Journal of Theoretical and Applied Information Technology
20th April 2016. Vol.86. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TOWARDS A SELF-ADAPTIVE AGENT-BASED SIMULATION MODEL

[1]YIM LING LOO, [2]ALICIA Y.C. TANG, [3]AZHANA AHMAD, [4]AIDA MUSTAPHA

[1] Centre for Agent Technology, College of Information Technology, Universiti Tenaga Nasional, Jalan Ikram-Uniten, 43000 Kajang, Selangor, Malaysia.

[2] Dr., Department of Systems and Networking, College of Information Technology, Universiti Tenaga Nasional, Jalan Ikram-Uniten, 43000 Kajang, Selangor, Malaysia.

[3] Dr., Department of Software Engineering, College of Information Technology, Universiti Tenaga Nasional, Jalan Ikram-Uniten, 43000 Kajang, Selangor, Malaysia.

[4]Dr., Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia.

Email: [1]yimling1st@yahoo.com, [2]aliciat@uniten.edu.my, [3]azhana@uniten.edu.my, [4]aidam@uthm.edu.my

ABSTRACT

Agent-based simulation (ABS) modelling has been a widely applied approach for simulating domain-specific phenomena. Currently, parameters and environments are simulated by a domain-specific model that is strictly used proprietarily by the ABS model developer. This causes inflexibility towards extension of the developed ABS model, which will further result in difficulties for validation and verification of the robustness and reliability of the ABS model. To address this issue, this paper proposes a self-adaptive ABS model that is capable of modelling cross-domain phenomena by selecting the required parameters based on the environment. The capability to self-adapt will allow the model to be easily extended and replicated. The self-adapt capability is enabled by a governing algorithm within the model and is conceptually illustrated through a case study of crime report process ABS modelling.

Keywords: *Self-adaptive Model, Agent-based Simulation Model, Processes Simulation, Extensive Model, Model Reuse and Replication*

1. INTRODUCTION

Agent-based simulation (ABS) is an approach for modelling dynamic environments with interactive individuals. This approach is known to best model different real-life phenomena at the highest level of imitation [1]-[5]. ABS modelling enable testing of real-life phenomena without the need of setting up costly experiments. For example, consider a real-life phenomenon of three individuals interactively communicating with one another in an office. Instead of acting out the phenomena in a series of repeating experiments to refine the events. ABS can be used to model the interactions of three agents in an office environment as shown in Figure 1.

Based on the model, ABS can be used to precisely simulate any specific phenomena of interest. Modelling a specific phenomenon such as the interaction of three individuals in an office is known as ad-hoc or domain-specific ABS modelling [1],[5]-[7]. Although ad-hoc or domain-specific ABS modelling has been growing to cater simulations of new different phenomena, such ABS models carry several limitations. The results from a domain-specific ABS model may be inaccurate due to phenomena simulated and tested being too specific [8]-[10]. This is essentially the problem of validation and verification of results from the ABS model. Second, the models are not easily extensible due to proprietary issues [7],[11],[12], which in turn will cause excessive cost and resource for developing a new ABS model. This further causes

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the problems of replication and reuse of ABS models hence affecting the robustness of the ABS model [13],[14].

To address the issues of validation and verification and replication and reuse in ad-hoc or domain-specific ABS modelling, this paper proposes a new conceptual model of a self-adaptive ABS. The remainder of this paper is organized as follows. Section 2 discusses research efforts to date on extensible ABS model and the differences with the proposed self-adaptive ABS model. Section 3 presents the self-adaptive ABS model. Section 4 discusses a case study applying the self-adaptive ABS model. Finally, Section 5 concludes with indication for future directions.

2. RELATED WORKS: ADDRESSING THE GAPS

Limitations of being ad-hoc or domain-specific, difficulty in validation, verification, extensibility and customizability as well as replication and reuse are becoming more evident. A number of research in ABS modelling has been found to address the limitations of ad-hoc or domain-specific ABS [11],[12],[15]. The focus has been on developing a generic simulation platform or framework that allows plug-ins of different ABS models for non-domain-specific simulation as shown in Figure 2(a).

As illustrated in Figure 2(a), the generic ABS framework receives input from three ABS models (A, B, C) of different domains, namely Registrar of University, Transportation and Security. The framework is then used to simulate for achieving a specific objective. For instance, DYNAMOD [11] allowed different digital business ABS models to be plugged-in to the framework in order to simulate a common market phenomena. Bagneris [12], plugged-in different financial marketing ABS for financial predictions of different domains to address limitations of replication and reuse. Likewise, Bosse et al. [7], simulate ambient intelligence with a generic ABS framework tested with three different environment.

