**Journal of Theoretical and Applied Information Technology**
31st January 2017. Vol.95. No.2

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

# A GENERIC AGENT-BASED DYNAMIC PROCESS SIMULATION FRAMEWORK: A SELF-ADAPTIVE MODELLING APPROACH

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

[1]Centre for Agent Technology, College of Computer Science and Information Technology,
Universiti Tenaga Nasional,
Jalan Ikram-Uniten, 43000 Kajang, Selangor, Malaysia.
Email: yimling1st@yahoo.com, aliciat@uniten.edu.my, azhana@uniten.edu.my

[2]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia,
86400 Parit Raja, Batu Pahat, Johor, Malaysia.
Email: aidam@uthm.edu.my

**ABSTRACT**

Agent-based simulation (ABS) modelling had been a prevalent approach in simulating dynamic processes for a variety of domains. However, the constructed models are usually domain-specific in nature. The proprietary issue of existing modelling approaches limit the possibilities of reusing those constructed models in other application domains. Such limitation also poses issues in model extensibility and customization. Thus, this paper presents a generic ABS framework with self-adaptive modelling approach to address the aforesaid limitations. In the ABS framework, self-adaptive modelling architecture supports construction and re-construction of models autonomously. Simulations were conducted for a few dynamic processes derived from two case studies of New Student Registration and Crime Investigation. Simulation results show that different ABS models were successfully developed, extended and customized by the self-adaptive modelling component in the framework.

**Keywords**: *Agent-based Simulation Model, Self-adaptive Modelling, Dynamic Processes, Generic Agent-based Simulation Framework, Case Study*

## 1. INTRODUCTION

Simulation had been a renowned approach for experimenting phenomena without physical re-enactment for cost and time saving. Agent technology assist simulation by capability of simulating dynamics and interactions of environment and human. Thus, agent technology was often used side-by-side with simulation, namely agent-based simulation (ABS), for various phenomenal explanations and experimentations. ABS had been widely applied in simulation of dynamic processes in different domains such as medical [1]-[3], transportation [4]-[6] and scheduling [7]-[9].

ABS modelling and systems in simulating dynamic processes are usually constructed based on specific phenomena. Specific phenomena or domain-specific simulations imply that a fixed set of parameters, domain and environment are used to model the simulation. As a result, continuous new ABS models are constructed to cater every different domain specific simulations. This caused questionable reliability of simulation results, as the aims of simulations and parameters used are too specific [1],[3]. Different languages and protocols used in construction of domain-specific ABS models and systems further caused concerns of model extension, customization and simulation result replication. Both concerns are main breakthroughs for limitations of too specific ABS modelling and results. Thus, justifications of robustness of ABS model and results' reliability as well as validation and verification were severed in domain-specific ABS modelling.

As a result, generic ABS frameworks were founded to address the limitations [10]-[16]. Generic ABS frameworks allows different ABS model to be plugged-in to the framework for simulation. In this way, simulation results could be

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

replicated and ABS model construction language were limited to one. The approach attempted to address limitations of simulation replications and diversified languages of domain-specific ABS models [17]-[18]. Figure 1(a) illustrates the concept of generic ABS frameworks. However, limitations of continuous construction of new ABS model and model extension which could secure robustness and reliability of ABS model and results remained unaddressed.

In order to address the limitations, this paper discusses the application of self-adaptive ABS model [18], into an ABS framework that conceptualize similar idea of current generic ABS frameworks. Figure 1(b) illustrates the concept of self-adaptive modelling in a generic ABS framework and the main difference between the current generic ABS frameworks with this research.

The remainder of this paper is organized as follows. Section 2 discusses current research efforts on generic ABS frameworks and the differences with the self-adaptive ABS model applied framework. Section 3 presents the self-adaptive ABS model. Section 4 discusses implementation of the self-adaptive ABS model in two case studies and results of simulations. Section 5 discusses the contributions of this work on addressing different limitations. Finally, Section 6 concludes the paper with explanation of further work of the research.

## 2. RELATED WORKS: BREAKING THROUGH LIMITATIONS

Generic ABS frameworks were introduced vastly after the realizations of limitations and the consequences of having such limitations in domain-specific ABS model and systems. Generic ABS frameworks were constructed and applied for similar purpose of prediction and forecasting in several domains. For example, DANUBIA [12] was constructed for multiple agricultural water resource managements, DYNAMOD [10] for digital business popularity forecast and TransiTUM [11] for multiscale coupling of pedestrian evacuation prediction. A summary of generic ABS frameworks addressing different limitations is tabulated in Table 1.

*Table 1: Related Works Addressing Different Limitations of Current Domain-Specific ABS Modelling.*

| Limitations | Address the Limitations |
|---|---|
| 1. Domain-specific or diversified modelling | DYNAMOD [10]; TrnasiTUM [11]; DANUBIA [12]; FMS [13]; Luo et al. [15] |
| 2. Validation and verification | DANUBIA[12]; FMS [13]; Gurcan and Bernon [14] |
| 3. Extensible or customizable model | DYNAMOD [10]; TransiTUM [11]; DANUBIA [12]; FMS [13]; Luo et al. [15] |
| 4. Replication and reuse | DANUBIA [12]; Schreinemachers and Berger [19]; Bosse et al. [20] |

The generic ABS frameworks mentioned above allow different ABS models to be plugged-in the framework for simulations. Figure 1(a) illustrates the concept of these ABS frameworks. Specific restrictions such as same language and design methodologies used for construction of the ABS models are imposed to achieve the breakthrough of diversified language and designs of ABS models. It is believed that once a common language is used, extension and customization of an ABS model could be achieved. Extensible and customizable ABS model results in capability of replication of simulation results. This will eventually lead to breakthrough of questionable reliability of simulation results.

