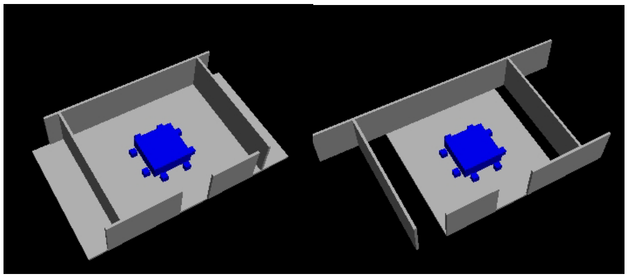

Figure 12. Agents experimenting by modifying the scale and position of the *Wall* and *Floor* objects.

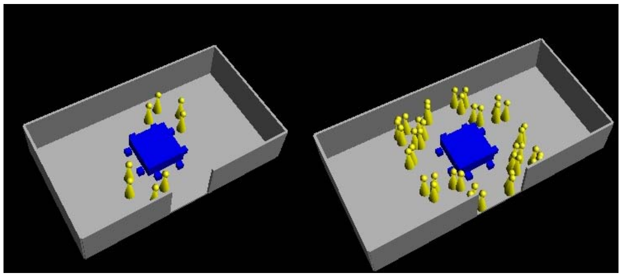

Figure 13. Agents learn to cause interesting events to occur, such as this room expansion which causes the number of avatars using the room to increase.

number of objects from the 3D virtual world. Motivated agents have no communication mechanisms.

We illustrate the performance of this agent model using the example of a virtual meeting room implemented in a simulated Java 3D virtual environment. In this implementation, there is a single agent controlling a *Floor* and five *Wall* objects as shown in Figure 11. The agent can sense the scale and position of each *Floor* and *Wall* object and whether there are avatars nearby. The agent can modify the world by modifying the scale or position of objects as shown in Figure 12. Avatars will use the room if it satisfies the requirements of visual privacy, that is, if all *Wall* and *Floor* objects are scaled so that there are no gaps between them as shown in Figure 13. Larger groups of avatars will tend to favour larger rooms. The agent identifies that, in its experience, increases in the number of avatars in the room are rare. These rare events are interesting to the agent which creates goals to learn how to achieve them. Over time the agent develops behaviours that enable it to influence the number of avatars that will use the room. For example the following "expansion" behaviour will increase the number of avatars likely to use the room:

```
[expand-rear-wall, move-right-wall-right,
expand-front-right-wall, expand-front-left-
wall].
```

Unlike agents using the cognitive and swarm models, motivated agents will not necessarily react to more avatars entering the room, however if they perceive changes in the number of avatars to be interesting they will be motivated to learn how to create bigger rooms in order to entice larger groups to use them. The motivated agent, being more domain independent is easier to implement than the swarm or cognitive models. In addition, because the motivated agent is not limited to a fixed set of goals or microrules it is more capable of responding to a wider range of new situations that may arise. However, because the motivated agent is autonomous and has the freedom to create its own goals, it is more difficult to predict exactly how it will respond to new situations.

5. Concluding Remarks

While agents are able to increase the behavioural complexity of virtual environments and reduce the work required to implement those behaviours, the use of agent models in virtual environments is ultimately limited by the technical attributes of the virtual environment they inhabit. The three agent models described in this paper were implemented in three different virtual environments. Chronologically, the cognitive agent model was implemented first in the Active Worlds environment. Active Worlds exposes an API which allows agents to be written in programming languages such as C/C++ or Java. The Active Worlds SDK was sufficient for implementing cognitive agents which make only a few environmental changes in response to their goals. However, the SDK was not robust enough to deal with the thousands of commands required for a swarm to function or for a motivated agent to learn. It tended to crash after agents had performed only a small number of actions.

Active Worlds' lack of robustness prompted us to use Second Life for our swarm agent experiments. We implemented the swarm agents in Linden Scripting Language (LSL) which has much of the functionality of a full programming language. Two notable exceptions are a limit in the size of a script file and built in server side delays on certain built-in global functions making simulations very slow. The LSL script length prompted us to implement motivated agents in Java. While these agents could have communicated with Second Life via XML-RPC, speed issue and limits on the amount of data that can be communicated in this way prompted us to implement a simple Java 3D virtual environment for motivated agents to learn in.

