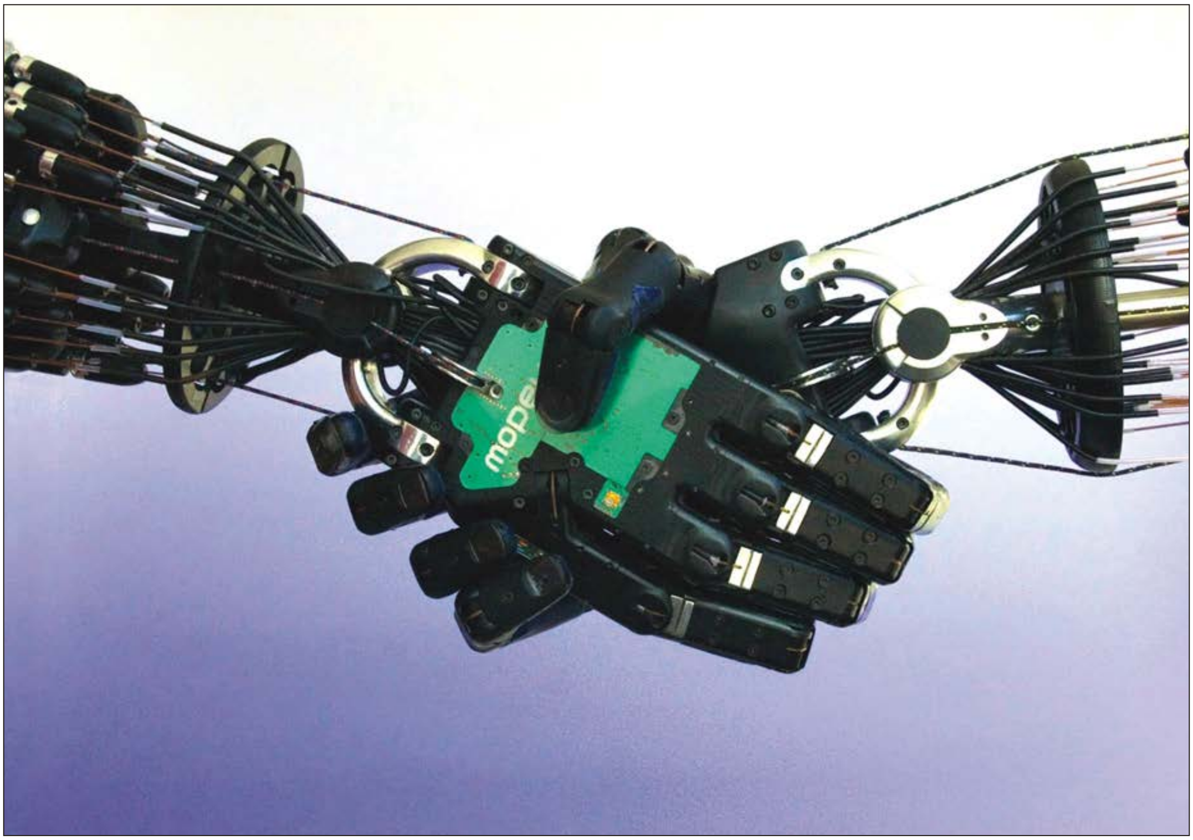

Journal of Artificial Intelligence

ISSN 1994-5450

Journal of Artificial Intelligence 6 (3): 191-209, 2013
ISSN 1994-5450 / DOI: 10.3923/jai.2013.191.209

Optimal Environmental Simulation Settings to Observe Exceptional Events in Social Agent Societies

[1]Moamin A. Mahmoud, [1]Mohd Sharifuddin Ahmad, [1]Azhana Ahmad, [2]Aida Mustapha, [1]Mohd. Zaliman M. Yusoff and [3]Nurzeatul Hamimah Abdul Hamid

[1]College of Information Technology, Universiti Tenaga Nasional, Kajang, Selangor, Malaysia
[2]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor, Malaysia
[3]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia

Corresponding Author: Moamin A. Mahmoud, College of Information Technology, Universiti Tenaga Nasional, Kajang, Selangor, Malaysia

ABSTRACT

Social norms learning in agent societies through reward or penalty observations have become the subject of interest in many studies. However, very few studies have examined the optimal environmental settings that would allow agents to learn through such observations effectively. This study presents a combination of environmental simulation parameters to discover the optimal settings for observing reward or penalty events, which are called the exceptional events, within a social agent group. The environmental settings consist of several variables which are the cycle time, observation limit of detector agent, domain size, population density of domain agents and occurrence of reward or penalty (exceptional) events in the domain. The value of each variable is arbitrarily set to low, medium or high. To implement the simulation, a virtual environment has been created with the variables settings to examine different situations. Within the steps of the tests, some cases are excluded because they do not significantly contribute to optimal environment for social learning. The results of the tests show that each variable has different effect on the environment and that a variable that has a strong positive effect does not individually offer the optimal solution. However, combining variables that have strong positive effects could offer optimal solutions. Briefly, the study aims to examine and identify the effect of some environmental variables on observation process of exceptional events and suggests the optimal settings to learn through observation.

Key words: Intelligent software agent, normative system, simulation model, exceptional events observation

INTRODUCTION

The concepts of norms and normative systems are used to determine the behaviours of agents within a society and are commonly accepted as efficient means to normalize their behaviours (Alberti *et al.*, 2011). Recently, many studies in agent-based systems have been conducted to explore social norms learning or identification within normative systems (Choi and Kim, 2009; Centeno *et al.*, 2010; Centeno and Billhardt, 2012; Savarimuthu *et al.*, 2010a-c; Campos *et al.*, 2010; Andrighetto *et al.*, 2007).

In social learning, an agent learns new behaviour through observation of rewards or penalties and by monitoring the actions of other agents (Hollander and Wu, 2011; Conte and Dignum, 2001; Conte and Paolucci, 2001; Bandura, 1997). Most of current studies exploit the events that are exceptional (Centeno *et al.*, 2010; Centeno and Billhardt, 2012; Natarajan *et al.*, 2010; Savarimuthu *et al.*, 2010a-c; Campos *et al.*, 2010; Andrighetto *et al.*, 2007), specifically those events that entail rewards or penalties to learn or identify the obligation or prohibition norms. Consequently, an agent identifies the obligation or prohibition norms by observing exceptional events from a given series of events. Savarimuthu *et al.* (2010a-c) develop two algorithms, one to identify obligation norms, which they called Obligation Norm Identification (ONI) and another to identify the prohibition norms, which they called Candidate Norm Inference (CNI). These two algorithms are designed based on the observation of exceptional events. Natarajan *et al.* (2010) developed a technique based on reward functions to observe an agent acting in the environment. The observations entail the agent's behaviour over time.

