Adopting Agent Oriented Methodology (AOM) For Modelling and Simulation in Epidemiology and Ecological Studies

Cheah Wai Shiang[1], Shane Nissom[1], Nurfauza binti Jali[1], Sim YeeWai[2]
[1] *Faculty of Computer Science & IT, Universiti Malaysia Sarawak*
[2] *School of Computing, University College of Technology Sarawak*
wscheah@unimas.my

***Abstract*—AOM (Agent Oriented Modeling) is a comprehensive and unified agent methodology for agent oriented software development. AOM methodology was proposed to aid developers with the introduction of technique, terminology, notation and guideline during agent systems development. Although AOM methodology is claimed to be capable of developing a complex real-world system, its potential is yet to be realized and recognized by the mainstream software community and the adoption of AOM is still in its infancy. Among the reason is that there are not much case studies or success story of AOM. This paper presents two case studies on the adoption of AOM for individual-based modelling and simulation. It demonstrates how the AOM is useful for epidemiology study and ecological study. Hence, it further validates the AOM in a qualitative manner.**

***Index Terms*—Agent Oriented Modeling; Agent Modeling; Agent Simulation; Software Development.**

I. INTRODUCTION

Agent methodology is introduced to cope with the development of the complex system. Agent methodology is a software engineering paradigm that makes use of agent paradigm to engineer complex systems. The agent can be described as a program or software that is autonomous, able to exhibit social interaction behaviour and can perform tasks in proactive or reactively manner [5]. Agent paradigm is suitable for complex system development because it can provide systems with problem solving or social interaction capabilities through means of agent abstraction [7].

To date, various agent methodologies have been introduced. AOM (Agent Oriented Modeling) methodology is one of agent methodologies introduced to aid developers with the introduction of technique, terminology, notation and guideline during agent systems development [5]. AOM methodology is suitable to cope with complex system development due to the nature of this methodology that it is able to describe a complex system with high level of abstraction, has less degree of ambiguity and can promote communications across different stakeholders [8]. However, there are not many case studies, be it empirical or heuristic studies or experience sharing to support the usefulness of AOM. Therefore, it is vital to conduct an investigation through case studies to validate AOM methodology. Validation and evaluation are needed to investigate to what extent this methodology is good at and also, to what extent it might fail, and importantly, highlight what extent AOM needs for improvement. Understanding the strength and weaknesses of agent methodologies can lead to develop better solutions and ultimately, increase the chance of methodology being successfully adopted by industrial players. Additionally, exploring agent methodology with many case studies from a range of domains can develop the confidence of industrial players to adopt agent methodology [10].

In this paper, the AOM is investigated as a technique for individual-based modeling and simulations. The individual-based modeling and simulation have received much attention in the area of epidemiology, criminal study, social study and etc. It would be great to identify the potential of AOM for individual-based modelling as IBM may be a potential “killer” application for agent technology. The paper presents two case studies on the adoption of AOM for individual-based modelling and simulation. It demonstrates the potential usage of AOM for epidemiology study and ecological study. Hence, it further validates the AOM in a qualitative manner. Section two presents background knowledge of Agent Oriented Methodology (AOM). Section three presents the case studies to be used to validate AOM. They are malaria transmission study and eutrophication study. In this section, a simplified modelling process is presented together with the NetLogo simulation. Section four presents the findings from the case studies. It describes the insufficiency of AOM towards NetLogo modelling and simulation. The related works of validating agent methodologies are presented in Section 5. The paper is concluded in Section 6.

II. AGENT ORIENTED METHODOLOGY (AOM)

Agent Oriented Methodology (AOM) is a methodology that is introduced for complex system development (the art). The AOM consists of three phases. They are conceptual domain modelling, platform independent design and modelling and platform specific design and modelling. The conceptual domain modelling was also known as motivation layer in which it models the system from an owner perspective. This involves understanding the goal of the system without further details on how the system is designed and implemented. The platform-independent design and modelling involve designing the system without looking into any particular implementation platform and language. The PSM layer is the lowest level of the system design. The design description at this layer allows the system to be deployed and executed in a particular environment like specific platform, hardware, technology, and architecture. The AOM consists of several modeling types like goal model, role model, organizational model, domain model, knowledge model, scenario model, interaction model and behaviour model to model a complex system. The modeler first models goals, roles, the organization and the domain which are relevant to the organization. Then, knowledge models, scenario models, interaction models and behaviour models are produced prior to system implementation by [5]. To date, the AOM has been used in collaborative learning [6], games development [3], rural ICT [4] projects, sustainability engineering [1].

III. RELATED WORKS

This section presents works to validate agent oriented methodology. Validation and evaluation are needed to investigate to what extent this methodology is good at and also, to what extent it might fail, and importantly, highlight what extent AOM needs for improvement. Understanding the strength and weaknesses of agent methodologies can lead to develop better solutions [18] and ultimately, increase the chance of methodology being successfully adopted by industrial players. Additionally, exploring agent methodology with many case studies from a range of domains can develop the confidence of industrial players to adopt agent methodology [19].