DANUBIA [15] also used a generic ABS framework for evaluating sustainability of future water resources management alternatives. TransiTUM [8] used a generic ABS framework for multi-scaled coupling pedestrian simulation, which emphasized on accurate results through extensibility capability of an ABS model. Similarly, Luo et al. [9] brought up the issue through a generic ABS framework for modelling human-like behaviours and decision making. In general, all of the mentioned works emphasized on the need of extensible, replicable and reusable capability and the importance of validation and verification in a generic ABS framework. A summary of different works addressing different limitations are tabulated in Table 1.

Table 1: Related Works Addressing Different Limitations in Existing ABS Models.

Past ABS model limitations	Related works addressing the limitations.
1. Ad-hoc or domain-specific	Zutshi et al. [11]; Biedermann et al. [8]; Bagneris [12]; Hennicker et al. [15]; Luo et al. [9]
2. Validation and verification	Bagneris [12]; Hennicker et al. [15]
3. Extensible or customizable model	Biedermann et al. [8]; Zutshi et al. [11]; Bagneris [12]; Hennicker et al. [15]; Luo et al. [9]
4. Replication and reuse	Schreinemachers and Berger [16]; Hennicker et al. [15]; Bosse et al. [7]

Nonetheless, developing a new ABS model to be plugged-in the generic ABS framework is not a direct task. The input models come from domain-specific environment with their own set of parameters and the need of developing new models for different domain-specific simulation purposes is not solved. Thus, research efforts has progressed towards real-time or run-time modelling to accommodate dynamic systems that change parameters from time to time. Dynamic systems that apply such approach are complex overhead crane schedules in [17] and notification sending to different parameters of computer and mobile phone in [18].

However, such research efforts are yet to address the limitation of ad-hoc or domain-specific ABS. In a nutshell, current research efforts have yet to precisely address the limitations claimed to be alarming. Therefore, a robust ABS model is needed to cater different parameters and domains. To fill this gap, this research proposes a self-adaptive ABS model that is robust enough to model inputs from different domains, as illustrated in Figure 2(b). The objective of the self-adaptive capability in an ABS is two-fold: to facilitate development of new ABS models and to address limitations of existing ad-hoc or domain-specific ABS models.

Journal of Theoretical and Applied Information Technology
20th April 2016. Vol.86. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. PROPOSED SELF-ADAPTIVE AGENT-BASED SIMULATION MODEL

A self-adaptive ABS modelling aims to solve the need for developing separate ABS models to cater different domains. Based on the existing generic ABS framework as shown in Figure 2(a), a self-adaptive modelling algorithm is proposed to collectively model the input domains and automatically produce separate ABS models for simulation execution as illustrated in Figure 2(b). For feasibility purpose, certain scopes of simulation objectives are established for the proposed self-adaptive ABS model. As discussed in earlier research work [19],[20], a self-adaptive ABS model is feasible to model processes or chains of process from different domains. Modelling of phenomena that does not involve processes such as predictions, human intelligence or mechanical kinematics are not included in this study.

The self-adaptive ABS model is further refined of its feasibility with common parameters involved in the processes as discussed in our earlier work [19],[20]. It is worth noting that even with processes from different domain, similar parameters are used in the ABS model and these are termed as general inputs or parameters within this paper. However, apart from similar parameters used, there are specific parameters used only for the certain simulation purpose. Thus, these parameters are termed as domain-specific inputs or parameters within this paper. Hence, with the scopes and terminologies implied, a conceptual self-adaptive ABS model is constructed and illustrated in Figure 3.

The proposed self-adaptive ABS modelling in Figure 3 aims to automatically sort, categorize and arrange data or parameters inserted from different input domains. To achieve this, an agent will function as filterAgent(), to filter all the inputs and categorize them into general and domain-specific input categories. After initial categorization of inputs, storageAgent() categorizes and stores all the inputs into different parameter categories which are further categorized into domain and agent parameters into repository.

In Figure 3, General domain parameters, GDP consists of Resources, R, Task Size, TS, Task Type, TT, Number of Tasks, TN, Time, T and Workflow, W. General agent parameters, GAP, consists of Agent Capacity, C, Agent Attributes, A, and Agent Behaviours, B. In a nutshell, GDP and GAP can be denoted as follows;

$$GDP = \{R_i, TS, TT, TN, T_i, W_i\} \quad (1)$$

$$GAP = \{C_i, A_i, B_i\} \quad (2)$$

where,

$i = \{1, 2, 3, 4, \ldots\}$

It is the domain-specific parameters DDP_i and DAP_i, in the self-adaptive ABS model that supports model extension and customization. storageAgent() creates new domain-specific domain, DDP_i and agent parameters, DAP_i, to store inputs not belong to GDP or GAP, for ABS modelling and execution. ABS model is constructed by modelAgent() based on the inputs categorized and stored. After the modelling process, a simulation engine will execute the ABS models. This approach addresses the limitations of ad-hoc or domain-specific ABS models by allowing different simulation models to be constructed concurrently. At the same time, this approach solves current research limitations of having to build a new ABS model for different domains.