While social norms learning through observations of exceptional events have been the subject of intense study (Choi and Kim, 2009; Centeno *et al.*, 2010; Centeno and Billhardt, 2012; Savarimuthu *et al.*, 2010a-c; Campos *et al.*, 2010; Andrighetto *et al.*, 2007), few studies have examined the optimal environmental simulation settings that would allow agents to learn through such observations effectively (Sen and Airiau, 2007; Centeno *et al.*, 2010; Centeno and Billhardt, 2012; Savarimuthu *et al.*, 2010a-c). Consequently, this study examines some conditions under which environmental variables influence agents' environmentally-related behaviours.

This study presents a combination of simulated environmental variables' settings to discover the optimal settings for observing reward or penalty events, which are called the exceptional events, within a social agent group. The simulation is implemented by creating a virtual environment with the variables settings to examine different scenarios. A scenario consists of two types of agents: detector agents, i.e., visitor agents and domain agents, i.e., local agents. The settings of the scenario are made on several variables, i.e., the condition of detection, which is the cycle time; the ability of detector agents, which is their limit of observation and the domain variables, which are domain size, population density of domain agents and occurrence of reward and penalty events in the domain. The value of each variable is categorised as low, medium and high. Within the steps of the tests, there are some variables that are excluded because they do not significantly contribute to the optimal environment for social learning.

A simulation system is a low fidelity operation of a model of the system (Maria, 1997). It generates a number of tracks and gathers statistics from these tracks to measure the desired performance (Sanders, 2005). Simulation offers a strong methodology in complex behaviours (Harrison *et al.*, 2007), because of its ability to model random behaviour or variation (Reeb and Leavengood, 2003). Harrison *et al.* (2007) defined a computer simulation as "a computational model of system behaviour coupled with an experimental design".

Computer simulation is considered as a third methodology in scientific progress following the theoretical and empirical analysis methodologies (Axelrod, 1997; Waldrop, 1993). It starts with modelling of behaviours of the target system to experiment various scenarios (Harrison *et al.*, 2007). Computer simulation comprises a computational model and experimental designs. A computational model consists of variables which are the components of the system and the processes of changing these variables (Harrison *et al.*, 2007). Five elements are involved in experimental designs which are, initial settings, time constraints, output determination, repetition and variations (Harrison *et al.*, 2007). The outcomes are represented by some behaviour functions of the system and are calculated from the variables of the system after each run or for each time period depending on the simulation's target. Subsequently, the results may be subjected to additional analyses. Based on the parameters and initial settings, simulation produces large quantities of data for every variation involving the system's variables values. It can also produce results for each time period with summary statistics across iterations (Harrison *et al*., 2007).

There are several types of simulation models, one of these is agent-based models. Agent-based models focus on modeling the behaviours of adaptive actors, which structure the system and influence each other by interactions (Macy and Willer, 2002; Van Dyke Parunak *et al*., 1998). The behavior of the system is an emergent property of the interactions between agents (Harrison *et al*., 2007).

While ample studies have exploited the reward or penalty observations approach to make an agent learns a new behaviour (Hollander and Wu, 2011; Conte and Dignum, 2001; Conte and Paolucci, 2001; Bandura, 1997), few studies have discussed the effectiveness of environmental settings on the success of learning or identification of social norms. Sen and Airiau (2007) proposed a model in social norms emergence using learning from interaction experiences base on a reward function. In their model, they study the effects of population size, number of actions and number of interactions period. They developed a simulation model for an agent to learn the rules of the road. In particular, they focus on the problem of driving side of the road and who earns when there are two agents in neighbouring roads reach at the same time an interaction (Sen and Airiau, 2007).

Savarimuthu *et al*. (2010a) proposed a norm identification technique that infers the norms of an agent community without the norms being explicitly imposed on the agents. Their mechanism exploits the sanctioning action in the environment to identify the obligation norms. They develop algorithm to identify the tip norm in an agent-based simulation of a virtual restaurant in which agents are located in the restaurant where other agents entering the restaurant may not be aware of the protocol associated with ordering and paying for food items and the associated norms. In their simulation they test several factors which are grid size and observation threshold. They discovered that when the observation threshold of the agents increases, the agents identified the norms faster and when the grid size increases, the number of candidate norms generated decreases.

SIMULATION MODEL

This section presents the simulation model by creating a virtual scenario for the elevator domain. By using Win-Prolog programming language (http://www.lpa.co.uk), three interfaces are created in three windows (Fig. 1). The windows have the functions to create a new domain, select and run a domain, and set the variables of the domain. The first window (Fig. 1) is used to create a new domain when the environmental variables are set as shown by the upper part of the second window (Fig. 1) and the lower part of the second window displays the results of computation. The third window (Fig. 1) is used to select and test the domain.

Variables classification: The variables are classified into three categories and each variable is set an arbitrary value of Low, Medium or High for testing. This study argued that such values are adequate to show the effects of variables' combinations.

- The first category of variables belongs to the Task Condition category, which is the Cycle Time. It is the time given for one cycle of events