However, the main limitation of having more domain-specific ABS models developed for different simulation objectives remained. New ABS models need to be constructed in order to be plugged-in the generic ABS framework. Time and cost are still not subtracted from the picture. Thus the introduction of run-time modelling for dynamic systems which changes over time. FIoT which allows dynamic modelling of internet of things [21], overhead crane scheduling [22] and real-time notification [23] are the examples of research that adopted such concept. However, this concept is yet to be applied in ABS.

Current direction for generic ABS framework and modelling as well as non-ABS run-time modelling are moving towards model extension and customization which addressed robustness, reliability and replication. Both generic ABS framework and non-ABS run-time modelling made incomplete efforts to address the limitations in ABS modelling. However, they are useful references for addressing the limitations with further tweaks on the concept. Thus, this research referred to the advancements of both concepts to address the limitations, which is illustrated in Figure 1(b).

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

The usefulness of ABS in simulating dynamic processes became prevalent due to mushrooming research efforts especially in medical [1]-[3], transportation policies [4]-[6] and services planning or scheduling [7]-[9],[22]. It was found that having ABS for these dynamic processes resolved the need to do physical research for outcomes of policies and processes implementations. However, domain-specific nature of these ABS models and frameworks inherited similar limitations in simulation models and results.

In order to break-through the limitations, this research integrates the concept of current generic ABS framework with self-adaptive modelling to simulate dynamic processes. The main objective of generic ABS framework with self-adaptive modelling is to enable autonomous modelling of ABS model at runtime to address the limitation of inextensible or non-customizable ABS model.

## 3. THE FRAMEWORK: SELF-ADAPTIVE ABS MODELLING

A generic ABS aims to have common language for different ABS models to enable multi-domain ABS simulation in one framework while self-adaptive modelling aims at developing different model at runtime. Both concepts are useful to address the limitations of current domain-specific ABS modelling and systems for dynamic processes. Hence, this research integrates the two concepts to construct a generic ABS framework with self-adaptive modelling approach where the architecture is illustrated as a block diagram in Figure 2.

The framework consists of two repositories and five modules. Parameters and model repositories consists of real data collected through a series of data collection as discussed previously in [], and users inputs for different simulation purposes. Four modules in the framework consist of Input Module, Self-Adaptive Modelling Module, Output Module, Validation Module and Replication Module.

At Input Module, agents in the framework differentiate some random inputs by users into common values (general inputs) and specific values (domain-specific inputs). The inputs are then brought into Self-Adaptive Modelling Module for model construction. The self-adaptive modelling module is further illustrated in Figure 3, which was discussed in previous paper [18]. Agents in self-adaptive modelling module further categorize user inputs into existing and new parameters. Existing parameters refers to general environmental and agent parameters available in the framework's parameter repository. New parameters refers to domain-specific environmental and agent parameters that needed to be made known to the framework's parameters repository for model extension and customization to ensure reliable simulation results.

Upon processing parameters, agents in the framework determines the model to be modelled out for the categorized parameters. The agents determine whether the model required is available in the Model Repository, which refers to existing model and vice versa refers to new model to be modelled out. This ensure accurate modelling of intended simulation objectives by the user. It also enable robust simulation modelling and address the alarming consequences of domain-specific ABS modelling limitations. The intelligence of agents in processing parameters and models lies in the self-adaptive algorithms. It is worthy to note that the self-adaptive algorithms were previously discussed in [18]. Thus, this paper focuses on discussing the implementation or application of the algorithms.

Prior to processing of parameters and models, the agents in the framework proceed with constructing the intended ABS model. The constructed model is then simulated in Output Module using an ABS model simulator in which, is Netlogo for this research. After simulation results are produced, both simulation results and model are validated with real data in parameter repository and real phenomena data in model repository respectively. This process is achieved in Validation Module.

The validation and verification of simulation model's robustness and simulation results' reliability is obtained through Replication Module. As said by Axelrod in [24], replication of results ensure the reliability of simulation results and robustness of simulation models, this framework enables replication of results through the Replication Module. The framework was implemented in simulation of two dynamic processes of different domains as a pilot study of its claims. The implementations are discussed in the next section.

## 4. APPLICATION OF THE GENERIC ABS FRAMEWORK: SELF-ADAPTIVE ABS MODELLING

Specific scopes were determined in this research as discussed previously in [18],[25] to ensure feasibility of autonomous ABS model construction and on the other hand, achieving the claims of self-adaptive capabilities. Thus, only simulations of dynamic processes are considered in this research

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

with the current construction of framework architecture. A fixed set of three different domains were chosen based on the common parameters the domains shared, which promotes feasibility. Diversified simulation objectives between the three domains promotes the generic and self-adaptive ABS modelling capability of the framework. The three domains are; education, security and transport.

This paper focuses on two domains: education and security. In order to further ensure feasibility of the study, the framework is implemented in specific case studies for each of the domains, namely new student registration process for education domain and crime investigation process for security domain. Generally, the dynamic processes consist of similar parameters with the other case studies but the uniqueness of each of the processes prevails. Thus, these case studies are chosen for the research.