However, as parameters differ between domains, a reasoning algorithm has to govern storageAgent() and modelAgent() for parameters storage and modelling of desired simulation objective. This reasoning algorithm is also self-adaptive. The algorithm enables self-adaptive ABS model to autonomously construct ABS model, tailored to desired simulation domain and phenomena. The self-adaptive algorithm is explained in Section 3.1

3.1 A Self-Adaptive Algorithm

Self-adaptive algorithm governs storageAgent() and modelAgent() on the reasoning activities of input categorizations and modelling. The algorithm ensures model to store parameters and model ABS specifically to desired domain and phenomenon so that simulated models will not be ambiguous with other domains or phenomena. The top-level of the algorithm is as follows;

The self-adaptive algorithm:

```
Start
Prompt input from user
Define: Domain
GetDomain(Domain) = DDPi(D)
Check repository for history
if NewDomain(NewDomain) == TRUE, then
   Define: General and Domain-Specific Domain and Agent Parameters
   Define: Processes
   If GDP(Workflows) OR GDP(Time) == NULL , then
      Halt
      Else,
      GDP(Workflows)i = GDP(W)
      GDP(Time)i = GDP(t)
```

Journal of Theoretical and Applied Information Technology
20th April 2016. Vol.86. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
  storageAgent() define General Domain Parameters
  If GDP(Resources)i == NULL, then
    GDP(R) = 0
    Else,
    GDP(Resources)i = GDP(R)
  If GDP(TaskSize) == NULL, then
    GDP(TS) = 0
    Else,
    GDP(TaskSize) = GDP(TS)
  If GDP(TaskType) == NULL, then
    GDP(TT) = 0
    Else,
    GDP(TaskType) = GDP(TT)
  If GDP(NumberofTask) == NULL, then
    GDP(TN) = 0
    Else,
    GDP(NumberofTask) = GDP(TN)
  storageAgent() define General Agent Parameters
    If GAP(Attributes)i == NULL, then
      GAP(A) = 0
      Else,
      GAP(Attributes)i = GAP(A)
    If GAP(Behaviour)i == NULL, then
      GAP(B) = 0
      Else,
      GAP(Behaviour)i = GAP(B)
    If GAP(Capacity)i == NULL, then
      GAP(C) = 0
      Else,
      GAP(Capacity)i = GAP(C)
  If DDPi == NULL then,
    DDPi = 0
    Else,
    storageAgent() declare New DDPi Parameter
    storageAgent() define New DDPi
  If DAPi == NULL then
    DAPi = 0
    Else,
    storageAgent() declare New DAPi Parameter
    storageAgent() define New DAPi

Else if NewDomain(NewDomain) == FALSE
Then
  Define: General and Domain-Specific Domain and Agent Parameters
  Define: Processes
```