According to Sturm, the industry is still unaware of the potential of agent paradigm in conventional software development where object-oriented modeling (OOM) such as UML is still much favoured [21]. Among the reasons is the maturity of OOM which has become the de facto standard in the software industry. Another reason of lack of adoption in agent methodology is that agent methodology developers like to employ difficult terms (like agent autonomy, intelligence, for instance) when promoting agent methodology [20]. This approach may drive potential adopters to distant themselves from agent oriented software engineering. On the other hand, there are not many case studies, be it empirical or heuristic studies or experience sharing to support the usefulness of AOM [22]. Therefore, it is vital to conduct investigative studies to validate the AOM methodology.

Until now, validation and evaluation of AOM have taken place through case study and field testing only [23][24]. [23] have conducted a pilot testing on validating the use of AOM for flight operation management. In this case study, AOM was used to construct a complete requirements package. AOM was evaluated by engaging with stakeholders from aeronautical company to design and implement a prototype for aircraft turnaround simulator. [23] has reported that AOM requirements engineering approach is found useful for these stakeholders and as a consequence, they have adopted many parts of this method into their own requirements engineering process models.

Meanwhile, in the emergency systems case study [25], Miller, T has developed People Oriented Software Engineering (POSE) models, through the extension of AOM to capture human emotional needs. POSE models were evaluated through controlled user study (with 20 participants from a range of backgrounds) and a case study of an emergency system for old people (with nine older people). The evaluation was performed through test questions, participant survey and interviews. [25] has reported that these participants in controlled user study are comfortable with interpreting and modifying POSE models and view the addition of emotions to the requirements as a positive step for software engineering. On the other hand, [25] has reported that users are satisfied with the prototyped emergency system that addresses key emotional goals of the older person as compared to the existing system.

[6] reports the usability of interactive patterns for designing and developing applications for collaborative learning. Behavior model from AOM was used to design interaction patterns of single display groupware (SDG) for collaborative learning. The pattern styles designed from behaviour model were implemented in RimbaIlmu, an SDG applied as a case study. A field study was conducted among eight UNIMAS FCSIT students to evaluate the applicability of patterns in developing an SDG application. Meanwhile, patterns are able to reduce students' time in designing SDG. Additionally, patterns are also able to support better code understanding and making code modification or maintenance easier.

[24] described the use of AOM in the field of computer animation. Murdoch has developed MK I production model to produce 3D character animations (at requirements stage only). AOM goal model is used as the basis for MK I production model creation. MK I production model was entrenched in 12 student animators' project to inform animators of the production process and their expected activity. This model was used by project management team as a basis to evaluate animation and to convey rating progress and achievement. Based on the evaluation from students and project management team [24], has reported that further investigation and iterative development of the model is required to improve user engagement in the production of animation. Nevertheless, the novel use of AOM has presented a forward in the communication and engagement with the production process of 3D character animation, according to [24].

In summary, qualitative and quantitative approaches are adopted to evaluate agent methodologies. Meanwhile, case studies are conducted to overcome insufficient practices from most of the agent methodologies as stated by [22].The process of evaluation is important to advance agent methodologies from research to industrial practice. With these evaluation results, industry players can select the right methodology when implementing a socio-technical system, especially when platforms are available and methodologies are already matched with them. While there are many agent methodologies being introduced in agent research, work on evaluating and validating agent methodologies is still an active research [27].

IV. CASE STUDIES

Two case studies are developed based on AOM. They are malaria transmission modelling and simulation and eutrophication study.

Malaria transmission model is used to simulate malaria outbreak which can aid in the creation of a disease control systems that are appropriate for vector control units and health ministries. In this paper, the model is built upon the work from [9]. Malaria transmission model describes spread of malaria in terms of the flow of humans and mosquitoes between two states: Susceptible and Infectious. Susceptible represents the state of human and mosquito that are not infected yet but vulnerable to infection. Infectious represents the state of human and mosquito that have been infected with the disease and are capable of spreading the disease to susceptible.

Figure 1 shows the compartmental model of malaria for human and mosquito. This model implicitly describes the rule of interaction that controls the “flow of” malaria transmission between these four compartments (S_H, I_H, S_M and I_M).

In this diagram, mosquito and human are allowed to transmit disease when they are in opposing health states, which is represented by two red arrows that connect Infectious human (I_H) to Susceptible mosquito (S_M), and Susceptible human (S_H) to Infectious mosquito (I_M). The outcome of such interaction is the transition of human or mosquito health state from initially Susceptible to Infectious state. This is fulfilled through calculation of “rate of infection”, which is highlighted in Figure 1 as A and B.

In Figure 1, infection is assumed to be temporary in human. In this case, the outgoing arrow with from IH compartment is shown leading back to SH compartment, meaning that at some point, humans are able to recover from malaria. While humans can recover from disease, the same humans can get infected and recovered repeatedly without dying in the process – again, through that same outgoing arrow with r attribute linking SH compartment. Unlike humans, mosquitoes must become permanently Infectious until their death. This is true since the only outgoing arrow from IM compartment is death rate, as shown in Figure 1.