Self-adaptive capability is justified when the self-adaptive algorithm is able to determine the model and parameters for the user inputs and the desired model is able to be constructed and simulated. It is worthy to note that the self-adaptive algorithm is thoroughly discussed previously in [18]. Therefore, the case studies will present the application of the self-adaptive algorithms, simulation modelling within the framework. The applications are presented in Section 4.1 and 4.2 and thoroughly discussed in Section 5.

### 4.1 Case Study: New Student Registration Process

New Student Registration Process is a dynamic process for education domain in Universiti Tenaga Nasional environment. This dynamic process consist of eight parameters that dynamically change in accordance to different situations. The parameters includes six primary parameters and two secondary parameters [25]. In this case, the user inputs were "240 students", "3 registrars" and "1 financer" as shown in upper-left corner of Figure 4. The inputs are processed in the self-adaptive ABS modelling algorithms as follows:

The Algorithm:

```
storageAgent(identify inputs)
NewDomain(NewDomain) == FALSE
Domain: Education
Then
Define: GDPi, GAPi, DDPi and DAPi
GDP(Workflows)i = GDP(WHistory)
GDP(Time)i = GDP(tHistory)
GDP(R) = GDP(RHistory)
GDP(TT) = GDP(TTHistory)
GDP(TN) = GDP(TNHistory)
GAP(A)1 = "students"
GAP(A)2 = "registrars"
GAP(A)3 = "financer"
GAP(A)4 = GAP(AHistory: "document checkers")
GAP(B)i = GAP(BHistory)
GAP(C)1 = 240
GAP(C)2 = 3
GAP(C)3 = 1
GAP(C)4 = GAP(CHistory: "2")
modelAgent(model NewStudentRegistrationProcess)
```

Education domain is identified by storageAgent() based on "students" and "registrars" being in user inputs. Afterwards, storageAgent() identified available model and parameters for the domain in repository and incorporate domain-specific parameter values entered by user inputs. Workflows, $W_i$, time$_i$, $T_i$, resources, $R$, task type, $TT$, number of tasks, $TN$, are general domain parameters, $GDP_i$. Their values are fixed based on available values in repository for they are not fixed in the user inputs.

Agent attribute, $A_i$, agent behaviour, $B_i$ and agent capacity, $C_i$, are general agent parameters, $GAP_i$. Three $A_i$ and $C_i$ values are fixed based on user inputs while extra $A_4$ which is "document checkers" and $C_4$ which is "2" as shown in Figure 4(a) are fixed based on the available model and parameters in repository. ModelAgent() constructs the simulation model as shown in Figure 4(a): right column and detailed view in Figure(b) based on the model and parameters identified by storageAgent().

Then, the model is simulated by ABS simulator engine, NetLogo to simulate the dynamic process. The simulation results are shown in detail in the middle-left column of Figure 4(a) and graphically in lower-left column of Figure 4(a). The results shown time taken for the registration of 240 students, by 3 registrars, 2 document checkers and 1 financers with different $TT$ incorporated in the algorithm, as follows:

```
TT1: Documents not complete, initial fees paid
TT2: Documents not complete, initial fees not paid
TT3: Documents complete, initial fees paid
TT4: Documents complete, initial fees not paid
```

The simulation results and model as compared to the user's desired simulation objectives, real data and available model in repository were cross-validated and verified. The constructed simulation

Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

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

model reflected user's desired simulation objectives. According to the output segment middle-left column of Figure 4(a), the simulation results reflected the real data. Student queued number "74" might surpass the time of student queued before with condition $TT_3$. Thus, the generic ABS framework with self-adaptive modelling capability is conclusively succeeded in this case study.

### 4.2 Case Study: Crime Investigation Process

Crime Investigation Process is a dynamic process for security domain in Universiti Tenaga Nasional environment. This dynamic process consist of nine parameters that are dynamic and dependant to various scenarios. The parameters include six primary parameters and three secondary parameters [25]. For this case study, two types of scenarios were tested to further justify the self-adaptive capability to accommodate modelling of different domain-specific parameters.

#### 4.2.1 Crime Investigation Process: Stolen Car

The first scenario is having a stolen car in engineering parking bay. In this case, the user inputs were "engineering parking bay" and "a stolen car" as shown in upper-left corner of Figure 5(a). Prior to the input from user, the inputs are processed in the self-adaptive modelling algorithms as follows:

The Algorithm:

```
storageAgent(identify inputs)
NewDomain(NewDomain) == FALSE
Domain: Security
Then
Define: GDP_i, GAP_i, DDP_i and DAP_i
GDP(Workflows)_i = GDP(WHistory)
GDP(Time)_i = GDP(tHistory)
GDP(R) = GDP(RHistory: "CCTV")
GDP(TT)_1 = GDP(TTHistory: "Major")
GDP(TT)_2 = GDP(TTHistory: "Major Theft")
GDP(TN) = GDP(TNHistory)
GDP(TS) = GDP(TSHistory)
GAP(A)_1 = GAP(AHistory: Investigation Officer)
GAP(A)_2 = GAP(AHistory: Complainant)
GAP(B)_i = GAP(BHistory)
GAP(C)_i = GAP(CHistory)
storageAgent() declare newDDP_i (crime scene) = "engineering parking bay"
modelAgent(model CrimeInvestigationProcess)
```

Security domain is identified by storageAgent() based on "stolen car" being in user inputs. Subsequently, storageAgent() identified available model and parameters for the domain in repository and incorporate domain-specific parameter and parameter values entered by user inputs. Workflows, $W_i$, time$_i$, $T_i$, resources, $R$, task type, $TT$, number of tasks, $TN$ and task size, $TS$, are general domain parameters, $GDP_i$. Their values are fixed based on values stored in repository, for they are not fixed in the user inputs.