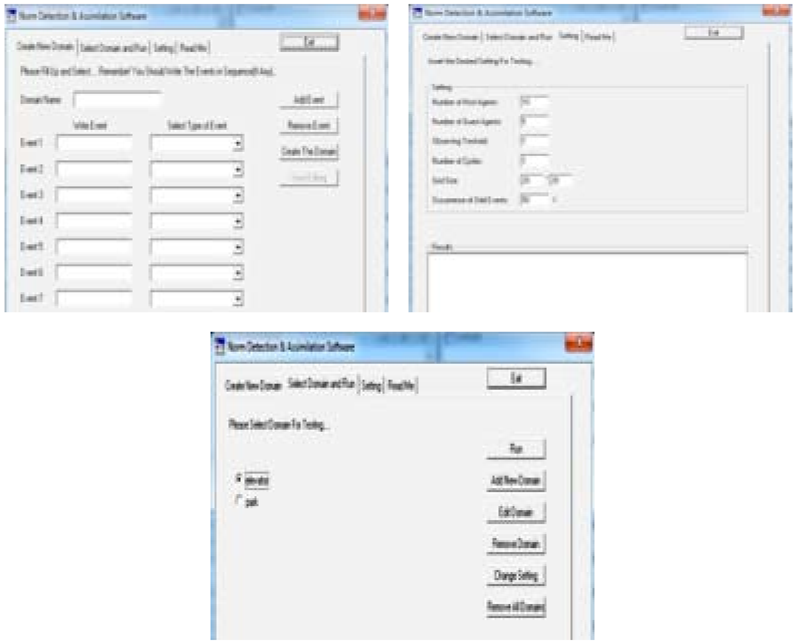

Fig. 1: Simulation interfaces

- The second category of variables belongs to Agent Ability category, which is the Observation Limit of detector agent. It is the extent to which a detector agent is able to observe the actions of domain agents
- The third category belongs to the Observed Domain category. The pertinent variables are
- Domain Size, which is defined by a grid size of M by N
- Population Density of Domain Agents, which is the spread of domain agents occupying the grid
- Occurrence Rate of Reward or Penalty events, which is the frequency of reward or penalty events as observed by a detector agent

The variables units and settings: In this simulation, each variable is assigned a measurement unit as follows:

- Cycle time: is defined as the time given to a number of event cycle for the domain. For example, in the elevator domain, the event cycle could be wait, enter, excuse, depart. Each domain agent assimilates and enacts these events. A detector agent observes and learns those events are given a number of cycles, which could be one cycle or more
- Domain size: the domain is simulated as a two-dimensional grid and the grid size represents the domain size. The grid M*N represents the size of domain X. Figure 2 shows an example of a 10*10 grid with domain and detector agents strewn within the gird
- Observation limit of detector agent: Each observing detector agents has a limit of observation to monitor the domain agents' behaviours. This study assumes that the agents in the grid are able to observe the surrounding agents within the limitation threshold. The unit of measuring the limit is a cell of the grid in the North, South, East and West direction. For example, a detector agent is able to observe other agents located one cell besides it. A mathematical model is developed to determine the visible cells for detector agents

Determining visible cells: If there is detector or observer agent, A_O; in a grid size, M*N; the location of A_O in the grid, $G_{(m,n)}$; the threshold limit, T and the observable (visible) cell, O_C; of the grid in Fig. 3, the following three cases are apparent:

- The first case is when the agent A_O is at a corner of the grid, e.g., the cell (3, 3). In this case, the agent can see in two directions only, as shown
- The second case is when the agent in at the middle, e.g., the cell (2, 2). For this case, the agent can see in four directions
- The last case is when the agent is at one of the sides e.g., the cell (2, 1). In such case, the agent can see in three directions

$$\text{If } M = m_1, m_2, m_3, \ldots, m_i$$
$$N = n_1, n_2, n_3, \ldots \ldots, n_j$$
$$T = t_1, t_2, \ldots, t_k$$
$$O_C = c_1, c_2, \ldots, c_\lambda$$

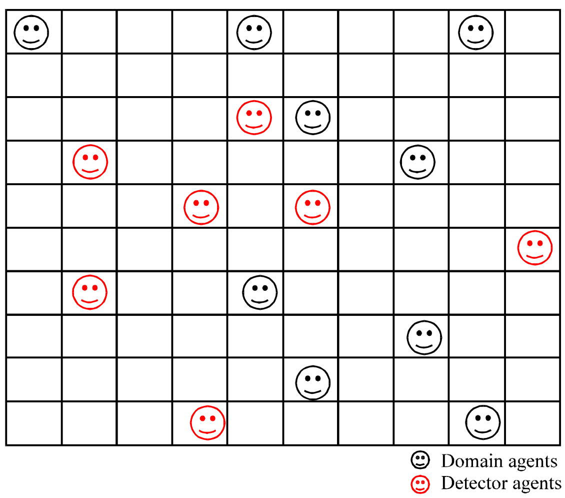

Fig. 2: A sample 10*10 Grid

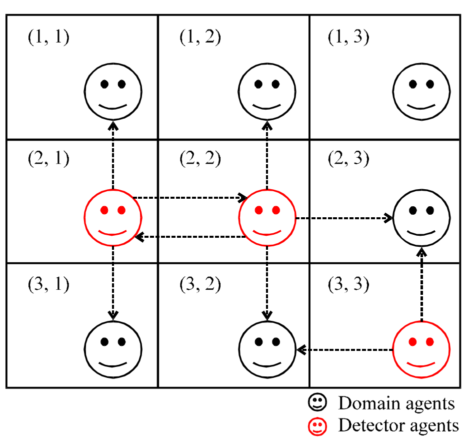

Fig. 3: Visible cells in 3*3 Grid

Case 1, when $m_i = 1$, $n_j = 1$:

$$c_\lambda = (m_i + t_k)\ , \text{ where } c_\lambda = M \ldots\ldots\ldots 1$$
$$c_\lambda = (n_j + t_k)\ , \text{ where } c_\lambda = N \ldots\ldots\ldots 2$$

Case 2, when $m_i = M$, $n_j = N$,

$$c_\lambda = (mi - tk)\ , \text{ where } c_\lambda = 1 \ldots\ldots\ldots 1$$
$$c_\lambda = (nj - tk)\ , \text{ where } c_\lambda = 1 \ldots\ldots\ldots 2$$

Case 3, when $m_i \neq 1$, $m_i \neq M$, $n_j \neq 1$, , $n_j \neq N$,

$$c_\lambda = (m_i + t_k), \text{ where } c_\lambda \leq M \ldots\ldots 1$$
$$c_\lambda = (m_i - t_k), \text{ where } c_\lambda \geq 1 \ldots\ldots 2$$
$$c_\lambda = (n_j + t_k), \text{ where } c_\lambda \leq N \ldots\ldots 3$$
$$c_\lambda = (n_j - t_k), \text{ where } c_\lambda \geq 1 \ldots\ldots 4$$

Other cases are subset of these above cases, for example when $m_i = M$, $n_j \neq 1$, , $n_j \neq N$,

$$c_\lambda = (m_i - t_k), \text{ where } c_\lambda \leq 1 \ldots\ldots 1$$
$$c_\lambda = (n_j + t_k), \text{ where } c_\lambda \geq N \ldots\ldots 2$$
$$c_\lambda = (n_j - t_k), \text{ where } c_\lambda \leq 1 \ldots\ldots 3$$

Accordingly, when an agent is at location $G_{(m,n)}$ in the grid M*N, then:

- If $G_{(m=1,n=1)} \Rightarrow O_{(m+T,\ n)}$, $O_{(m,\ n+T)}$
- If $G_{(m=M\ ,n=N)} \Rightarrow O_{(m-T,n)}$, $O_{(m,n-T)}$
- If $G_{((m \neq 1,\ m \neq M),\ (n \neq 1,\ n \neq N))} \Rightarrow O_{(m+T,\ n)}$, $O_{(m-T,n)}$, $O_{(m,\ n+T)}$, $O_{(m,n-T)}$

For example, for a grid size of 3*3, and T=1 as shown in Figure 3, following some locations tests:

- Agent A located in $G_{(1,1)} \Rightarrow O_{(2,\ 1)}$, $O_{(1,\ 2)}$
- Agent A located in $G_{(2,2)} \Rightarrow O_{(3,\ 2)}$, $O_{(1,2)}$, $O_{(2,\ 3)}$, $O_{(2,1)}$
- Agent A located in $G_{(3,2)} \Rightarrow O_{(2,2)}$, $O_{(3,\ 3)}$, $O_{(3,1)}$
- Agent A located in $G_{(2,1)} \Rightarrow O_{(3,\ 1)}$, $O_{(1,1)}$, $O_{(2,\ 2)}$

- Population density of domain agents: This is the number of domain agents in the grid. If there are many agents, then the density is considered to be high
- Occurrence Rate of Reward or Penalty (Exceptional) events: This variable determines the frequency of exceptional events happening in the domain

Table 1 explains and clarifies each variable with regard to its symbol and its values for Low, Medium and High.

Table 1: Variables units and settings

Variables	Low	Medium	High
Observation limit (O_L)	1 Cell	2 Cell	3 Cell
Domain grid size (G_S)	5*5	10*10	20*20
Cycle time (C_N)	1 Cycle	2 Cycle	3 Cycle
Occurrence of exceptional events (E_O)	25%	50%	75%
Population density of domain agents (A_C)	5 Agents	10 Agents	20 Agents

SIMULATION SETTINGS

The simulation model is built based on the above environmental variables and their values. The following defines the supplementary entities to complete the simulator:

Normative protocol generator: This generates events for domain agents to enact the normative protocol in some specific scenario. The generator constrains the enactment for each agent via a random function to distribute the events among the agents.

Location generator: When a scenario is enacted, the location generator moves each domain agent randomly from cell to cell. For example, if the normative protocol is wait, enter, excuse, depart, the location generator moves an agent in the *wait* cell, say at $G_{(1,2)}$, to the next cell, say, $G_{(2,2)}$ to enact the enter event and so on.

Agents types: There are two types of agents in any scenario, which are:

- **Detector Agents (A_{DT}):** These agents roam in the domain to detect exceptional events to learn from the domain agents
- **Domain Agents (A):** These agents have knowledge about the domain's norms, which could be obligation, prohibition, or recommendation norms. The domain agents enact the normative protocol, which is generated and assigned by the normative protocol generator

The domain norms (elevator scenario): This section presents the enacted norms of the elevator domain and assigns each norm type as follows:

- Enacted Norms (Np) : Wait, enter, greet, litter, excuse, depart
- Obligation norms (O) : Excuse
- Prohibition Norms (P) : Litter
- Recommendation (R) : Greet

Based on the norms' types (Obligation, Prohibition, Recommendation), according to Ahmad *et al*. (2011), an agent is rewarded or penalized as follows:

- If the agent enacts the obligation norms → no penalty
- If the agent does not enact the obligation norms → penalty
- If the agent enacts the prohibition norms → penalty
- If the agent does not enact the prohibition norms → no penalty
- If the agent enacts the recommendation norms → reward
- If the agent does not enact the recommendation norms → no reward

DOMAIN TESTING

Initial testing: This section discusses the tests for each variable and its effect on different values of Low, Medium and High, upon observation of exceptional events. For each test, the mean percentage of the test results is calculated after a number of runs, using this formula:

$$\text{Mean} = ((\text{Number of } A_{DT} \text{ in all runs/number of all } A_{DT})/\text{Number of runs})\times 100$$

Test No. 1: This test measures the effect of Observation Limit (O_L) on the observation process.

Settings:

A_{DT}	10 Agents	Detectors agents
A_C	5 Agents	Low
C_N	1 cycle	Low
G_S	20 * 20	High
O_L	N Cell	Test Low, Medium, High
E_O	50%	Medium

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
N =1 Cell - Low						
A_{DT}	1	2	0	1	2	12.0%

The mean value is calculated as follows:

- Number of A_{DT} in all runs = 6
- Number of all A_{DT} = 10
- Number of runs = 5
- Mean = (6/10)/5)*100 ⇒ Mean = 12%

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
N =2 Cells - Medium						
A_{DT}	3	2	3	4	3	30.0%
N = 3 Cells - High						
A_{DT}	4	2	3	4	5	36.0%

Figure 4, shows the effect of observation limit on observation of exceptional events.

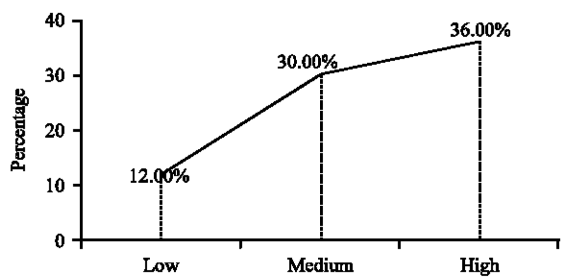

Fig. 4: Observation limit test Low: 12%, Medium: 30%, High: 36%

Test No. 2: This test measures the effect of Cycle Time (C_N) on the observation process.

Settings:

A_{DT}	10 Agents	Detectors Agents
A_C	5 Agents	Low
C_N	N cycle	Test Low, Medium, High
G_S	20 * 20	High
O_L	1 Cell	Low
E_O	50%	Medium

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
N =1 Cycle - Low						
A_{DT}	2	1	1	2	1	14.0%
N = 2 Cycles - Medium						
A_{DT}	4	4	2	3	4	34.0%
N =3 Cycles - High						
A_{DT}	3	4	3	6	5	42.0%

Figure 5, shows the effect of cycle time on observation of exceptional events.

Test No. 3: This test measures the effect of Domain (Grid) Size (G_S) on the observation process.