Aside from disease transmission cycle, the incoming arrows and outgoing arrows for Susceptible human (S_H), Infectious human (I_H), Susceptible mosquito (S_M) and Infectious mosquito (I_M) compartments in Figure 1 also present other behaviours of humans and mosquitoes. For instance, the mosquito recruitment rate Λ is represented as the incoming arrow for S_M, compartment, meaning that this behaviour leads to the introduction of new susceptible mosquitoes to the environment. Whereas for death rates (d_H and d_M), arrows are pointed as outgoing from S_H, I_H, S_M and I_M compartments, denoting that these behaviours will lead to the removal of humans and mosquitoes from the environment upon death. The death rates here are related to mosquito and human lifespan.

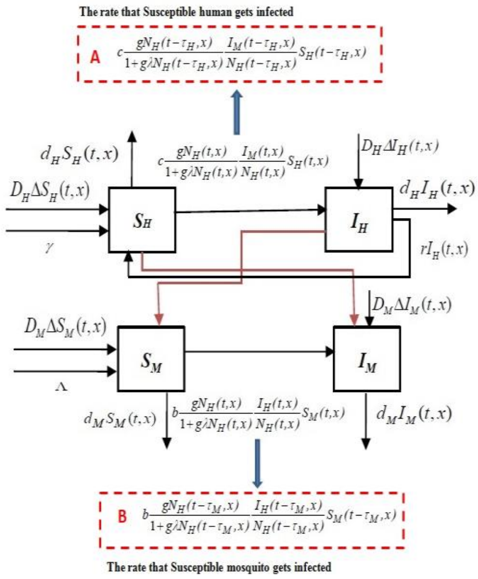


Figure 1: The compartmental model on the transmission of malaria for human and mosquito

As mentioned before AOM starts with goal modelling. This is followed by a role model, organization modelling, domain model, interaction modelling, behaviour modelling, knowledge modelling.

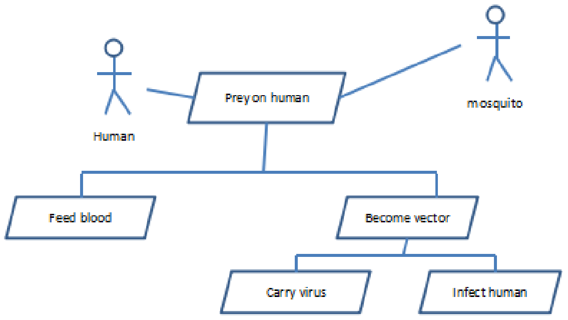


Figure 2: Goal model for malaria transmission

Figure 2 shows the goal model for malaria transmission. It models the main goal for malaria transmission is to ‘prey on human’. In order to achieve the goal, human and mosquito are needed. Figure 3 presents the domain model for the malaria transmission. It consists of domain entities that are required in malaria transmission.

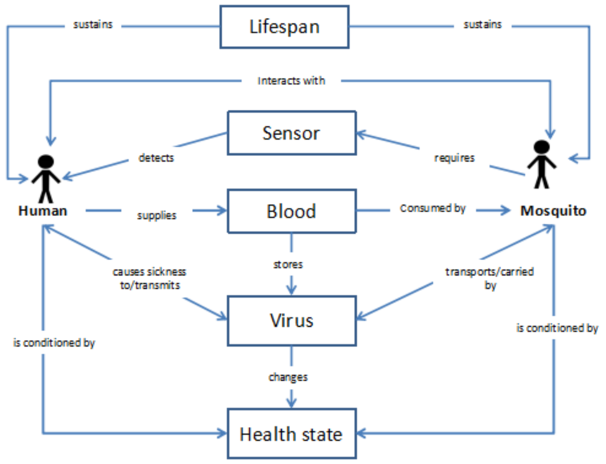


Figure 3: Domain model for malaria transmission

Table 1
Configuration value of NetLogo for Experiment 1

	NetLogo attributes	Configuration value	Remark
Population size	Initial number of Infectious human agents (IH)	22	It shows the density of infectious human population.
	Initial number of Susceptible human agents (SH)	3	It shows the density of susceptible human population
	Initial number of Susceptible mosquito agents (IM)	100	It shows the density of susceptible mosquito.
	Initial number of Infectious mosquito agents (IM)	200	Number of initial IM = 2 * area, 100 = 200 mosquitoes.
Human agent attributes	Incubation period in mosquito	5	
	Sick duration in human	5 days	Number of days taken to become sick.
	Human lifespan	24000 days	The death rate of human.
	Movement rate of human	3	For instance, setting this variable as “3” allows this agent to take three forward steps per day – assuming that one step in NetLogo is equivalent to 1 km.

Table 2
Configuration Value of NetLogo for Experiment 2

NetLogo attributes	Conf value	Remark
Initial number of Infectious mosquito agents (I_H)	0	$I_M = -2x + 2$ and Experiment 2 chooses <u>$x = 1$</u>, therefore, density of $I_M = -2(1) + 2 = 0$. Number of initial I_M = 2 * area, 100 = <u>0 mosquitoes</u>.

*Population size, human and mosquito attribute and simulation configuration: Same Experiment 1 configuration values for initial number of Susceptible human, Infectious human and Susceptible mosquito agents, incubation period in mosquito, incubation period in human, human sick period, lifespan for human and mosquito, reproduction rate for human and mosquito, movement rate for human and mosquito, mosquito search radius, simulation radius and carrying capacity.