```
  If GDP(Workflows) OR GDP(Time) == NULL, then
    GDP(Workflows)i = GDP(WHistory)
    GDP(Time)i = GDP(tHistory)
    Else,
    GDP(Workflows)i = GDP(W)
    GDP(Time)i = GDP(t)
  storageAgent() define General Domain Parameters
  If GDP(Resources)i == NULL, then
    GDP(R) = GDP(RHistory)
    Else,
    GDP(Resources)i = GDP(R)
  If GDP(TaskSize) == NULL, then
    GDP(TS) = GDP(TSHistory)
    Else,
    GDP(TaskSize) = GDP(TS)
  If GDP(TaskType) == NULL, then
    GDP(TT) = GDP(TTHistory)
    Else,
    GDP(TaskType) = GDP(TT)
  If GDP(NumberofTask) == NULL, then
    GDP(TN) = GDP(TNHistory)
    Else,
    GDP(NumberofTask) = GDP(TN)
  storageAgent() define General Agent Parameters
    If GAP(Attributes)i == NULL, then
      GAP(A) = GAP(AHistory)
      Else,
      GAP(Attributes)i = GAP(A)
    If GAP(Behaviour)i == NULL, then
      GAP(B) = GAP(BHistory)
      Else,
      GAP(Behaviour)i = GAP(B)
    If GAP(Capacity)i == NULL, then
      GAP(C) = GADP(CHistory)
      Else,
      GAP(Capacity)i = GAP(C)
  If DDPi == NULL then,
    DDPi = DDPiHistory
    Else,
    storageAgent() declare New DDPi Parameter
    storageAgent() define New DDPi
  If DAPi == NULL then
    DAPi = DAPiHistory
    Else,
    storageAgent() declare New DAPi Parameter
    storageAgent() define New DAPi
```

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In the algorithm, initialization of domain is crucial. It determines whether storageAgent() will store inputs and modelAgent() to model ABS with existing records in repository or not. If new domain is entered, all the inputs are stored and modelled new without historical inputs. Different parameters are defined by storageAgent() based on inputs of GDP and GAP received. If there are any inputs found not belonging to GDP and GAP, DDP$_i$ and DAP$_i$ are initialized and defined by storageAgent(). Amongst all the different parameters, Workflows, W and Time, T, are the most important inputs. They will determine final simulation results for processes modelled. Thus, the simulation halts if both or one of the input does not exist.

However, if input domain is found to be existing in the self-adaptive ABS model by filterAgent(), the inputs will be stored and modelled accordingly and compared with the historical inputs. If there is no new input, parameters' values are defined from historical input which is noted with "History" at the end of each parameter. This is the difference in inputs storing and modelling between new domains and existing ones. Ability to model new and historical inputs address extensive capability of the model to cater new inputs and parameters for larger or different objectives and scopes of desired ABS.

In order to ensure the proposed model and algorithm function as claimed, three case studies are exclusively selected to be modelled. The three case studies are crime investigation report process, new student registration procedures and transportation requests processes. These three case studies reflect the divergence in nature and domain from each other to test self-adaptive nature of the model. Real data are acquired from each case study for validation and verification of simulation results purposes. The simulation model is expected to be able to model the three case studies as similar to the real data acquired.

One of the case study is used to give an enhanced illustration of self-adaptive algorithm and ABS model being put to application. The application is explained in Section 3.2.

3.2 Application of the Self-Adaptive ABS Model: A Case Study of Crime Report

In this section, a specific case study as an input to the self-adaptive ABS model is used to further illustrate the applicability of the self-adaptive ABS model. The case study is mainly about simulation of processes took place for a crime report and investigation of a car theft happened in Universiti Tenaga Nasional. Considering that there was a car theft at a parking bay of Engineering Faculty on Monday morning, 28 December 2015. Victim of the theft upon realization at 11:30am, went to Security Department to lodge a crime report. Investigation officer of the Security Department then interviewed the victim to acquire details and proceed straight to the crime scene for investigation. Considering that the domain exists in repository, the model is expected to model the processes as well as resources and time taken for the case to be fully reported and adjourned for further actions. From the information above, inputs inserted will be as follows;

- Car theft
- Universiti Tenaga Nasional Engineering Faculty parking bay
- Monday, 28 December 2015
- Security Department
- A victim
- An investigation officer

Suppose storeAgent() found that the defined domain, Security Department, is an existing domain in the repository, self-adaptive algorithm will appear as follows;

The algorithm:
```
NewDomain(NewDomain) == FALSE
Then
   Define: General and Domain-
Specific Domain and Agent
Parameters
   Define: Processes
   GDP(Workflows)i = GDP(WHistory)
   GDP(Time)i = GDP(tHistory)
   storageAgent() define General
Domain Parameters
   GDP(R) = GDP(RHistory)
   GDP(TS) = GDP(TSHistory)
   GDP(TT) 1 = "Major"
   GDP(TT) 2 = "Major Theft"
   GDP(TaskType)1 = GDP(TT) 1
   GDP(TaskType)1 = GDP(TT) 2
   GDP(TN) = GDP(TNHistory)
   storageAgent() define General
Agent Parameters
   GAP(A)1 = "IO"
   GAP(A)2 = "V"
   GAP(Attributes)1 = GAP(A) 1
   GAP(Attributes)2 = GAP(A) 2
   GAP(B) = GAP(BHistory)
   GAP(C)1 = 1
   GAP(Capacity)1 = GAP(C) 1
   storageAgent() declare New DDPi
Parameter(WorkingDay)
   WorkingDay(WD) = "Monday"
```

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
   storageAgent() declare New DDP_i
Parameter(Environment)
   Environment(Env) = "UNITEN"
```

Considering storageAgent() found specific historical inputs from repository for GDP and GAP and new DDP_i from the new inputs, the self-adaptive ABS model will be as constructed in Figure 4. filterAgent() categorizes the inputs, namely, security department, car theft, a victim and an investigation officer are general inputs while UNITEN Engineering Faculty and crime reported on weekday is a domain-specific input. storageAgent() then categorize and store the inputs into different parameters, which in this case study, car theft is categorize as a “Major Theft” in Task Type, TT, a victim, “V” and an investigation officer, “IO” in agent attributes, A. storageAgent() detects Security Department as an existing domain and therefore fetching historical parameter records into the model, which are labelled in shaded-cloud shape, “CCTV”, “Major”, “4”, “48hrs”, “Workflow”, “HEP” and “Police” respectively, as illustrated in Figure 4.