Settings:

A_{DT}	10 Agents	Detectors Agents
A_C	5 Agents	Low
C_N	1 cycle	Low
G_S	M*N	Test Low, Medium, High
O_L	1 Cell	Low
E_O	50%	Medium

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
M *N=5*5 - Low						
A_{DT}	6	6	4	7	5	56.0%
M *N=10*10 - Medium						
A_{DT}	3	2	3	5	4	34.0%
M*N=20*20 - High						
A_{DT}	1	2	1	1	1	12.0%

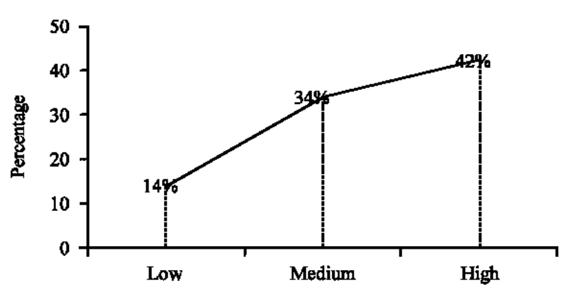

Fig. 5: Cycle time test Low: 14%, Medium: 34%, High: 42%

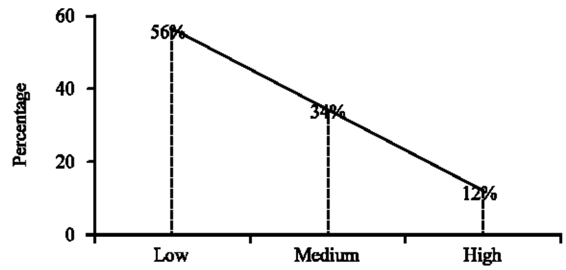

Fig. 6: Grid size test Low: 56%, Medium: 34%, High: 12%

Figure 6 shows the effect of grid size on observation of exceptional events.

Test No. 4: This test measures the effect of Exceptional Events Occurrence (E_O) on observation process,

Settings:

A_{DT}	10 Agents	Detectors Agents
A_C	5 Agents	Low
C_N	1 cycle	Low
G_S	20*20	High
O_L	1 Cell	Low
E_O	N%	Test Low, Medium, High

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
N = 25% - Low						
A_{DT}	1	2	1	1	0	10.0%
N = 50% - Medium						
A_{DT}	2	1	2	1	1	14.0%
N = 75% - High						
A_{DT}	2	3	2	3	2	24.0%

Figure 7 shows the effect of exceptional events occurrence on observation of exceptional events.

Test No. 5: This test measures the effect of Population Density of Domain Agents (A_C) on the observation process.

Settings:

A_{DT}	10 Agents	Detectors Agents
A_C	N Agents	Test Low, Medium, High
C_N	1 cycle	Low
G_S	20*20	High
O_L	1 Cell	Low
E_O	50%	Test Low, Medium, High

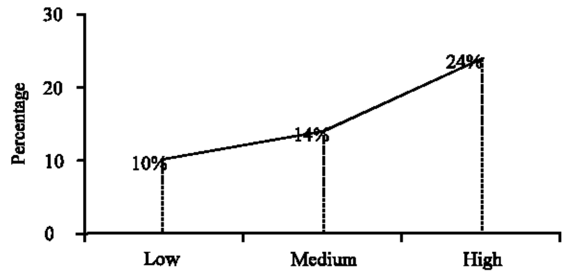

Fig. 7: Exceptional events occurrence test Low: 10%, Medium: 14%, High: 24%

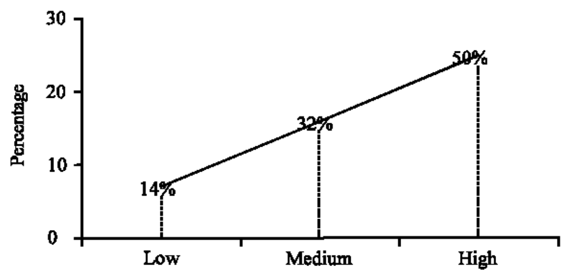

Fig. 8: Population density test Low: 14%, Medium: 32%, High: 50%

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
N =5 Agents - Low						
A_{DT}	1	1	2	1	2	14.0%
N =10 Agents - Medium						
A_{DT}	4	3	4	3	2	32.0%
N =20 Agents - High						
A_{DT}	5	4	4	5	7	50.0%

Figure 8 shows the effect of population density on observation of exceptional events.

Summary of results of initial testing: As shown in Fig. 9, the results of the initial testing are as follows:

- Grid size, G_S, is inversely proportional with observation of exceptional events. If G_S increases, the observation of exceptional events decreases and vice versa
- Population density of domain agents, A_C;cycle time, C_N; observation limit, O_L; exceptional events occurrence, E_O; are directly proportional with observation of exceptional events. If one of these factors increases, the observation increases and vice versa
- Population density of domain agents, A_C, and Grid size, G_S, have strong positive effect (50, 56%, respectively) on observation of exceptional events
- Cycle time, C_N and observation limit, O_L have the same intermediate positive effect (42, 42%, respectively) on observation of exceptional events
- Exceptional events occurrence, E_O, has the minimum positive effect (24%) on observation of exceptional events

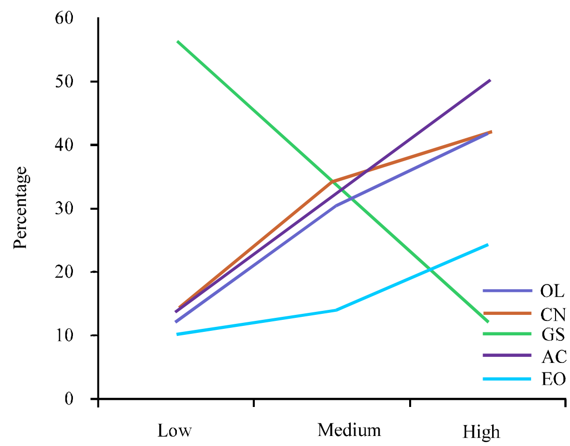

Fig. 9: Rate of observation of exceptional events among variables G_S Low: 56%, A_C High: 50%, C_N High: 42%, O_L High: 42%, E_O High: 24%

- The results show that a variable of strong positive effect is not enough to bring the detector agents to an acceptable level of performance. They should be provided with more than one variable of strong positive effect. The next section analyses the effect of such environments

Figure 9 shows the performance of observation of exceptional events for each variable.

Advanced testing: This test is based on the initial results and includes combinations of variables. However, the variable, E_O, is excluded because it gives minor effect on the observation process. To determine the optimal environment's variables for a detector agent to effectively observe the exceptional events, the following tests set two or more variables of strong positive effect.