While the use of agent models in virtual worlds is limited by the robustness of the environment and the richness of the scripting tools available, the benefit of using an agent model in place of scripted behaviours is that the agent model provides a generalised approach that need only be implemented once then applied, possibly with small modifications to a number of agents, controlling multiple objects. This provides coherence in the implementation of object behaviours and means that dynamic behaviour can be achieve over large areas of 3D virtual worlds more easily than using scripted behaviours. That said, we have seen that some agent models are easier to use or provide greater

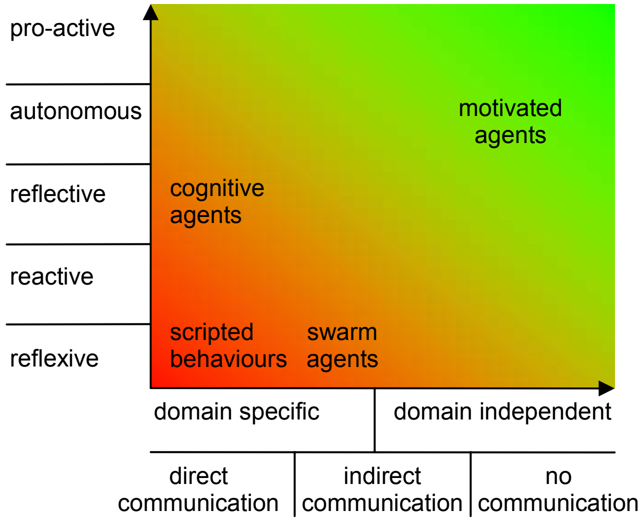

Figure 14. Detailed classification schema for techniques used to create dynamic 3D virtual worlds.

behavioural complexity than others. Our findings are summarised in Figure 14.

From Figure 14 we see that using the swarm agent model gives us the benefit of a single model that can be applied to a range of objects, that is, it is more domain independent than scripted behaviours. The main difference between the cognitive agent model and the swarm agent model is that the cognitive model includes processes that identify patterns and concepts from raw sensor data and use these to determine reactive and reflective behaviour. In contrast, the swarm model reacts directly to sensor data according to a fixed set of behavioural rules without building a hierarchy of labels describing the sense data. The cognitive model has greater initial implementation complexity due to its direct communication requirements, however once the initial implementation is done creating new agents for new environments or problems is simply a matter of creating a new goal set.

The motivated agent model shows a way to further reduce the implementation complexity of the cognitive agent model. Rather than specifying enough knowledge in the perception, conception and hypothesiser parts of the model to identify domain specific patterns and goals, it uses the idea of motivation to allow agents to develop their own goals and behavioural patterns according to their experience in the world. However, in doing this the behaviour of the agents becomes less predictable as they have the freedom to create their own goals rather than selecting from fixed set of pre-programmed goals.

One possible solution to this problem lies in the use of the pro-active behavioural mode. Pro-active agents do not merely react to new situations or hypothesis desirable future situations, rather they can anticipate future undesirable situations and act in advance to avoid them. None of the models discussed so far uses this mode. However, if the motivation process of a motivated agent incorporated the ability to act pro-actively to avoid situations in which other avatars need to rearrange the environment then the motivated agent would tend to learn tasks that anticipate the needs of other avatars and its behaviour would thus become more predictable. The ability to anticipate the needs of human controlled avatars is a step towards intelligent environments and thus a possible focus for future research.

6. Acknowledgements

This research was supported by an Australian Research Council Discovery Grant, the University of Sydney Research Grant Scheme and a National ICT Australia PhD scholarship. National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

7. References

[1] American-Heritage-Dictionaries, *American Heritage Dictionary of the English Language,* Houghton Mifflin Company, 2000.

[2] K. Kasmarik, W. Uther, M-L. Maher, Motivated Agents, The 19th International Joint Conference on Artificial Intelligene, Edinburgh, Scottland, pp 1505-1506, 2005.

[3] M-L. Maher, J. S. Gero, Agent Models of 3D virtual worlds, ACADIA 2002: Thresholds. California State Polytechnic University, Pamona, pp. 127-138, 2002.

[4] M-L. Maher, G. J. Smith, J. S. Gero, Design agents in 3D virtual worlds. In R. Sun (Editor) IJCAI Workshop on Cognitive Modelling of Agent and Multi-Agent Interactions, Acapulco, Mexico, pp. 92-100, 2003.

[5] A. Payman, Swarm Intelligence (http://dsp.jpl.nasa.gov/members/payman/swarm/), Jet Propulsion Laboratory, Pasadena, CA, 2004.

[6] G. J. Smith, M-L. Maher, J. S Gero, Designing 3D virtual worlds as a society of agents. In M-L. Chui, J-Y. Tsou, T. Kyan, M. Morozumi and T-S Jeng (Editors), Digital Design: Research and Practice – Proceedings of the 10th International Conference on Computer Aided Architectural Design Futures, pp105-114, 2003.

[7] L. Steels, When are robots intelligent autonomous agents? *Journal of Robotics and Autonomous Systems,* 15:3-9, 1995.

[8] M. Wooldridge, N. Jennings, Intelligent agents: theory and practice, *The Knowledge Engineering Review,* 10(2):115-152, 1995.