Upon completion of parameters categorization and storage, modelAgent() proceeds to model the simulation according to Workflow and specific parameter values. The ABS model is illustrated in Figure 4. Simulation results of total time for different processes and whole process of crime report is as illustrated in Figure 4. The new parameter of “Weekday”, a major crime report made on Weekday takes a shorter processing time, as UNITEN HEP does not function on weekends.

Through the illustrative application of the case study on the self-adaptive ABS model, it is conclusive that the self-adaptive ABS model is extensible and replicable with the ability of modelling an ABS based on new and historical inputs. The ABS model is self-adaptive, as the algorithm and the architecture accommodate automatic ABS modelling and execution tailored to specific desired phenomena. The proposed self-adaptive ABS model goes one step further from the current ones by addressing ad-hoc or domain-specific limitations without having the needs of manual ABS model development.

4. CONCLUSION AND FURTHER WORKS

An agent-based simulation framework should be extensible and replicable in nature rather than being domain-specific in order to better reflect real-life phenomena. This paper proposes a self-adaptive ABS model that allows automatic simulation modelling of processes from different domains without having to develop new ABS models for every new phenomena. A self-adaptive algorithm enabling the nature is proposed and feasibility of the model with integration of the algorithm is illustrated through ABS modelling of a case study of crime report process in Universiti Tenaga Nasional. It was shown in the application that the proposed model is able to model the case study with only data inputs instead of model integration. In future works, the proposed self-adaptive ABS model will be tested with three domains; education, security and transportation. The simulation results will be compared with real data acquired for each case study. It is hoped that the self-adaptive modelling algorithm is able to diminish modelling efforts with more accurate simulation results.

5. ACKNOWLEDGEMENTS

This work is sponsored by Kementerian Pengajian Tinggi, Malaysia under the Fundamental Research Grant Scheme (FRGS).

REFERENCES:

[1] Klügl, Franziska, and Ana LC Bazzan. "Agent-based modeling and simulation." AI Magazine 33, no. 3 (2012): 29.

[2] Macal, C. M., & North, M. J. “Agent-based modeling and simulation”. In Winter Simulation Conference. Winter Simulation Conference. December 2009. pp. 86-98.

[3] Macal, Charles M., and Michael J. North. "Introductory tutorial: Agent-based modeling and simulation." In Simulation Conference (WSC), Proceedings of the 2011 Winter, pp. 1451-1464. IEEE, 2011.

[4] Edmonds, Bruce, and Ruth Meyer. “Simulating Social Complexity, A Handbook”. Springer, Berlin, 2013. ISBN: 978-3-540-93812-5 (Print) 978-3-540-93813-2 (Online).

[5] Edmonds Bruce (2001) The Use of Models - making MABS actually work. In Moss S and Davidsson P (Eds.) Multi-Agent-Based Simulation, Lecture Notes in Artificial Intelligence 1979: 15-32. Berlin: Springer-Verlag.

[6] Bandini, S., Manzoni, S., & Vizzari, G. “Agent based modeling and simulation: an informatics perspective”. Journal of Artificial Societies and Social Simulation. 2009. 12(4), 4.

[7] Bosse, Tibor, Mark Hoogendoorn, Michel CA Klein, and Jan Treur. "An agent-based generic model for human-like ambience." In

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Constructing Ambient Intelligence, pp. 93-103. Springer Berlin Heidelberg, 2008.

[8] Biedermann, Daniel H., Peter M. Kielar, Oliver Handel, and André Borrmann. "Towards TransiTUM: A generic framework for multiscale coupling of pedestrian simulation models based on transition zones." Transportation Research Procedia 2 (2014): 495-500.

[9] Luo, Linbo, Suiping Zhou, Wentong Cai, Malcolm Yoke Hean Low, and Michael Lees. "Toward A Generic Framework for Modeling Human-like Behaviors in Crowd Simulation." 2009.

[10] Jones, Spencer S., and R. Scott Evans. "An agent based simulation tool for scheduling emergency department physicians." In AMIA Annual Symposium Proceedings, vol. 2008, p. 338. American Medical Informatics Association, 2008.

[11] Zutshi, Aneesh, António Grilo, and Ricardo Jardim-Gonçalves. "A Dynamic Agent-Based Modeling Framework for Digital Business Models: Applications to Facebook and a Popular Portuguese Online Classifieds Website." In Digital Enterprise Design & Management, pp. 105-117. Springer International Publishing, 2014.