Specific resources, $R$, is identified through recognition of "engineering parking bay" in the user inputs, where the location is facilitated with CCTV. Specific task type, $TT$, is identified through recognition of "stolen car", which implied major crime case. Agent attribute, $A_i$, agent behaviour, $B_i$ and agent capacity, $C_i$, are general agent parameters, $GAP_i$. Two $A_i$ "investigation officer" and "complainant" and $C_i$ values are fixed based on data found in repository for the model. Prior to identified model and parameters, modelAgent() constructs the simulation model as shown in Figure 5(a).

The model is then simulated using NetLogo the ABS simulator engine. The simulation results are shown in detail in the lower-left column of Figure 5(a). The results showed time taken in units of "ticks" for the different workflows that investigation officer and complainant went through until the case is solved. The constructed simulation model reflected user's desired simulation objectives. According to the output segment lower-left column of Figure 5(a), the simulation results reflected the real data. As engineering parking bay is facilitated with CCTV, the investigation included crime scene visitation and CCTV recording analysis. The second scenario can be compared to this result, to further justify the self-adaptive capability.

#### 4.2.2 Crime Investigation Process: Lost Wallet

Second scenario is a report of lost wallet in library as shown in the user inputs textbox in Figure 5(b). In this case, the user inputs were simply "library" and "lost wallet". The inputs are then processed in the self-adaptive modelling algorithms as follows:

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

The Algorithm:

```
storageAgent(identify inputs)
NewDomain(NewDomain) == FALSE
Domain: Security
Then
Define: GDPi, GAPi, DDPi and DAPi
GDP(Workflows)i = GDP(WHistory)
GDP(Time)i = GDP(tHistory)
GDP(R) = GDP(RHistory)
GDP(TT)1 = GDP(TTHistory: "Minor")
GDP(TT)2 = GDP(TTHistory:
"Negligence")
GDP(TN) = GDP(TNHistory)
GDP(TS) = GDP(TSHistory)
GAP(A)1 = GAP(AHistory:
Investigation Officer)
GAP(A)2 = GAP(AHistory:
Complainant)
GAP(B)i = GAP(BHistory)
GAP(C)i = GAP(CHistory)
storageAgent() declare newDDPi
(crime scene) = "library"
modelAgent(model
CrimeInvestigationProcess)
```

StorageAgent() identified the domain as "security" based on "lost wallet" being in user inputs. Loss of objects are identified as crime report. Subsequently, available model and parameters are identified in repository and domain-specific parameter and parameter values entered by user inputs are incorporated. $GDP_i$ values are fixed based on values stored in repository, for they are not found in the user inputs. Specific task type, *TT*, is identified through recognition of "lost wallet", which implied minor crime case. According to repository, a more specific *TT*, identified as "negligence" is appropriate for this scenario. For $GAP_i$, two $A_i$ "investigation officer" and "complainant" and $C_i$ values are fixed based on data found in repository for the model. ModelAgent() then constructs the simulation model accordingly, as shown in Figure 5(b).

Simulation results are shown in detail in the lower-left column of Figure 5(a) using NetLogo. The constructed simulation model reflected user's desired simulation objectives of simulating the time taken for investigation of lost wallet in library. The simulation results reflected real data, as investigation officer skip investigation process for "minor crimes" with "negligence" as the cause, which is also the case in real-world phenomena. Comparing with the constructed ABS model for first scenario, the self-adaptive modelling revealed the capability to accommodate simulation of different dynamics of parameters within a domain. This justifies the capability of self-adaptive modelling includes extensive and customizable ABS model.

## 5. DISCUSSIONS

Prior to the results in the pilot studies discussed in Section 4, it is conclusive that the generic ABS framework with self-adaptive modelling is able to autonomously model an ABS out of just a few user inputs. From the simulation results, it is conclusive that the framework is successful in its claims of addressing the limitations of domain-specific ABS model and frameworks. The framework has fulfilled both objectives; the constructed ABS model are autonomous, extensible and customizable for reliable simulation results.

The generic ABS framework was able to accurately process the few user inputs in natural language. Accurate models are constructed for both first and second pilot studies. This justifies the framework's capability of autonomously modelling multi-domain ABS at runtime. This capability address the limitation of domain-specific ABS models and frameworks for simulating too specific objectives and results. Subsequently, extensive and customizable construct of ABS model was enabled.

The model shows extensive capability, accommodating flexibility of skipping workflows in accordance to different *TT* in security domain to strictly executing every workflows in education domain. The ability of differentiating two different scenarios in the second pilot study further reveal the extensible and customizable capability of the ABS model. The model is able to be customized into skipping investigation workflow for self-negligence cases, which is similar to real data for this specific crime reports.

Models constructed with self-adaptive ABS modelling were robust to accommodate different variations of parameters. Too specific ABS simulation objectives produce questionable reliability of simulation results. Thus, the robust models which could accommodate changes of parameters as presented in first and second pilot study address this major limitation.