Test No. 6: This test measures the effect of Domain (Grid) Size, G_S and Observation Limit, O_L, on the observation process.

Settings:

A_{DT}	10 Agents	Detectors Agents
A_C	5 Agents	Low
C_N	1 cycle	Low
G_S	N * N	Test Low, Medium, High
O_L	X Cell	Test Low, Medium, High
E_O	50%	Medium

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
M*N=10*10 – Medium (M), X =2 Cells – Medium (M),						
A_{DT}	6	4	6	4	4	48.0%
M*N=10*10 – Medium (M), X =3 Cells – High (H)						
A_{DT}	5	5	7	7	6	60.0%
M*N=5*5 – Low (L), X =2 Cells – Medium						
A_{DT}	8	7	7	6	5	66.0%
M*N=5*5 – Low (L), X =3 Cells – High (H)						
A_{DT}	8	7	7	9	8	78.0%

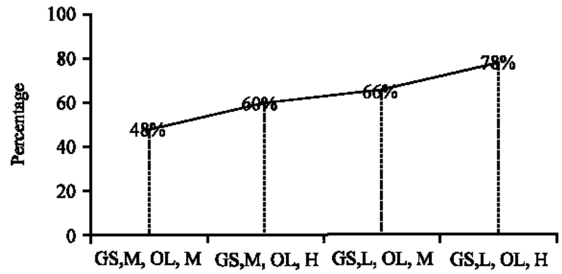

Fig. 10: Grid size and observation limit test G_S and O_L Medium: 48%, G_S Medium and O_L High: 60%, G_S Low and O_L Medium: 66%, G_S Low and O_L High: 78%

Figure 10 shows the effect of grid size and observation limit on observation of exceptional events.

Test No. 7: This test measures the effect of Domain (Grid) Size, G_S and the Cycle Time, C_N, on the observation process.

Settings

A_{DT}	10 Agents	Detectors Agents
A_C	5 Agents	Low
C_N	X cycle	Test Low, Medium, High
G_S	N*N	Test Low, Medium, High
O_L	1 Cell	Low
E_O	50%	Medium

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
M*N=10*10 – Medium (M), X =2 Cycles – Medium (M)						
A_{DT}	6	3	4	6	6	50.0%
M*N=10*10 – Medium (M), X =3 Cycles – High (H)						
A_{DT}	7	6	6	8	6	66.0%
M*N=5*5 – Low (L), X =2 Cycles – Medium						
A_{DT}	7	8	6	8	9	76.0%
M*N=5*5 – Low (L), X =3 Cycles – High (H)						
A_{DT}	9	10	8	9	10	92.0%

Figure 11 shows the effect of grid size and cycle time on observation of exceptional events.

Test No. 8: This test measures the effect of Domain (Grid) Size, G_S and Population Density of Domain Agents, A_C, on the observation process.

Settings:

A_{DT}	10 Agents	Detectors Agents
A_C	X Agents	Test Low, Medium, High
C_N	1 cycle	Low
G_S	N*N	Test Low, Medium, High
O_L	1 Cell	Low
E_O	50%	Medium

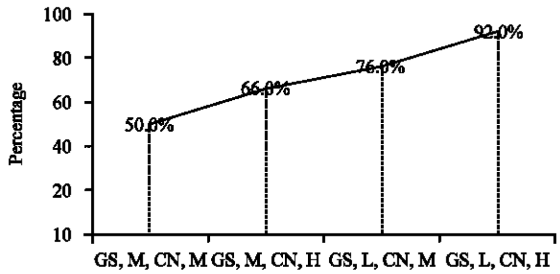

Fig. 11: Grid size and cycle time test G_S and C_N Medium: 50%, G_S Medium and C_N High: 66%, G_S Low and C_N Medium: 76%, G_S Low and C_N High: 92%

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
M*N=10*10 – Medium (M), X =10 Agents– Medium (M),						
A_{DT}	6	7	5	7	5	60.0%
M*N=10*10 – Medium (M), X =20 Agents – High (H)						
A_{DT}	8	6	8	7	5	68.0%
M*N=5*5 – Low (L), X =10 Agents – Medium						
A_{DT}	8	8	9	7	5	74.0%
M*N=5*5 – Low (L), X = 20 Agents – High (H)						
A_{DT}	10	10	8	10	9	94.0%

Figure 12 shows the effect of grid size and population density on observation of exceptional events.

Test No. 9: This test measures the effect of Cycle Time, C_N and Population Density of Domain Agents, A_C, on the observation process.

Settings

A_{DT}	10 Agents	Detectors Agents
A_C	X Agents	Test Low, Medium, High
C_N	Y cycle	Test Low, Medium, High
G_S	20*20	Low
O_L	1 Cell	Low
E_O	50%	Medium

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
X =2 Cycles – Medium(M), Y = 10 Agents – Medium (M)						
A_{DT}	5	5	3	5	2	40.0%
X =3 Cycles – High (H), Y = 20 Agents – High (H)						
A_{DT}	10	10	9	7	10	92.0%

Figure 13 shows the effect of cycle time and population density on observation of exceptional events.

Test No. 10: This test measures the effect of Domain (Grid) Size, G_S, Cycle Time, C_N and Population Density of Domain Agents, A_C, on the observation process.

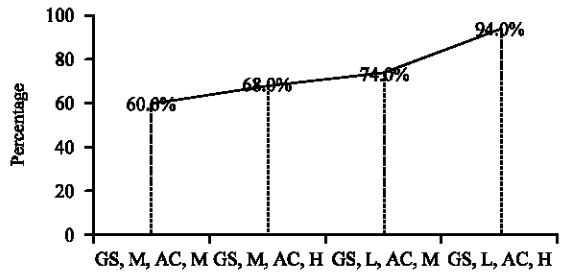

Fig. 12: Grid size and population density test G_S and A_C Medium: 60%, G_S Medium and A_C High: 68%, G_S Low and A_C Medium: 74%, G_S Low and A_C High: 94%

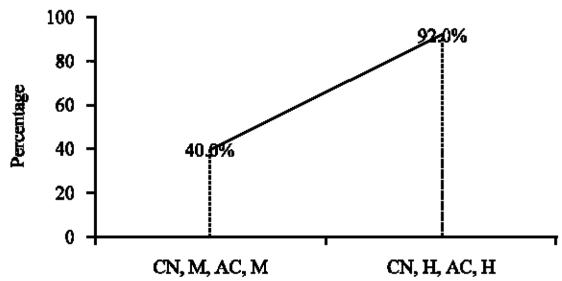

Fig. 13: Cycle time and population density test C_N and A_C Medium: 40%, C_N and A_C High: 92%