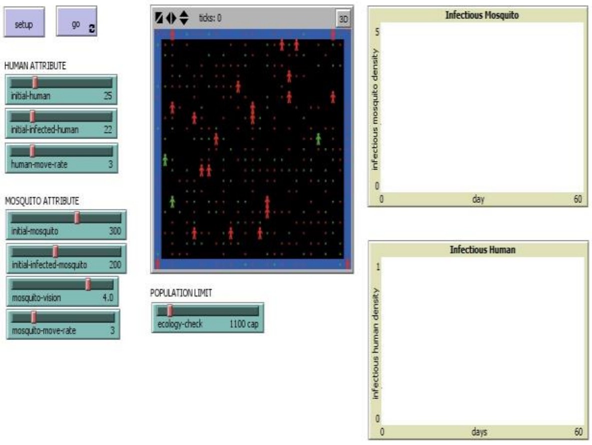
Table 1 describes the configuration for Experiment 1 (has 200 Infectious mosquito agents initially). Table 2 describes the configuration for Experiment 2 (has no initial Infectious mosquito agent). A new simulation variable called carrying capacity is proposed in this step to optimize the speed of ABMS. Optimization can be achieved through the setting of population limit in the simulation where carrying capacity can halt "recruitment" of a new agent in the environment when current population size has reached the carrying capacity limit.

Once configuration values have been decided for simulation, this step concludes ABMS development with the design of NetLogo user interface. First, plots are created to display the graphs when running agent-based malaria transmission simulation. Then elements like buttons and sliders are created to allow users to interact with NetLogo malaria transmission simulation (i.e. using sliders for making adjustments on a number of initial Infectious mosquito agents and buttons to setup agents and run/stop simulation).

In this step, two plots were created for agent-based malaria transmission model and simulation. These plots display behaviour of the spread of malaria as (a) Infectious humans vs. time and (b) Infectious mosquitoes vs. time. Each plot visualizes the spread of Infectious agents in terms of density vs. time. The area was assumed to be 100km2 and therefore, density can be formulated in NetLogo plot command by dividing a number of Infectious human or mosquito agents with this area value. For instance, plots will show 5 Infectious mosquitoes as Infectious mosquito density of 0.05.

The interface of NetLogo malaria transmission model and simulation can be seen in Figure 4.

This interface has two buttons – "setup" for setting up experiments and "go" for running simulation. Sliders are used for configuring population size of human and mosquito (i.e. initial-human), configuring attributes of human and mosquito (i.e. mosquito-move-rate) and configuring carrying capacity (ecology-check). Two plot interfaces are shown in this simulation – Infectious Mosquito (to display population density of Infectious mosquitoes vs. time) and Infectious Human (to display population density of Infectious humans vs. time). The interface in the middle is the 2D view for a visual representation of NetLogo mosquito and human agents (turtles) that are behaving in a virtual environment.

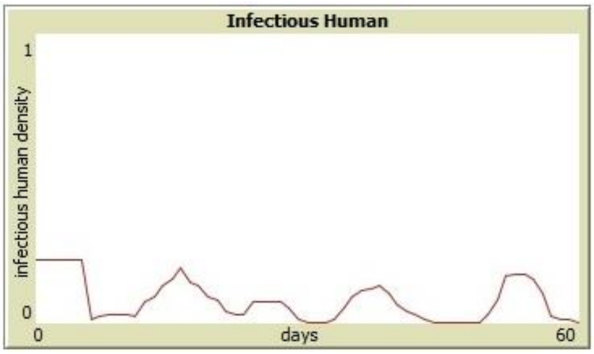


initial-human 25
initial-infected-human 22
human-move-rate 3
initial-mosquito 300
initial-infected-mosquito 200
mosquito-vision 4.0
mosquito-move-rate 3
ecology-check 1100 cap

Figure 4: The interface of NetLogo malaria transmission model and simulation

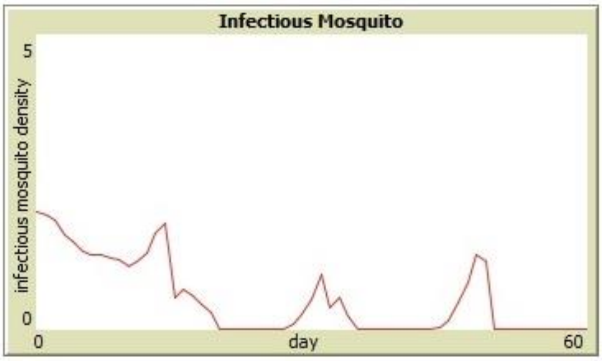
A. NetLogo Experiment 1 and Experiment 2 Results

This section shows the results produced by agent-based malaria transmission model and simulation when using two different experiment configurations. Figure 5 presents the outcome of ABMS through Experiment 1 and Experiment 2 configurations in NetLogo.