The robustness of simulation model and reliability of simulation results are not depending only on its capability to accommodate enlargement of ABS objectives. They are justified by constant validation and verification of simulation model to user desired simulation objectives and simulation results to real data. This justifies the claims of this research for addressing the specified limitations.

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

## 6. CONCLUSION AND FURTHER WORKS

It is no longer a necessity for developing new ABS models tailored to specific simulation objectives of dynamic processes or to be plugged-in to a generic ABS framework. The generic ABS framework with self-adaptive ABS modelling of this research enables construction of ABS models at runtime. Autonomous construction of ABS model addresses critical limitations of domain-specific ABS model, which includes high cost and time consumptions of constructing new models. Model extensibility and customizability addressed the major limitation of reliability of simulation results by domain-specific ABS models. The ability of the framework for achieving such goals was demonstrated through a series of case studies application of the framework in two different case studies i.e. registration of new students and crime investigation processes. The ability to construct simulation results that are comparable to real data justified reliability of simulation model and results. As commendable results were achieved through these case studies, the framework is proposed to be tested on another case study from other domain for further validations and verifications.

## 7. ACKNOWLEDGEMENTS

This work is supported by the Fundamental Research Grant Schemes CFRGS) by the Ministry of Education Malaysia under the Grant Ref. No. FRGS/1/2013/ICT07/UNITEN/02/02.

**REFERENCES**:

[1] 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.

[2] Othman, Sara Ben, Nesrine Zoghlami, Slim Hammadi, Alain Quillot, Alain Martinot, and Jean-Marie Renard. "Dynamic patients scheduling in the Pediatric Emergency Department." Latest Trends on Systems 2 (2010).

[3] Lu, Ta-Ping, Pei-Fang Tsai, and Ya-Chen Chu. "An agent-based collaborative model for orthopedic outpatient scheduling." In Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on, pp. 621-626. IEEE, 2014.

[4] Kokkinogenis, Zafeiris, Nuno Monteiro, Rosaldo JF Rossetti, Ana LC Bazzan, and Pedro Campos. "Policy and incentive designs evaluation: A social-oriented framework for Artificial Transportation Systems." In Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on, pp. 151-156. IEEE, 2014.

[5] Dalapati, Poulami, Arambam James Singh, Arin Dutta, and Surya Bhattacharya. "Multi agent based railway scheduling and optimization." InTENCON 2014-2014 IEEE Region 10 Conference, pp. 1-6. IEEE, 2014.

[6] Takahira, Satoshi, Ryo Kanamori, and Takayuki Ito. "Experiment on Activity-travel Survey System based on Scheduling System." Journal of information processing 22, no. 2 (2014): 263-269.

[7] Ginanzar, Arief, Cahyadi Nugraha, and Khuria Amila. "Model Simulasi Untuk Alat Evaluasi Rencana Jadwal Perkuliahan Jurusan Teknik Industri Itenas." Reka Integra 2, no. 4 (2014).

[8] Ruiz, Mercedes, Pablo Javier Tuya González, and Daniel Crespo. "Simulation-based optimization for software dynamic testing processes". International Journal on Advances in Software, 7 (1-2) (2014).

[9] Xi, Hui, Chi Keong Goh, Partha Sarathi Dutta, Meng Sha, and Jie Zhang. "An Agent-based Simulation System for Dynamic Project Scheduling and Online Disruption Resolving." In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pp. 1759-1760. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[10] 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.

[11] 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.

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

[12] Nikolic, V. V., S. P. Simonovic, and D. B. Milicevic. "Generic Modeling Framework For Integrated Water Resources Management." 21st Canadian Hydrotechnical Conference. Recent Flood Management Lessons in Canada. Banff, Alberta, May 14-17, 2013.
[13] Bagneris, Jean-Charles. "FMS, a Generic Framework for Agent-Based Financial Markets Simulations." Available at SSRN 2149543 (2012).
[14] Gürcan, Önder, Oğuz Dikenelli, and Carole Bernon. "Towards a generic testing framework for agent-based simulation models." In Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on, pp. 635-642. IEEE, 2011.
[15] Luo, Linbo, Suiping Zhou, Wentong Cai, Malcolm Yoke Hean Low, and Michael Lees. "Toward a generic framework for modeling human behaviors in crowd simulation." In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02, pp. 275-278. IEEE Computer Society, 2009.
[16] Bosse, Tibor, Mark Hoogendoorn, Michel CA Klein, and Jan Treur. "An agent-based generic model for human-like ambience." In Constructing Ambient Intelligence, pp. 93-103. Springer Berlin Heidelberg, 2007.
[17] Loo, Yim Ling, Alicia YC Tang, and Azhana Ahmad. "Identifying key factors in agent-based simulation model on processes in time-constrained environment." In Agents, Multi-Agent Systems and Robotics (ISAMSR), 2015 International Symposium on, pp. 71-76. IEEE, 2015.
[18] Loo, Yim Ling, Alicia YC Tang, Azhana Ahmad, and Aida Mustapha. "Towards A Self-Adaptive Agent-Based Simulation Model." Journal of Theoretical & Applied Information Technology 86, no. 2 (2016).
[19] 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.
[20] Bosse, Tibor, Mark Hoogendoorn, Michel CA Klein, and Jan Treur. "An agent-based generic model for human-like ambience." In Constructing Ambient Intelligence, pp. 93-103. Springer Berlin Heidelberg, 2007.
[21] do Nascimento, Nathalia Moraes. "FIoT: A Framework for Self-Adaptive and Self-Organizing Internet of Things Applications." Monografias em Ciencia da Computacao, No. 04/2015. ISSN: 0103-9741. November 2015.
[22] 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.
[23] Morin, Brice, Olivier Barais, Jean-Marc Jezequel, Franck Fleurey, and Arnor Solberg. "Models @ Runtime to support dynamic adaptation." Computer 42, no. 10 (2009): 44-51.
[24] Axelrod, Robert. "Advancing the art of simulation in the social sciences." In Simulating social phenomena, pp. 21-40. Springer Berlin Heidelberg, 1997.
[25] Loo, Yim Ling, Alicia YC Tang, Azhana Ahmad and Aida Mustapha. "Workflow Analysis for Self-Adaptive Agent-based Simulation Model." In 2nd International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR), pp. 17-22. 2016.

Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

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

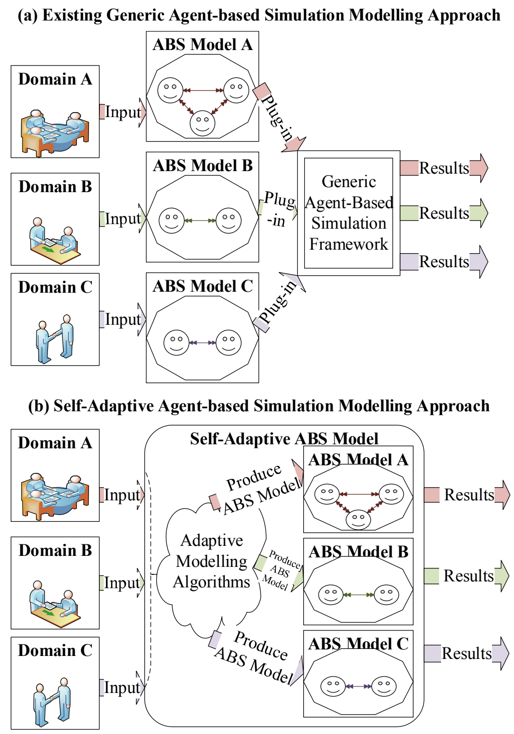



*Figure 1: Existing Generic Agent-based Simulation Modelling Approach vs Self-Adaptive ABS Modelling Approach.*