[12] Bagneris, Jean-Charles. "FMS, a Generic Framework for Agent-Based Financial Markets Simulations." Available at SSRN 2149543 (2012).

[13] Axelrod, Robert. "Advancing the art of simulation in the social sciences." In Simulating social phenomena, pp. 21-40. Springer Berlin Heidelberg, 1997.

[14] Heath, B., Hill, R., & Ciarallo, F. “A survey of agent-based modeling practices (January 1998 to July 2008)”. Journal of Artificial Societies and Social Simulation, 2009. 12(4), 9.

[15] Hennicker, Rolf, Sebastian S. Bauer, Stephan Janisch, and Matthias Ludwig. "A generic framework for multi-disciplinary environmental modelling." PhD diss., International Environmental Modelling and Software Society, 2010.

[16] Schreinemachers, Pepijn, and Thomas Berger. "An agent-based simulation model of human–environment interactions in agricultural systems." Environmental Modelling & Software 26, no. 7 (2011): 845-859.

[17] Graunke, Adam, Gabriel Burnett, Charles Hu, and Glen Wirth. "Decision support model to evaluate complex overhead crane schedules." In Proceedings of the 2014 Winter Simulation Conference, pp. 1608-1619. IEEE Press, 2014.

[18] Morin, Brice, Olivier Barais, Jean-Marc Jezequel, Franck Fleurey, and Arnor Solberg. "Models@ run-time to support dynamic adaptation." Computer 42, no. 10 (2009): 44-51.

[19] Loo, Y.L., Alicia Y.C. Tang and Azhana Ahmad, "The Gap of Current Agent Based Simulation Modeling Practices and Feasibility of a Generic Agent Based Simulation Model", International Journal of Advanced Computer Research (IJACR), Volume-5, Issue-19, June-2015, pp.115-123.

[20] Loo, Y.L., Alicia Y.C. Tang and Azhana Ahmad, " Identifying Key Factors in Agent Based Simulation Model on Processes in Time Constrained Environment", International Symposium on Agents, Multi-agent Systems and Robotics (ISAMSR); 08/201.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

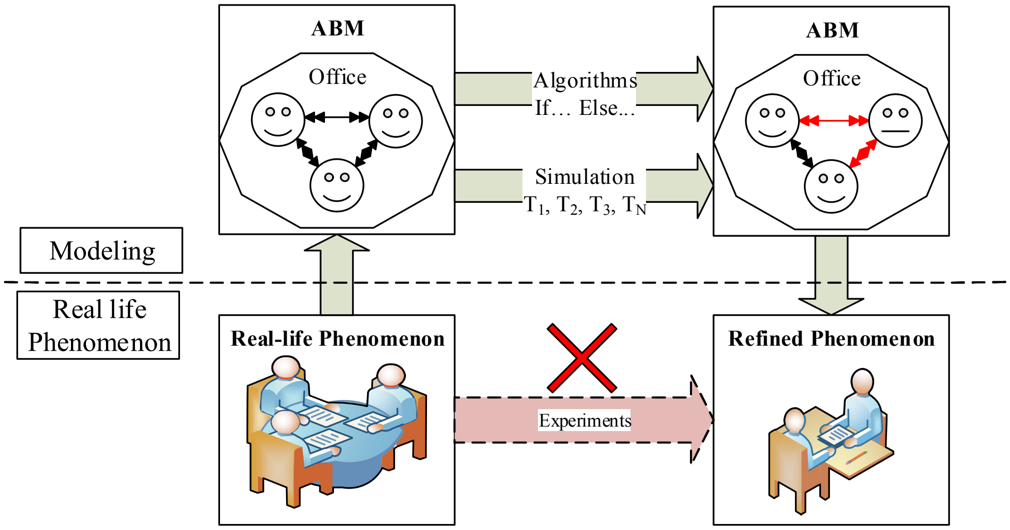

Figure 1: Real Phenomena Modelled through Agent-Based Modelling Approach.