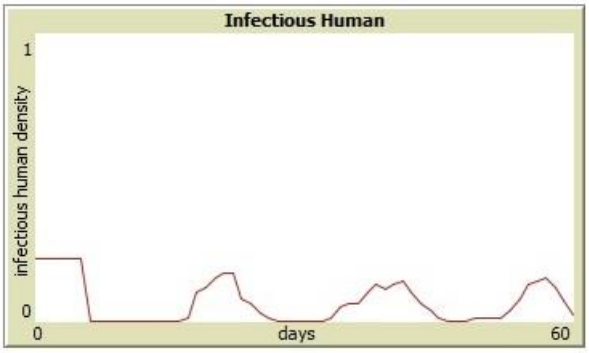
Panel-A

(a) The spread of Infectious Human agents in 60 days for Experiment 1

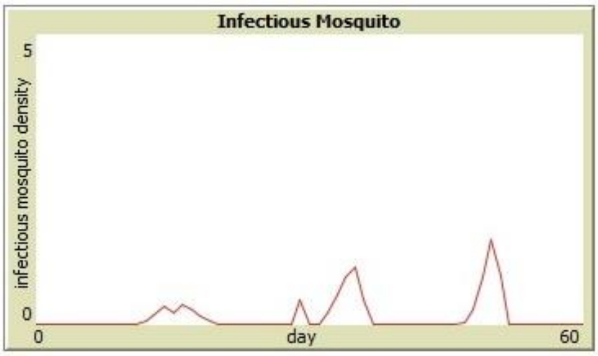


Panel-B

(b) The spread of *Infectious Mosquito agents* in 60 *days for Experiment 1*



Panel-C
(c) The spread of Infectious Human agents in 60 days for Experiment 2



Panel-D
(d) The spread of Infectious Mosquito agents in 60 days for Experiment 2

Figure 5: Results of the NetLogo simulation for experiment 1 and experiment 2

Figure 5 shows the behaviour of the spread of Infectious mosquitoes and humans agents in NetLogo after 60 days. Panel-A and Panel-B illustrate the spread of Infectious humans and mosquitoes in NetLogo Experiment 1 when there are 200 initial Infectious mosquito agents). Meanwhile, Panel-C and Panel-D illustrate the spread of Infectious humans and mosquitoes in NetLogo Experiment 2 when there is no initial Infectious mosquito agent.

Results have shown that modification of the number of initial Infectious Mosquitoes can lead to a different pattern of spread of Infectious humans and mosquitoes in NetLogo. Curves become higher and wider when the number of Infectious mosquitoes is increased. For instance, in first 20 days of simulation, the curve of density population of Infectious mosquito in Panel-B appears to be larger compared to Panel-D’s curve of density population of Infectious mosquito.

Case study 2:

Eutrophication is an ecosystem response, caused by over-enrichment of water by input nutrients, notably phosphorus, from agricultural activities and it is a widespread and growing problem in lakes, rivers and coastal oceans [11]. According to [11], mitigation of eutrophication is impacted not only by the complexity of lake but also by phenomena of complex human behaviour that causes non-point pollution. [11] has noted that successful mitigation of eutrophication requires insights on the behaviour of farmers causing non-point pollution. A similar configuration process is conducted for this case study. Due to the space limit, we only present the interface and results of the simulation.

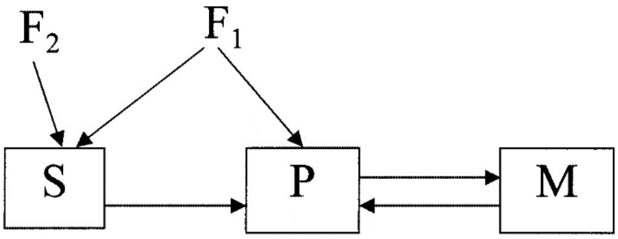


Figure 6: Lake model diagram

A flow of interaction is designed by [28] to describe how these layers of environment and farmers are linked together in Lake Model, which can be seen in Figure 6. Figure 6 describes this link in terms of the flow of phosphorus (the arrows) from farmers to the environment and from one environment layer to another.

In this diagram, the use of phosphorus in an intensive manner (F1) by farmers leads to a direct flow of phosphorus into the water and thus, increases the concentration of phosphorus in water, P (at a rate of L_1). At the same time, this practice also leads to increase of the concentration of phosphorus in the soil, S (at a rate of a1). When conservative approach (F2) is adopted by farmers, the concentration of phosphorus in the soil decreases (at a rate of a2). Some part of phosphorus in soil flows into the lake due to weather influence (at a rate of SF) and therefore, the concentration of phosphorus in water is increased. Some part of phosphorus in water sinks to the bottom of the lake (through sedimentation rate, s) and remains in the mud until phosphorus concentration in water reaches high level that phosphorus in mud gets recycled (through recycling rate, r).

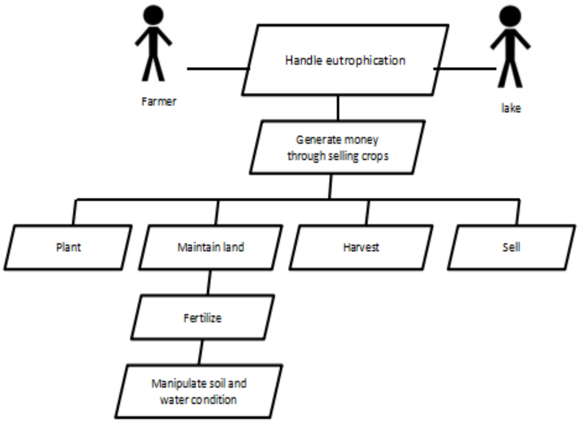


Figure 7: Goal model for lake model study

We adopt the AOM in modelling the lake model in the second case study. Figure 7 shows the goal model for eutrophication study. Here, farmer and lake and involved to achieve the main goal to 'handle eutrophication'. From the study, the eutrophication is influenced by income generation in which it models as the sub-goal of 'handle eutrophication'. In order to achieve the sub-goal of 'generate money', the farmer needs to achieve its goal to 'plant', 'manipulate land', 'harvest' and 'sell crop'.