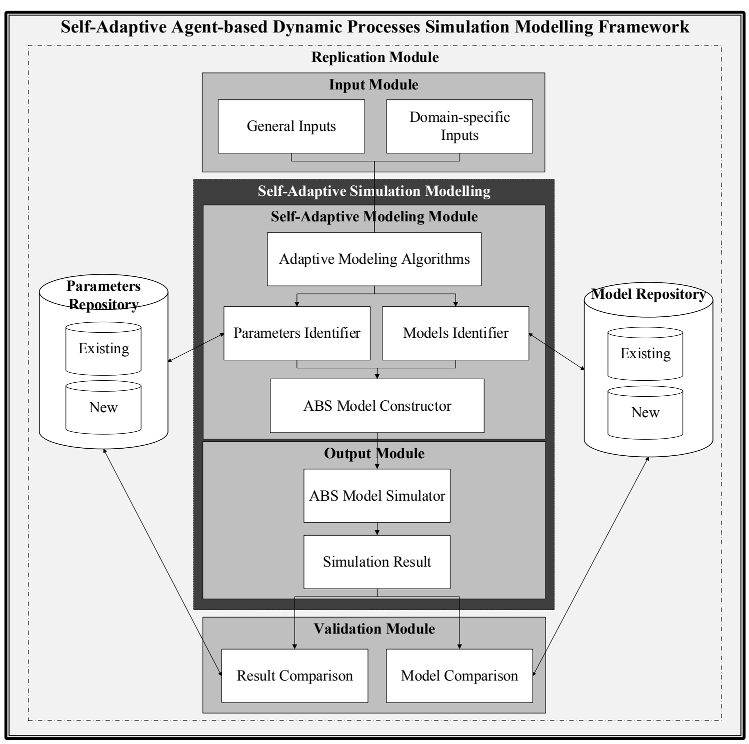



*Figure 2: A Generic ABS Framework: Self-Adaptive Modelling*

Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

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

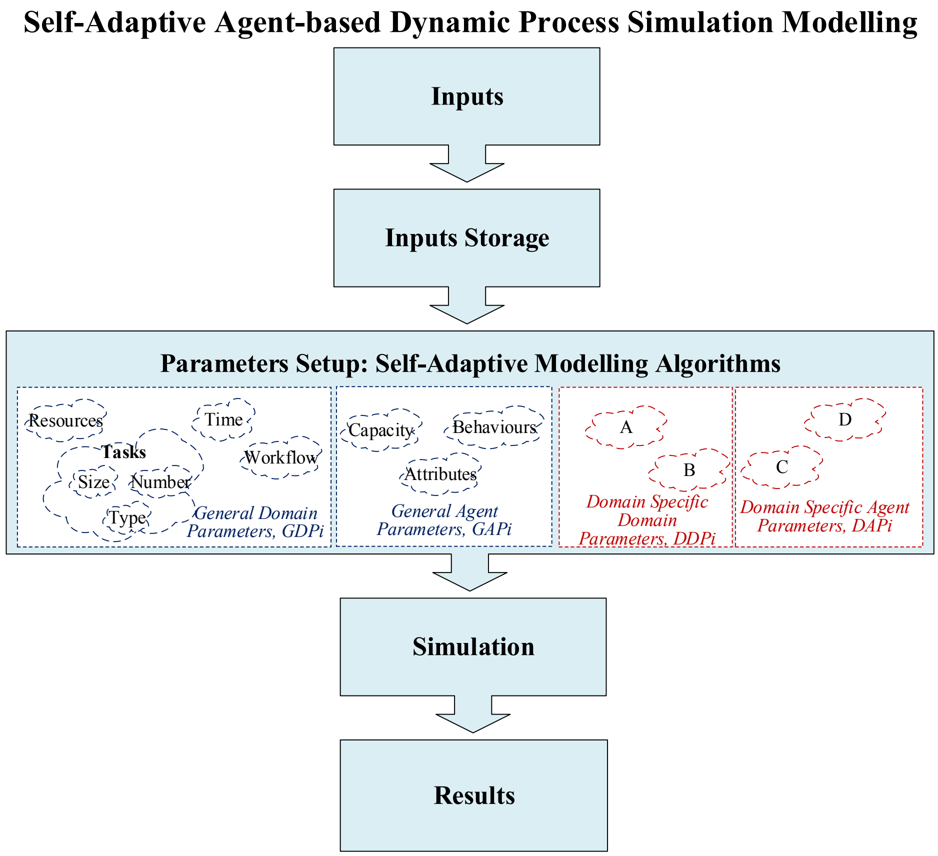
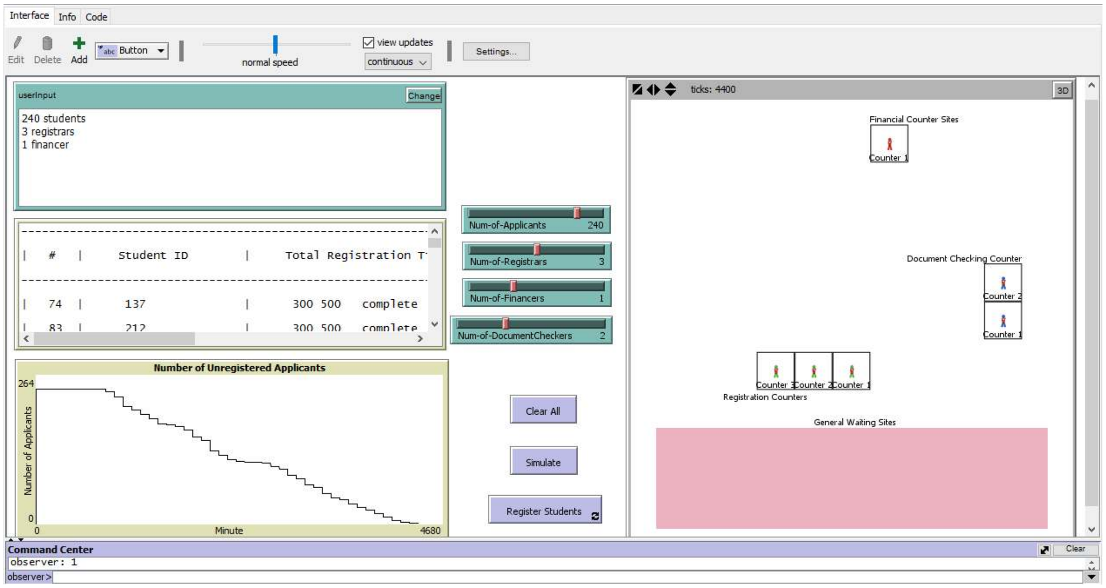