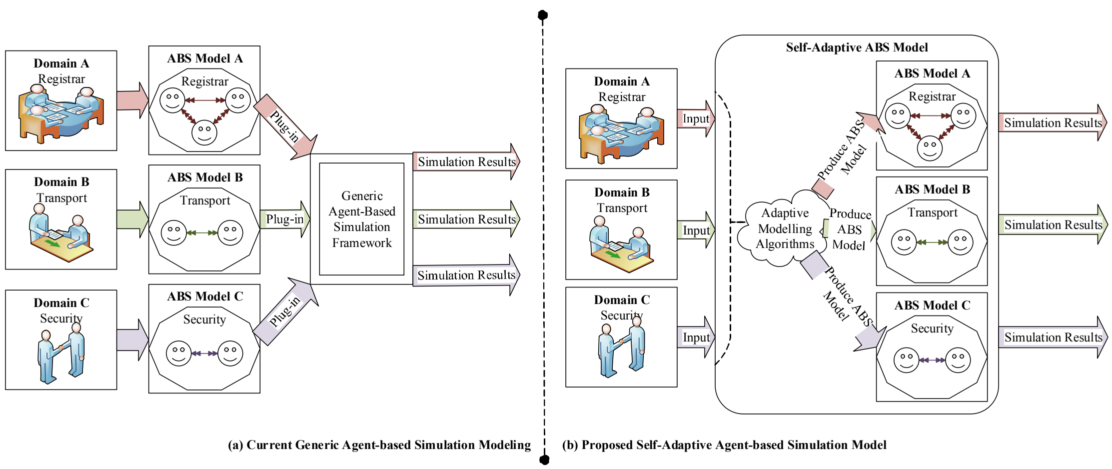

Figure 2: Current Generic Agent-based Simulation Modeling vs Proposed Self-Adaptive ABS Model.

Journal of Theoretical and Applied Information Technology
20th April 2016. Vol.86. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