Figure 8 shows the NetLogo model for eutrophication study. In the model, sliders are used for configuring farmer, lake and experiment attributes. Six plot interfaces are shown in this simulation to display the level of phosphorus inputs, the phosphorus concentration in water, mud and soil and farmers' returns versus time. The interface in the middle is the 2D view for a visual representation of NetLogo farmers and lake agents that are behaving in a virtual environment. During the runtime, the plots model the farmers' behaviour in regard to phosphorus inputs to the lake.

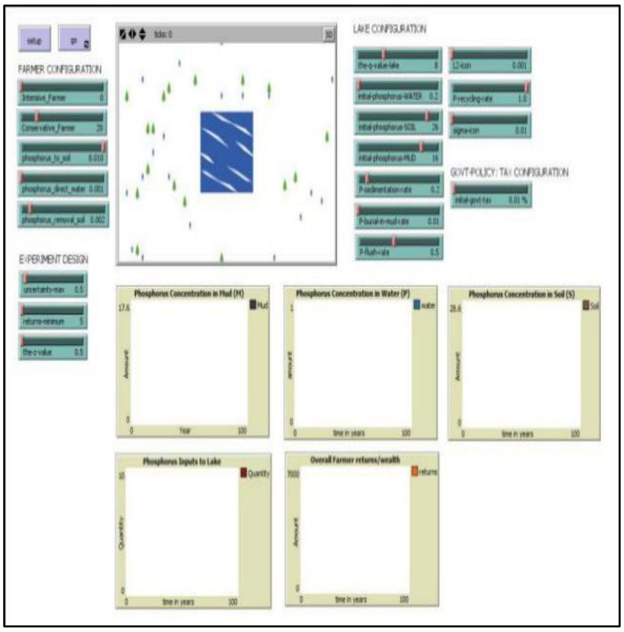

Intensive_Farmer 0
Conservative_Farmer 20
phosphorus_to_soil 0.010
phosphorus_direct_water 0.001
phosphorus_removal_soil 0.002
uncertainty-max 0.5
returns-minimum 5
the-z-value 0.5
the-q-value-lake 8
initial-phosphorus-WATER 0.2
initial-phosphorus-SOIL 26
initial-phosphorus-MUD 16
P-sedimentation-rate 0.2
P-burial-in-mud-rate 0.01
P-flush-rate 0.5
12-icon 0.001
P-recycling-rate 1.0
sigma-icon 0.01
initial-govt-tax 0.01 %

Figure 8: The interface of NetLogo Lake Model

V. FINDING FROM THE CASE STUDIES

In this paper, two case studies are presented in order to validate AOM in real-world practices. From the studies, some insufficiency of AOM has been identified. The insufficient can be ranged from requirement elicitation to the notation of agent models and from agent modelling to NetLogo simulation.

Missing stakeholder interaction in AOM- AOM starts with goal modelling. Hence, there is a missing stakeholder interaction to identify the model given. As a result, the modeler will build the model based on its assumptions and trial from errors. Although we are able to build the NetLogo simulation, lots of assumptions have been made. Also, it is hard to validate the model as some information may be implicitly decided by the agent modeler.

Missing natural artifacts in AOM – AOM is used to model complex system which is originally perceived system that contains many interacting parts and many interactions among actors (people, devices and software agents) in the socio-technical system [8]. However, the natural artifacts like non-living creature and environment have not been addressed. This leads to a difficulty to model a mathematical modelling for a real-world problem.

Communication difficulty-The AOM is built based on the agent paradigm. Mapping the mathematical concepts into agent modelling and simulation is not a trivial task. One must need to understand the mathematical terminology prior map into the agent paradigm. As AOM is started with goal modelling, much of the problem description and formulation are still implicit to the domain expert.

Notation problem in goal model – This problem is due to the fact that AOM role notation is explicitly human and for human-oriented activities only. In AOM, the "stickman" figure is the only notation available to represent roles in goal model. It would make no sense to match this figure to any creatures – for instance, mosquito, wolf, sheep, fish, bird and etc. This problem is entrenched by HOMER's adoption of the organization as elicitation metaphor. As a result, a role will always be associated with a function assumed by a person or some employment position, i.e. owner, buyer, seller, assistant, government, secretary, manager, domain expert, and so on. In order to overcome this insufficiency, a non-human notation has been introduced in agent modelling.

Lack of model transformation from agent models to NetLogo: As AOM is designed to be platform independent [5], working on simulation model is based on modeler interest and skill. As a result, it is hard to translate the agent models into simulation model especially for the novice developer.