Settings:

A_{DT}	10 Agents	Detectors Agents
A_C	X Agents	Test Low, Medium, High
C_N	Y cycle	Test Low, Medium, High
G_S	M*N	Test Low, Medium, High
O_L	1 Cell	Low
E_O	50%	Medium

Run No.	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
M*N=10*10 – Medium (M), X =2 Cycles – Medium (M), Y = 10 Agents – Medium (M)						
A_{DT}	7	7	8	8	7	74.0%
M*N=5*5– Low (L), X =3 Cycles – High (H),, Y = 20 Agents – High (H)						
A_{DT}	9	10	10	9	10	96.0%

Figure 14 shows the effect of grid size, cycle time and population density on observation of exceptional events.

Summary of results of advanced testing: The aim of this study is to discover the most effective variables of the environment that enhance the observation process of detector agents. In general, when the agents have knowledge about the optimal cases for observation, they can achieve their goal faster and with higher accuracy.

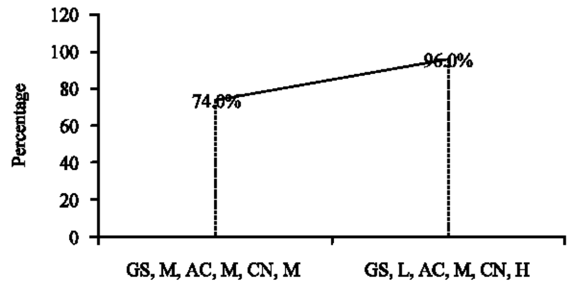

Fig. 14: Grid size, cycle time and population density test G_S and C_N and A_C Medium: 74%, G_S Low and C_N High and A_C High: 96%

Table 2: Summary of results of advanced testing

Settings	Result (%)	Optimal
G_S, M, O_L, M	48.0	No
G_S, M, O_L, H	60.0	No
G_S, L, O_L, M	66.0	No
G_S, L, O_L, H	78.0	No
G_S, M, C_N, M	50.0	No
G_S, M, C_N, H	66.0	No
G_S, L, C_N, M	76.0	No
G_S, L, C_N, H	92.0	Yes
G_S, M, A_C, M	60.0	No
G_S, M, A_C, H	68.0	No
G_S, L, A_C, M	74.0	No
G_S, L, A_C, H	94.0	Yes
C_N, M, A_C, M	40.0	No
C_N, H, A_C, H	92.0	Yes
G_S, M, A_C, M, C_N, M	74.0	No
G_S, L, A_C, H, C_N,H	96.0	Yes

As shown in Table 2, the results of advanced testing are as follows:

- It is discovered that there are three significant variables, which are population density of domain agents, A_C; cycle time, C_N and grid size, G_S
- The most optimal variables for agent to learn through observation are, G_S, L, A_C, H, C_N, H
- When the given cycle time (C_N) of agent is low (L), the optimal variables to learn are G_S, L, A_C, H
- When the population density of domain Agents (A_C) is low (L), then the optimal variables to learn are, G_S, L, C_N, H
- When the domain (grid) size (G_S) is high (H), then the optimal variables to learn are, A_C, H, C_N, H
- If the above variables are not available, agents should find at least one of the three significant variables that have strong positive effect. Otherwise the learning or identification success could be trivial

RESULTS AND DISCUSSION

As shown in Table 3, the results are attributed by four cases that are combined from three variables of population density of domain agents, A_C; cycle time, C_N and grid size, G_S.

Table 3: Significant settings of variables

	A_C			C_N			G_S			O_L			E_O			
Case	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	Optimal
G_S, L, C_N, H	*					*	*			*				*		*
G_S, L, A_C, H			*	*			*			*				*		*
C_N, H, A_C, H			*			*			*	*				*		*
G_S, L, A_C, H, C_N, H			*			*	*			*				*		*

Table 1 specifies all the variables' units and settings that are used in the tests. The tests are conducted based on the five elements that are proposed by Harrison *et al.* (2007) which are, initial settings that have are fixed in each test 1 to 10 described above; time constraint which represents the cycle time in this test; output determination, which is represented by Fig. 4-14; repetitions in each test which is represented by the number of runs and variations that are represented by fixing all variables except one to test its effect.

Figure 9 shows the initial results of testing single variable among fixed variables to discover the performance of each variable. The results shows the variables with strong positive effect, which are the grid size, 56%; the population density of domain agents, 50%; cycle time, 42% and observation limit, 42%. These results are supported by Savarimuthu *et al.* (2010a), who discovered that when the observation limit of agents increases, the agents identified the norms much faster and when the grid size increases, the number of agents which could identify the norms decreases. Xu *et al.* (2012) discovered that unlimited observation is more efficient but the observation is limited in more realistic scenarios. However, according to Sen and Airiau (2007), when the learning is based on the agents interaction (not on observation), the learning decreases when the population density increases. This is exactly the opposite of our finding in agents based on observation to learn. Savarimuthu *et al.* (2010a) proposed that observation in high population density increases the history log, which increases the probability of identifying the norms.

The variables are tested separately and this study extends the tests' settings by combining more than one variables to discover the significant settings of variables that lead to optimal environment for an agent to learn through observation of exceptional events. As shown in Table 2, there are four combinations of variables' settings that could offer optimal environment. These are:

Grid size, (G_S, Low); population density of domain agents, (A_C, High) and cycle time, (C_N, High), which show 96% performance.

Grid size, (G_S, Low) and population density of domain agents, (A_C, High), which record a 94% performance.

Two combinations of variables' settings produce 92% performance; grid size, (G_S, Low) and cycle time, (C_N, High) and population density of domain agents, (A_C, High) and cycle time, (C_N, High).

This study is significant in focussing on identifying the optimal environmental settings for an agent to learn through observation of exceptional events, unlike other studies that superficially discuss the environmental effects among other objectives (Sen and Airiau, 2007; Savarimuthu *et al.*, 2010b; Xu *et al.*, 2012).

CONCLUSION AND FUTURE PERSPECTIVE

The progress of research in norms learning and identification based on observing exceptional events offer a new capability in social learning for software agents. However, as shown in this study, the effectiveness of the learning process is dependent on the environmental variables and their values.