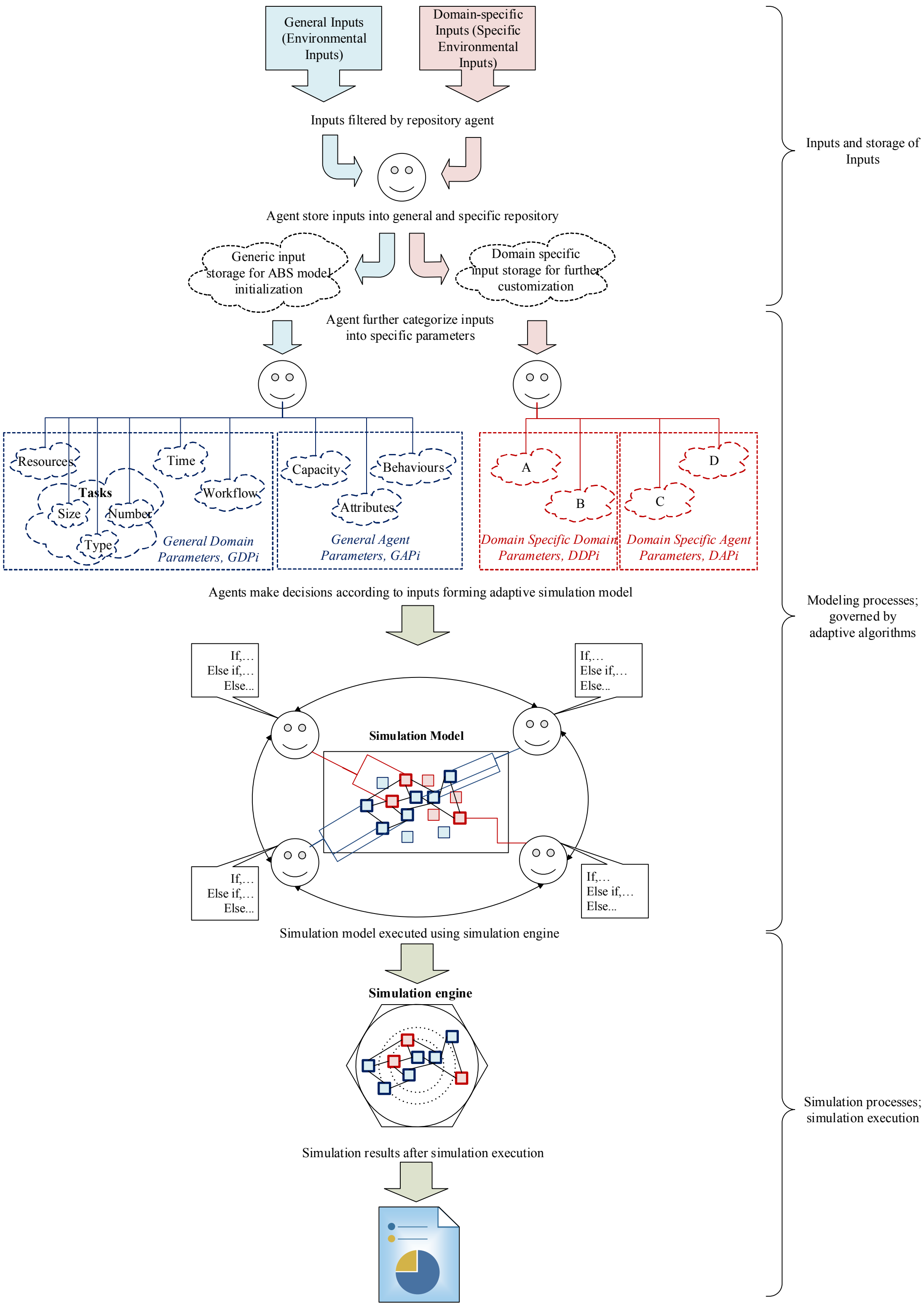

Figure 3: Architecture of the Self-Adaptive Agent-Based Simulation Model

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

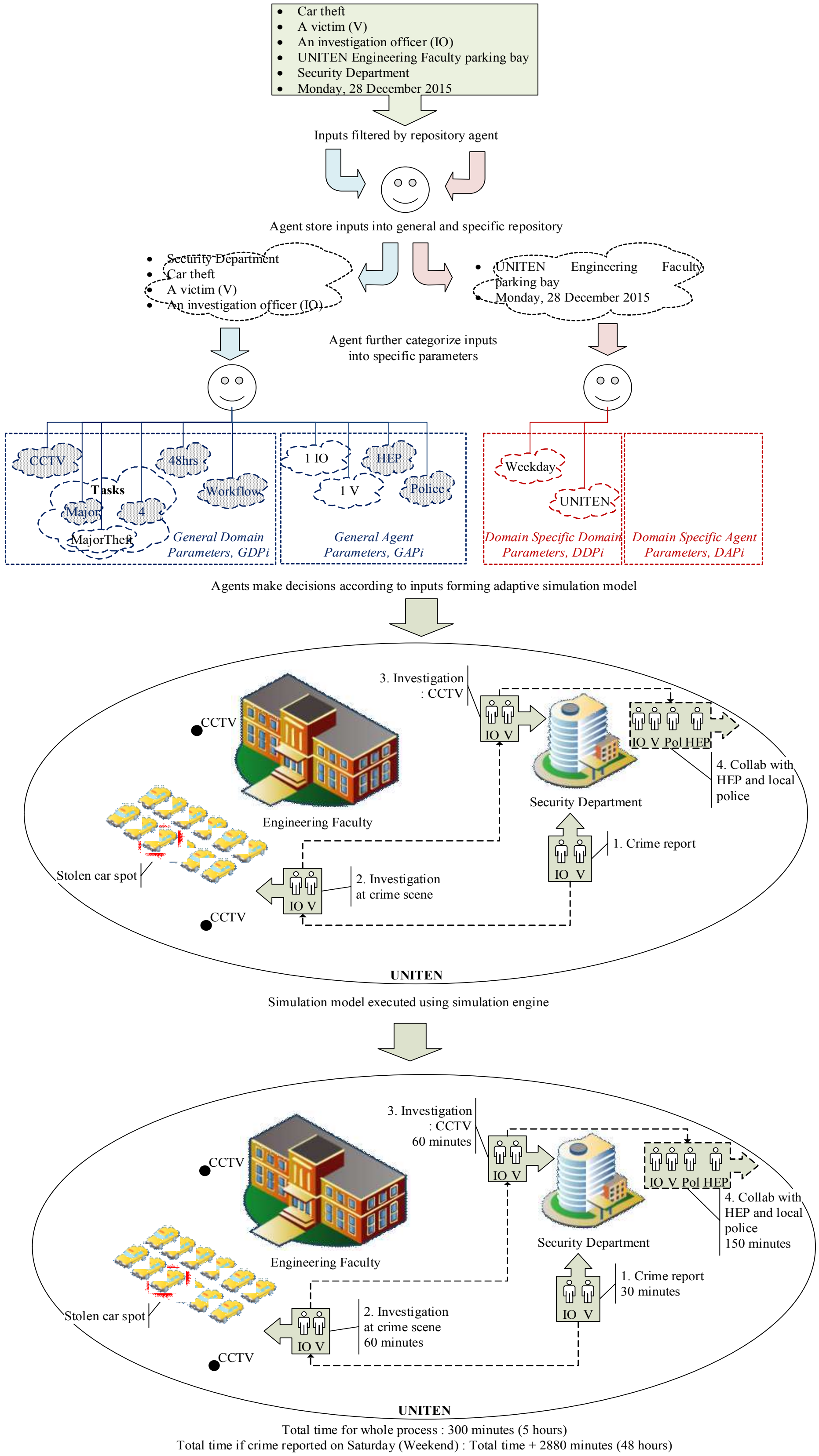

Figure 4: Self-Adaptive Agent-Based Simulation Model in Application of Case Study.