VI. RECOMMENDATION

We present our finding upon working on the two case studies. From the experiments, we propose some recommendations to enhance the AOM for mathematical modelling.

1) *The needs to have a platform to bridge the communication gaps between the mathematical modeler and agent modeler.*

Hence, HOMER is a good choice to explore. HOMER, which is part of AOM, supports this notion by using organizational metaphor to elicit requirements. However, in a real-world problem, there are characteristics like aspects of human societies, animal populations, living cells or even non-living things like gas molecules situated in some type of environment where they can self-organized, interacting with each other and can even produce emergent behaviours. For example, in HOMER[2], an elicitation technique in AOM, it is difficult to agree to what extent this organizational metaphor in HOMER is suitable to characterize complex system elements from domains such as biology, environmental science, chemistry and climate change where human, non-human organism and environmental entities make up the whole complex system. Thus, it would be awkward to classify non-human entities in mathematical models with any organizational terms – imagine if AOM is used to represent some organism type with position title or referring to an ecosystem as a company! In order to overcome this limitation, HOMER extension is needed. The extension involves the following suggestions.

2) *Calibration of HOMER questions for requirements elicitation*

As mentioned before, AOM begins with HOMER to elicit requirements. Therefore, new requirements have to be assimilated to all questionnaire expressions in HOMER to support the collection of agent contexts derived from mathematical models. The following describes what becomes of HOMER when its framework is calibrated with above-mentioned requirements.

3) *Trigger of elicitation is altered*

Already, the expression in HOMER Question 1, "If you were to hire more staff to handle your current problem..." constraints this elicitation scope to address staff recruitment as the problem statement. In the opening of computational science elicitation between agent modeler and domain experts, an open-ended question should be used to inquire information about any kind of problem statement, since CS involves a bigger spectrum of issues ranging from studies in epidemiology, microbiology, economics, evolution, etc.

4) *Substitution of organizational terms with comparable terms suitable for the complex system.*

To allow AOM in the domain of CS, the notion of a role in HOMER needs to be redefined to portray the functions performed by both human and non-human entities (in which AOM originally does not perceive). Meanwhile, it would be sensible to substitute these organizational terms with coherent terms that can be understood and related to any domain experts. Furthermore, the viewpoints mentioned must be preserved during the extension process.

5) *Additional elicitation layer in HOMER for environmental elicitation*

Coupling of social and environmental criteria demands the knowledge about the environment to solve the model problem but these were not addressed by original HOMER. Environmental objects such as lake, river and land patch in an ecosystem, can be treated simply as unique “artifacts” that are somewhat able to influence the behaviour of social entities and therefore HOMER should be adapted to input this information. For example, in [28], farmer’s fertilization behaviour can affect the condition of the lake (one of artifact in the environment) and in turn, the quality of water can impact farmers’ decision (i.e. stop polluting lake when noticing a reduction of yearly income).

6) *Extend goal model to include additional role notations for non-human entities;*

Role type will be sketched according to agent’s biological classes rather than constrained to single notation type (the stickman figure). This is important to distinguish human roles from non-human roles in goal models. Therefore, original “stickman” figure must be limited to representing human-oriented roles such as farmers, villagers, employees, etc. whereas animal roles will be represented with animal symbols related to it – for instance, mosquito symbol is used to represent the role of mosquito as a vector that is tasked with infecting people with malaria.

7) *Introducing mediator agent at the design phase.*

Common agent, or simply known as an agent, is constrained to live entities types like people or creature that have social behaviour. On the other hand, mediator agent is a special class of agent that is purposed for bridging the gap between environment and agents situated in it. Mediator agent is used to handle tasks that are environment-related since agents do not have the control over environmental properties. As environment is one of important aspect in mathematical models in science that can influence the social behavior of populations, mediator agent can support this notion by handling intermediary tasks such as controlling flow of natural resources from agent to the environment or vice versa and reports the level of environmental quality to agents for impacting their decision-making ability.

VII. CONCLUSION

This paper presents a validation of Agent Oriented Methodology (AOM) through case studies. From the study, we address that AOM is insufficient to model epidemiology domain and environment model and suggest the extension of AOM. In future, more works need to explore on the extension of AOM in various NetLogo domain. Also, quantitative evaluation on AOM among users is worth to explore. This will provide more scientific evidence to the industry and another research committee. Indirectly, it can promote the AOM to a wider audience.

REFERENCES

[1] L. CheeWyai, C. WaiShiang, A. Kabir Chowdhury and C. Gulden, “Engineering sustainable software: A case study from offline computer support collaborative annotation system,” 9th Malaysian Software Engineering Conference (MySEC), pp. 272-277, 2015.

[2] C. WaiShiang, S. YeeWai, S. Nizam, and L. CheeWyai, “Agent Oriented Requirement Engineering for Lake Mathematical Modelling: Preliminary Study,” Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 8(2), pp. 5-10, 2016.

[3] C. WaiShiang, J.J. Meyer and K. Taveter, “Agent-Oriented Methodology for Designing Cognitive Agents for Serious Games,” Engineering Multi-Agent Systems, 39, 2016.

