**Introduction to Multi-agent Systems**

**Full Unit Project: Report 2  
Benedict Wilkins**

**Supervisor: Zhiyuan Lou**

**2016**

**Intelligent Agents**

There are a variety of definitions for what a software agent actually is. A good way to look at an agent as a metaphor is to look at the real world. We may consider a human as a real world agent. We exist in an environment – the universe, we have beliefs, desires and intentions which lead us to make decisions and ultimately perform actions which may or may not affect our environment. In essence this is what a software agent does but in a software context – it and its environment will be running on one or more machines. [[[1]](#footnote-1)]

We can model an agent using the real world metaphor. We as humans have a means to affect and observe our environment though use of our body. Parts of our body are used to perform actions in our environment – for example our hands can be used to pick up objects. In agent terms these parts are known as actuators or effectors. Other parts of our body are used for observation – for example our eyes, we look at the world and gather information about our environment. These parts are known as sensors. See figure 1 for an illustration of this idea. It is interesting to see how far this metaphor can be taken and how well it represents software agents – this will be discuss further in the *Starlite* section.

The environment is key aspect of agent based systems. An environment is just some space in which the agents in the system are housed. Agent environments have a number of properties [[[2]](#footnote-2)], in the later section ‘MAS in context’ some attempt will be made to classify the environment of the system to be developed with these properties in mind. The properties should be considered from the point of view of the agent – they are as follows: (According to Russel and Norvig 1995)

* Accessible vs inaccessible – If an environment is accessible, an agent has complete access to the environmental state, its sensors can sense everything that is required to make an optimal decision. If the environment is inaccessible these is some information mission from the agent’s perception.
* Deterministic vs nondeterministic – the environment is deterministic if the current state of the environment has been completely determined by the previous states including the actions performed by the agent. It is nondeterministic otherwise.
* Episodic vs nonepisodic – An agent in an episodic environment has its experience divided into discrete episodes. Each episode is independent from the previous and wont effect the agents future decisions. A nonepisodic environment is one in which an agents experience is continuous in the sense that previous events/actions affect future decisions.
* Static vs dynamic – ­In a static environment an agent can be sure that while making its decision the environment in which it’s situated has not changed. If the environment changes independently of the agent then it is dynamic.
* Discrete vs continuous – If there are a finite number of possible actions and perceptions in the environment it is discrete. Otherwise it is continuous.

In a lot of cases agents are in environment where other agents are also situated and so the need for communication between agents arises. These kind of systems are known as multi-agent systems.

Agent

Environment

Action

Perception

**Figure 1.** An illustration of an agent action on and observing an environment via is actuator and sensor respectively

**Multi-agent Systems (MAS)**

Agents interact with others agents in a variety of different ways depending on context. These interactions can be classified as a result from the following two agent behaviours; cooperative and competitive. Cooperative agents work together to achieve a common goal while competitive agents work to achieve their individual goals. It is not unusual for cooperative agents to display competitive behaviour and vice versa. [[[3]](#footnote-3)] If an agent is *clever* enough it is easy to see how a competitive agent may behave like it is a cooperative. It may temporarily work with other agents if it finds that this is a more effective way of achieving its own goal.

Communication between agents is an important aspect of a MAS and it provides the basis for agent interaction. Cooperative agents must have a means of communication in order to complete their goal as a group. They may need to dynamically organise themselves or communicate information that may be useful to the other agents. Competitive agents may also need to communicate. An agent based e-commerce trading system is a good example of a need for competitive communication. Each agent in the system will represent a real world entity such as a person or corporation. Their goal is to negotiate the best deal for the entity they are representing. They will require some communication protocol in order to negotiation. [[[4]](#footnote-4)] It is up to the agent designer to provide these communication protocols in way that best suits the system.

**Distributed agent environment**

The nature of the smart grid forces the system to utilise some distributed agent environment. Smart meters are physically separate entities each with an environment that they are responsible for – the household of the consumer. The smart meter agents (SMA) will need to be capable of communicating information to physically separate systems. This can be done using an IP protocol. Each SMA will be capable of communicating in this way with its manager agent. There are a few different possible configurations of management agents all with different associated positives and negatives and are to be discussed in this section.