This study presents a simulation model to examine different environmental settings to discover the optimal cases of observing exceptional events within a social agents' group. To implement the simulation, a virtual scenario is created with the variables set to different values to examine different situations. In the scenario, two types of agents are created: detector agents and domain agents. The value of each variable is arbitrarily set to low, medium and high. Within the steps of the tests, some variables are excluded as they do not strongly influence the optimality of the environment for social learning. The results of tests show that a single variable of strong positive effect is inadequate to produce the optimal solutions. But when two or more variables of strong positive effect are tested, four optimal solutions are produced.

For future study, the virtual environment will be extended with a number of agents' societies which have the knowledge of optimal solutions, to further explore if those agents learn faster and with higher accuracy.

ACKNOWLEDGMENT

This project is sponsored by the Malaysian Ministry of Higher Education (MoHE) under the Exploratory Research Grant Scheme (ERGS) No. ERGS/f/2011/STG/UNITEN/02/8.

REFERENCES

Ahmad, A., M.Z.M. Yusoff, M.S. Ahmad, M. Ahmed and A. Mustapha, 2011. Resolving conflicts between personal and normative goals in normative agent systems. Proceedings of the 7th International Conference on Information Technology in Asia, July 12-13, 2011, Kuching, Sarawak, pp: 153-158.

Alberti, M., A.S. Gomes, R. Goncalves, J. Leite and M. Slota, 2011. Normative systems represented as hybrid knowledge bases. Proceedings of the 12th International Workshop on Computational Logic in Multi-Agent Systems, July 17-18, 2011, Barcelona, Spain, pp: 330-346.

Andrighetto, G., R. Conte and P. Turrini, 2007. Emergence in the loop: Simulating the two way dynamics of norm innovation. Proceedings of the Dagstuhl Seminar on Normative Multiagent Systems, March 18-23, 2007, Dagstuhl, Germany, pp: 1-29.

Axelrod, R., 1997. The Complexity of Cooperation. Princeton University Press, New Jersey, United States.

Bandura, A., 1997. Social Learning Theory. Prentice Hall, New Jersey.

Campos, J., M. Lopez-Sanchez and M. Esteva, 2010. A case-based reasoning approach for norm adaptation. Proceedings of the 5th International Conference on Hybrid Artificial Intelligence Systems, June 23-25, 2010, San Sebastian, Spain, pp: 168-176.

Centeno, R. and H. Billhardt, 2012. Auto-adaptation of open MAS through on-line modifications of the environment. Proceedings of the 10th International Conference on Advanced Agent Technology, May 2-6, 2011, Taipei, Taiwan, pp: 426-427.

Centeno, R., H. Billhardt and S. Ossowski, 2010. Inducing desirable behaviour through an incentives infrastructure. Proceedings of the 8th German Conference on Multiagent System Technologies, September 27-29, 2010, Leipzig, Germany, pp: 64-75.

Choi, J. and K.E. Kim, 2009. Inverse reinforcement learning in partially observable environments. Proceedings of the 21st International Joint Conference on Artificial Intelligence, July 11-17, 2009, Pasadena, CA., USA., pp: 1028-1033.

Conte, R. and F. Dignum, 2001. From social monitoring to normative influence. J. Artif. Soc. Social Simul., Vol. 4, No. 2.

Conte, R. and M. Paolucci, 2001. Intelligent social learning. J. Artif. Soc. Social Simul., Vol. 4, No. 1.

Harrison, J.R., G.R. Carroll and K.M. Carley, 2007. Simulation modeling in organizational and management research. Acad. Manage. Rev., 32: 1229-1245.

Hollander, C.D. and A.S. Wu, 2011. The current state of normative agent-based systems. J. Art. Societies Social Simulation, Vol. 14. No. 2.

Macy, M.W. and R. Willer, 2002. From factors to actors: Computational sociology and agent-based modeling. Annu. Rev. Sociol., 28: 143-166.

Maria, A., 1997. Introduction to modeling and simulation. Proceedings of the 29th Conference on Winter Simulation, December 7-10, 1997, Atlanta, GA., USA., pp: 7-13.

Natarajan, S., G. Kunapuli, K. Judah, P. Tadepalli, K. Kersting and J. Shavlik, 2010. Multi-agent inverse reinforcement learning. Proceedings of the 9th International Conference on Machine Learning and Applications, December 12-14, 2010, Washington, DC., USA., pp: 395-400.

Reeb, J.E. and S.A. Leavengood, 2003. Simulating a manufacturing system: An introduction. Performance Excellence in the Wood Products Industry Operations Research Oregon State University, Extension Service. http://ir.library.oregonstate.edu/xmlui/handle/1957/20322.

Sanders, W.H., 2005. Simulation basics. ECE/CS 541: Computer System Analysis. http://users.crhc.illinois.edu/nicol/ece541/slides/simulation-basics.pdf

Savarimuthu, B.T.R., S. Cranefield, M. Purvis and M. Purvis, 2010a. Norm identification in multi-agent societies. The Information Science Discussion Paper Series, University of Otago, New Zealand, pp: 1-36.

Savarimuthu, B.T.R., S. Cranefield, M.A. Purvis and M.K. Purvis, 2010b. Internal agent architecture for norm identification. Proceedings of the 5th International Workshop on Coordination, Organization Institutions and Norms in Agent Systems, September 2009, Turin, Italy, pp: 241-256.

Savarimuthu, B.T.R., S. Cranefield, M.A. Purvis and M.K. Purvis, 2010c. Obligation norm identification in agent societies. J. Artif. Soc. Social Simul., Vol. 13, No. 4.

Sen, S. and S. Airiau, 2007. Emergence of norms through social learning. Proceedings of the 20th International Joint Conference on Artificial Intelligence, January 6-12, 2007, Hyderabad, India, pp: 1507-1512.

Van Dyke Parunak, H., R. Savit and R.L. Riolo, 1998. Agent-based modeling vs. equation-based modeling: A case study and users? guide. Proceedings of the 1st International Workshop on Multi-Agent Systems and Agent-based Simulation, July 4-6, 1998, Paris, France, pp: 10-25.

Waldrop, M.M., 1993. Complexity: The Emerging Science at the Edge of Order and Chaos. Simon and Schuster, New York, USA., ISBN-13: 9780671872342, Pages: 380.

Xu, J. and M. van der Schaar, 2012. Social norm design for information exchange systems with limited observations. IEEE J. Selected Areas Commun., 30: 2126-2135.