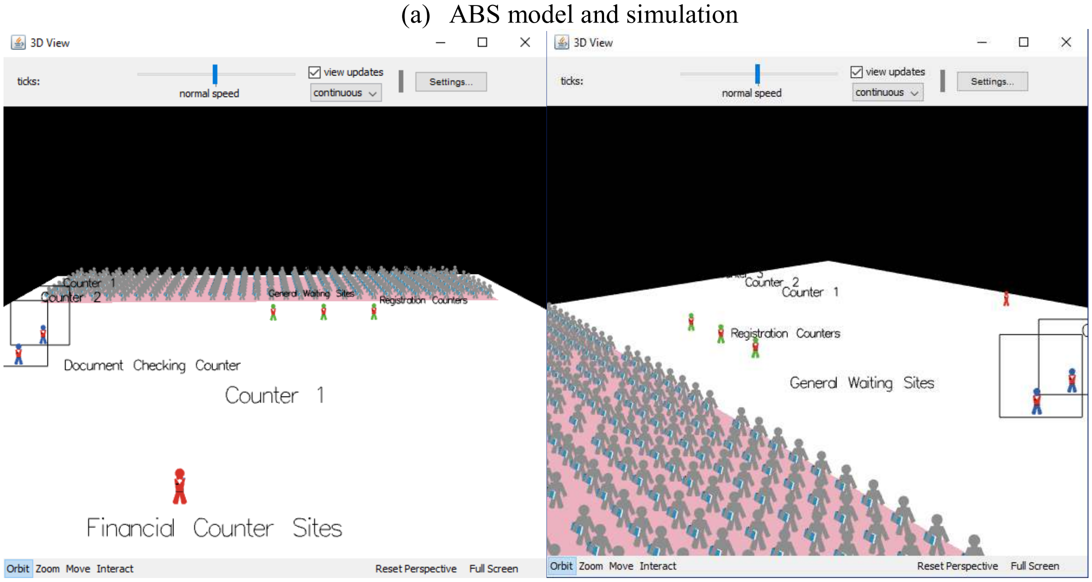

*Figure 3: Architecture of the Self-Adaptive Agent-Based Simulation Model*

**Journal of Theoretical and Applied Information Technology**
31st January 2017. Vol.95. No.2

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



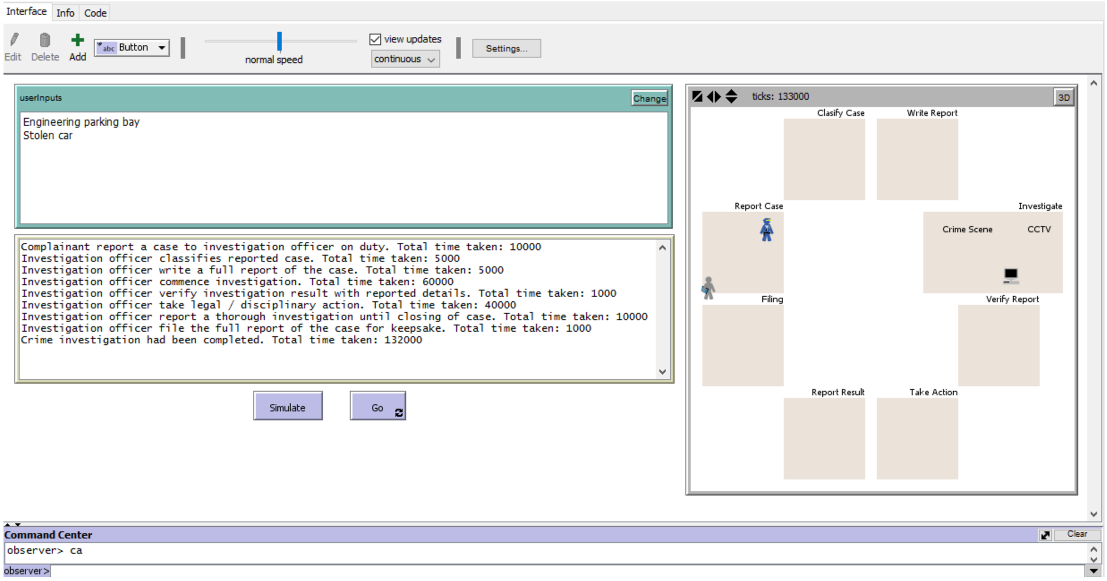

(a) ABS model and simulation



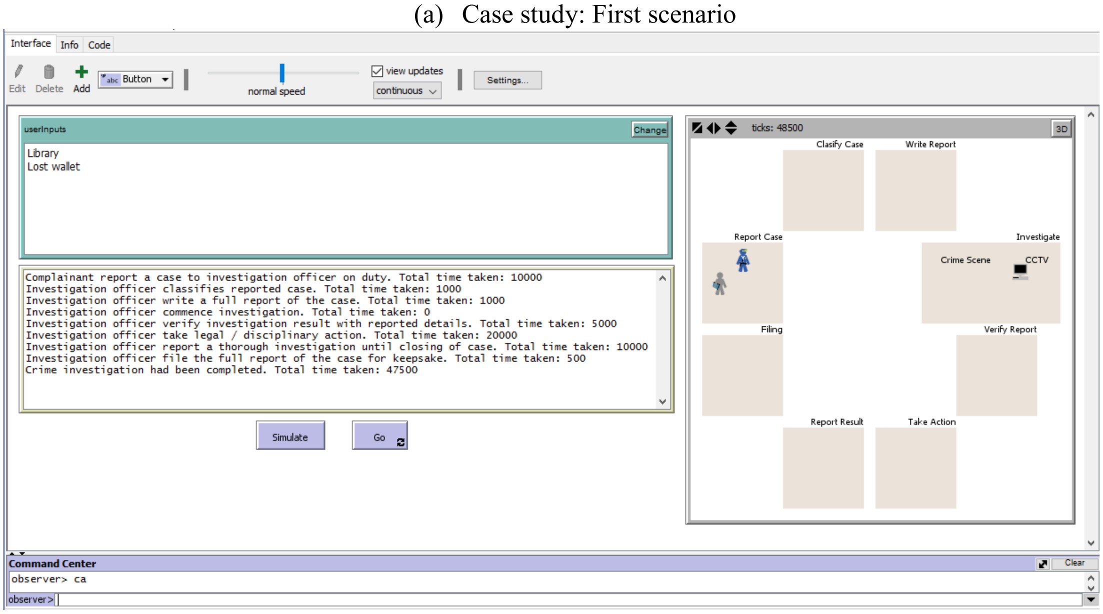

(b) ABS Model in detailed view

*Figure 4: Case Study: Self-Adaptive Agent-Based Simulation Model Application in Simulating New Student Registration Process.*

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




(a) Case study: First scenario




(b) Case study: Second scenario

*Figure 5: Case Study: Self-Adaptive Agent-Based Simulation Model Application in Crime Investigation Process*