The hierarchical structure of the MAS implies groupings of the agents. The SMAs already have a natural grouping, by geographical location or address which is already used by all household service providers. The manager agent’s job should be to forward data on behalf of its group as well as manage and control them by forwarding commands from higher in the hierarchy.

To keep the system as de-centralised as possible it may be desirable to have the SMAs doing management themselves. They could delegate one of the other agents in their group to be a manager. This configuration may avoid system failures; if the delegated agent fails, a different SMA can be delegated as the manager. There is a potential privacy issue with sending data to a Smart Meter in a household, some measures would have to be in place to prevent data theft or alteration at the management agent location.

The alternative configuration is to have the management agent reside further up the hierarchy. It would be more complicated to deal with system failures as the system would have no alternative management agent. However there will be no privacy issues with sending the data.

**MAS in Context**

There have been various other studies involving multi-agent systems and the smart grid. [[[5]](#footnote-5)][[[6]](#footnote-6)][[[7]](#footnote-7)]. *Multi-Agent Systems (MAS) controlled Smart Grid - A Review* in particular gives a good summary of relevant studies. Most are concerned with managing operations and control of the smart grid and some address problems such as the centralisation of power plants and service restoration.

This project is less concerned with the technology that will be used to implement the control/management of the smart grid after predictions have been made. However from the studies mentioned previously it seems likely that MAS will play a key role. This motivates the use of an extendable MAS in this project.

**Classifying the agent environment**: (From afore mention standard environmental attributes (Russel and Norvig 1995)

To begin classifying the environment we must first define what exactly we mean in context. Each SMA is situated in its own ‘house’ environment in which it interacts (take readings). The house environment is however linked to the global environment; the environment containing all houses and any other agents in the system. This is the environment which we will try to classify as it contains all other environments.

**Inaccessible** – due to the nature of networking. Each SMA does not know what is happening in some other partition of the environment. It could ask for more information from the other agents but again there is no guarantee of a reply. Similarly the manager agents (whether they are central or not) do not have direct access to the house environments.

**Nondeterministic** – again due to the nature of networking. It is possible that the system will fail at some point, an SMA may fail and so no data will be sent up the hierarchy. It should be noted that the receiving agents should be able to handle not receiving any data. We do not want the employed machine learning method to fail because of this.

**Nonepisodic** – This attribute is a little harder to pin down. It could be said that the SMAs will reside in an episodic environment, all they will do it send data regardless of success or failure in the previous episode. In the simulated system the agents will receive an energy usage value every logical[[8]](#footnote-8)\* half hour interval. However it may be the case that in the real system the agent will be continuously monitoring the energy usage. Whether we consider the physical monitoring device part of the agent is up for debate and should be carefully considered in a fully deployable version of the system. The non-SMA (the agents higher up the hierarchy) are situated in a nonepisodic environment. Although data may be sent every half hour from the SMAs perspectives their time may be out of sync (this should be minimised as it will have an impact on prediction). The non-SMAs must be capable of continually receiving packets from the SMAs.

**Dynamic** – The environment will change independently of the each agent. Network failures and other agents may affect the environment.

**Discrete** – SMAs may only take an energy reading and send it. Non-SMAs may only receive messages, send messages.

**Agent Architecture**The agent architecture was taken from the …

The arrows represent the forwarding of events, results and actions.

Body

* The mind generates an action based on some reasoning, this is forwarded to the brain, which forwards it to the body. The body then generates an event from the action which is forwarded to the actuator. The actuator then forwards the event to the environment.
* The sensor receives a result from the environment, this is the agent’s perception. The sensor forwards it to the body and in the same fashion as above it reaches the mind as a result. The mind can then process the result as it sees it.

**In Java:** Each part of the agent is represented as a class, each class is an observer observable pair (each class can communicate via observer/observable design pattern with its respective parts). The environmental also uses the observer/observable design pattern in order to send and receive results and events respectively. The body class contains all of the parts of the agent and so is essentially the complete agent. A body may also have an appearance which is defined as being; the external appearance of the body – what the agent looks like to other entities in the environment.

The framework for the agents is given by the GAWL (Generic Agent World Library) package. This framework also includes packages for, actions, events, perceptions, environment physics and the observer/observable implementation.