[4] C. WaiShiang, A.A. Halin, M. Lu. and L. CheeWhye, “Long Lamai Community ICT4D E-Commerce System Modelling: An Agent Oriented Role-Based Approach,” The Electronic Journal of Information Systems in Developing Countries, 75, 2016.

[5] L. Sterling and K. Taveter, “The art of agent-oriented modeling”, MIT Press, 2009.

[6] C. WaiShiang, E. Mit, and A.A. Halin, “Shared Single Display Application:An Interactive Patterns Approach,” International Journal of Software Engineering & Its Applications,9(2), pp. 233-250, 2015.

[7] V. Seidita, and M. Cossentino, M., “Supporting Autonomy in Agent Oriented Methodologies,”. In Intelligent Interactive Multimedia Systems and Services, pp. 599-608, Springer.

[8] T. Miller, B. Lu, L. Sterling, G. Beydoun, and K. Taveter, “Requirements elicitation and specification using the agent paradigm: the case study of an aircraft turnaround simulator,” IEEE Transactions on Software Engineering, 40(10), pp.1007-1024, 2014.

[9] C. Kon, and J. Labadin, “Reaction-diffusion generic model for mosquito-borne diseases,” 8th IEEE International Conference on information technology in Asia (CITA), pp. 1-4, 2013.

[10] M. Winikoff, “Future directions for agent-based software engineering,” International Journal of Agent-Oriented Software Engineering, 3(4), pp.402-410, 2009.

[11] M.A. Janssen, “An exploratory integrated model to assess management of lake eutrophication. Ecological modelling”, 140(1), pp.111-124, 2001.

[12] J. Barbosa, P. Leitão, E. Adam and D. Trentesaux, “Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution,” Computers in Industry, 66, pp.99-111, 2015.

[13] Z. Shen, H. Yu, C. Miao, S. Li, and Y. Chen, Y., “Multi-agent system development made easy,” In Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16), 2016.

[14] F. Zambonelli, F., “Towards a General Software Engineering Methodology for the Internet of Things,”. arXiv preprint arXiv:1601.05569, 2016.

[15] J. Sudeikat, L. Braubach, A. Pokahr, and W. Lamersdorf, “Evaluation of agent–oriented software methodologies–examination of the gap between modeling and platform,” In International Workshop on Agent-Oriented Software Engineering, pp. 126-141, 2004.

[16] A. Sturm, and O. Shehory, “A framework for evaluating agent-oriented methodologies,” 5th International Bi-ConferenceWorkshop of Agent-Oriented Information Systems, Melbourne, Australia. 3030: pp. 94-109, 2004.

[17] A. Sturm and O. Shehory, “Agent-oriented software engineering: revisiting the state of the art,” In Agent-Oriented Software Engineering, pp. 13-26, Springer Berlin Heidelberg, 2014.

[18] J.L.C. de Moraes, W.L. de Souza, L.F. Pires and A.F. do Prado, “A methodology based on openEHR archetypes and software agents for developing e-health applications reusing legacy systems,” Computer Methods and Programs in Biomedicine, 134, pp.267-287, 2016.

[19] Lin, C. E., Kavi, K. M., Sheldon, F. T., Daley, K. M., & Abercrombie, R. K. (2007, January). A methodology to evaluate agent oriented software engineering techniques. In System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on (pp. 60-60). IEEE.

[20] Winikoff, M. (2009). Future directions for agent-based software engineering. International Journal of Agent-Oriented Software Engineering, 3(4), 402-410.

[21] DeLoach, S. A. (2009). Moving multi-agent systems from research to practice. International Journal of Agent-Oriented Software Engineering, 3(4), 378-382.

[22] Sturm, A., & Shehory, O. (2014). Agent-oriented software engineering: revisiting the state of the art. In Agent-Oriented Software Engineering (pp. 13-26). Springer Berlin Heidelberg.

[23] Miller, T., Lu, B., Sterling, L., Beydoun, G., & Taveter, K. (2014). Requirements Elicitation and Specification Using the Agent Paradigm: The Case Study of an Aircraft Turnaround Simulator. IEEE Transactions on Software Engineering, 40(10), 1007-1024.

[24] Murdoch, S. (2016). Agent-Oriented Modelling in the Production of 3D Character Animation. Studies in Australasian Cinema, 10(1), 35-52.

[25] Miller, T., Pedell, S., Lopez-Lorca, A. A., Mendoza, A., Sterling, L., & Keirnan, A. (2015). Emotion-led Modelling for People-Oriented Requirements Engineering: The Case Study of Emergency Systems. Journal of Systems and Software, 105, 54-71.

[26] Sudeikat, J., Braubach, L., Pokahr, A., & Lamersdorf, W. (2004, July). Evaluation of agent–oriented software methodologies–examination of the gap between modeling and platform. In International Workshop on Agent-Oriented Software Engineering (pp. 126-141). Springer Berlin Heidelberg.

[27] De Moraes, J. L. C., de Souza, W. L., Pires, L. F., & do Prado, A. F. (2016). A methodology based on openEHR archetypes and software agents for developing e-health applications reusing legacy systems. Computer Methods and Programs in Biomedicine, 134, 267-287.

[28] Janssen, M. A. (2001). An Exploratory Integrated Model to Assess Management of Lake Eutrophication. Ecological Modelling, 140(1), 111-124.