**Environment Architecture (Simulation)**

**A house:**

**Data Generator**

**­­­­**

The SMA will receive a reading from the Data Reader agent – who in the simulation will be set to read a value from the Data Generator at a given time frame. The time frame will be decided at the beginning of the simulation, each frame will represent a half hour interval in the deployable system. The house environment/physics will be responsible for forwarding the Data Reader -> SMA message. Once the message is received the SMA will forward the data up the agent hierarchy. The Data generator module will be global and will depend on the set up of the simulation. It will provide data reading from all houses based on their properties e.g. what financial positon the house is in.

**Environment and Physics:**

Each house has its own physics and environment, these are responsible for message forwarding within a house and will evaluate events that pass through them making sure they are valid in context. For example a house environment will receive an event from the Data Reader agent, it will forward it to the house physics which will evaluate it in context, make sure it is possible and valid. It will then execute the action (in this case executing the action means giving it back to the environment enabling a forward to the SMA).

**Global Environment and Physics:**

The arrows represent the transfer of events. Each SMA in its respective House Environment will send a message to the Global Environment. This will be processed (sent to the physics) and given to the Group agent. The Group agent may be a neighbourhood, region, country or other defined at the start of the simulation. The Group agent will aggregate the data in some way and format it so that it may be useful to the Predictor agent. The aggregated data will be sent in the same manner from the Group agent to the Predictor agent.

**Global Environment**

**House Environment**

**House Environment**

**House Environment**

**Global Physics**

There will be room in the architecture to provide multi group agents for different aggregation if needed. As an example we may have multiple neighbourhoods about which we want to make an aggregated prediction (a prediction regarding all houses in each neighbourhood). This can be done by simply having two Group (neighbourhood) agents with their respective House Environments. There is an obvious bottle neck at the Global environment/physics, this may be avoided by making a logical partition of the global environment e.g. for each group; a version of the global environment will be created to handle their communication specifically. In the real system the idea of Global environment will not hold as strong. The SMAs will forward their messages via IP communication – the machine who receives the message will essentially be the Global Environment.

# Bibliography

Hoen, P., Tuyls, K., Panait, L., Luke, S., & La Poutré, J. (2006). An Overview of Cooperative and Competitive Multiagent Learning. *Lecture Notes Computer Science, 3898*, 1-46.

Narkhede, M. S., Chatterji, S., & Ghosh, S. (2013). Multi-Agent Systems (MAS) controlled Smart Grid - A Review. *Proceedings on International Conference on Recent Trends in Engineering and Technology.* IJCA.

Rahwan, I., Kowalczyk, R., & Pham, H. (2002). Intelligent Agents for Automated One-to-Many e-Commerce Negotiation. *Autralian Computer Science Communications, 24*(1), 197-204.

Rogers, A., Ramchurn, S., & Jennings, N. (2012). Delivering the Smart Grid: Challenges for Autonomous Agents and Multi-Agent Systems Research. *Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*, (pp. 2166-2172). Toronto.

Russell, S., & Norvig, P. (1995). *Artifical Intelligence. A modern apprach.* Prentice Hall.

Wooldridge, M. (2009). *An introduction to multiagent systems.* John Wiley & Sons.

Yilmaz, C., Albayrak, S., & Lutzenberger, M. (2014). Smart Grid Architectures and the Multi-Agent System Paradigm. *Proceedings of the 4th International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies.*, (pp. 90-95). Chamonix.

1. (Wooldridge, 2009) [↑](#footnote-ref-1)
2. (Russell & Norvig, 1995) page 46 [↑](#footnote-ref-2)
3. (Hoen, Tuyls, Panait, Luke, & La Poutré, 2006) [↑](#footnote-ref-3)
4. (Rahwan, Kowalczyk, & Pham, 2002) [↑](#footnote-ref-4)
5. (Yilmaz, Albayrak, & Lutzenberger, 2014) [↑](#footnote-ref-5)
6. (Rogers, Ramchurn, & Jennings, 2012) [↑](#footnote-ref-6)
7. (Narkhede, Chatterji, & Ghosh, 2013) [↑](#footnote-ref-7)
8. \* The half hour interval will not be in real time, it will be scaled to simulation speed. [↑](#footnote-ref-8)