

Agent-based Simulation with Process- interaction Worldview

Shufang Xie, Tao Zhang*, Oliver Rose

Fakultät für Informatik, Universität der Bundeswehr München, Werner-Heisenberg Weg 39, D-85577 Neubiberg, Germany; *tao.zhang@unibw.de

SNE 29(4), 2019, 169 - 177, DOI: 10.11128/sne.29.tn.10492
Received: May 10, 2019 (Selected ASIM SST Hamburg 2018
Postconf. Publ.), Accepted: Sept. 10, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. After analyzing the agent-based simulation (ABS), we realize that most researchers focus on the agent-based modeling, but only a few researchers pay attention to the simulation. The reason is that the agent-based model (ABM) can run directly in a real-time manner by communication among agents and the running of the ABM is already one kind of simulation. However, the ABS is less efficient when being used into a non-real-time system even though it can be speeded up by giving a timescale. So, to speed up the ABS, we introduce a process-interaction worldview (PIW) originated in the discrete event simulation to the ABS. A method for combining the agent-based simulation and the PIW is proposed. The method is validated by applying in a simple queue system and compared with the normal ABS with a timescale.

Introduction

There still exists controversy over the definition of agents. One concept of the agent appears in the distributed intelligence in Artificial Intelligence [1], in which besides distributed, autonomous, and social features, agents have to be intelligent, such as being able to perceive, learn, and adapt to the environment. Most of the researchers following this concept direct towards the multi-agent system [2], and some researchers focus on the hardware agent [3] which has been widely used in the robotics [4]. Another concept, which is derived from the emergence theory [5] in which a key notion is that simple rules generate complex behaviors, in other words, system properties emerge from its constituent

agent interactions [6], is very similar to the cellular automata [7] in which agents are asked to keep simple and short, which is contrary to the first concept because the intelligence certainly makes agents more complex. However, they also have a lot in common, such as autonomy, society, distribution and so on. In most research, the contrary parts of the agents seem to be discarded. Both the intelligence and the simplicity are not given weight, but much attention is paid to the common parts, autonomy, society, and distribution [8-10]. These research lead to the formation of a new bottom-top modeling method [11], i.e., agent-based modeling, in which a system is modeled as a collection of autonomous agents. Based on a set of rules each agent individually assesses its situation, makes decisions and may execute various appropriate behaviors for the system it represents [6]. The agent-based modeling is kind of similar to the object-oriented modeling [12], but more flexible and more natural to describe the system. There are no strict requirements of intelligence or simplicity for the agents in the agent-based model.

Now we define the ABS. The ABS is the process of designing an ABM of a real system and conducting experiments with this model to understand the behavior of the system and evaluating various strategies for the operation of the system [13].

At present, the researches on the ABS mainly focus on two directions, agent-based modeling, and its application. The first one is trying to build a standard, and universal framework for the modeling [14-16]. Autonomy, society, and distribution involve in. Another one focuses on creating domain agents through studying their attributes and behaviors [17-19]. Both have provided lots of approaches and mechanisms for agent-based modeling and application in practice. However, only a few researches focus on the simulation and its efficiency.

The study will discuss the efficiency of the non-real-time ABS and combine the ABS with the discrete event simulation to boost the efficiency of the simulation. The paper is structured as follows: In Section 1, we present the efficiency problem of the ABS and analyze some worldviews which may speed up the simulation. In Section 2, a procedure for introducing process-interaction worldview to the ABS is proposed. In Section 3, some experiments are carried out to validate the procedure. The paper is concluded in Section 4.

1 Agent-based Simulation

1.1 Simulation in ABS

As we mentioned before, the current studies on the ABS focus on the ABM, and few researchers pay attention to the simulation in the ABS. Probably because the ABM can run directly in a real-time way[11], in which the running of the ABM is a simulation run, and there is no necessity to study the simulation separately. However, this type of simulation misses some important contents in the computer simulation such as the worldview in the simulation, also referred to as a simulation strategy. Therefore, many researchers studying on the computer simulation, especially the discrete event simulation, cannot help asking where the simulation is in the ABS. The so-called simulation in the ABS is a real-time simulation in which the simulation time equals the real time. However, the ABS is less efficient when being used in a non-real-time system. In some researches, the ABS is speeded up by giving a timescale under the condition of synchrony. There are two ways to achieve the synchronization: conservative algorithm and optimistic algorithm. The Conservative algorithm keeps the model running in sync exactly, but the optimistic algorithm allows asynchronous phenomenon to occur and then makes it synchronous, such as SimJade does. The SimJade[20] is a synchronization service for the JADE using an optimistic synchronization technique to manage the time in a distributed way. Because the optimistic synchronization techniques allow the asynchronous phenomenon to appear, the agents influenced by the asynchronous phenomenon have to roll back. So, lots of time is consumed by the rollback.

In addition, the real-time ABS with a timescale has two features: (1) there is no central time manager, and the agents move on according to their local time (computer clock); (2) The time spent on executing code is

counted in the simulation time. In the real-time ABS without a timescale, the time for code execution is very short and can be ignored. On the other hand, in the real world, it also takes time while people make a decision. So, the time for executing the codes exists reasonably in the real-time ABS. However, in the non-real-time ABS with the time scale, the execution time is enlarged, and errors occur. And the worse thing is that the precision of simulation results will decrease as the timescale increases. We face a tradeoff between the efficiency and the precision. So how to speed up the ABS without losing any precision is a key issue.

1.2 Efficiency of the ABS

If back to the computer simulation again, we can find that the simulation has different efficiencies when different worldviews are used. There are two main types of the worldview, time-driven worldview, and event-driven worldview. In the time-driven worldview, the world progresses as the time is passing with a fixed increment (time step). Correspondingly, the world progresses as some events occur in the event-driven worldview which includes three sub worldviews: event scheduling, activity scanning, and process interaction. Introducing a suitable worldview into the ABS will be a good way to speed up the simulation. But some researchers [12, 20] argue that the simulation worldview violates the autonomy principle of the agents due to the centralization of time handling and sharing. So, most researchers did not study the ABS and the worldview together. Siebers even declares that the discrete event simulation is dead, long live the ABM [21]. However, the time handling is only in charge of simulation time which is independent and never affected by the agent. The local clocks in each agent merged into one sharing simulation clock does not intervene in the behavior of agents at all, and the agents still take action autonomously. Therefore, it is possible to introduce the worldview to the ABM.

A few researches have already focused on this field and obtained great progress, such as the entity-relationship and agent-oriented-relationship (ER\AOR) [22]. Agents, objects, events, and messages are entities in the ER\AOR; the agents and the objects are distinguished; the messages and the events are managed together to control the simulation time.

In the ER\AOR, it is natural to partition the simulation system into the environment simulator and some agent simulators. The environment simulator is responsible for advancing simulation time and managing the state of all external (or physical) objects and the external/physical state of each agent; a number of agent simulators are in charge of managing the internal (or mental) state of agents. By means of ER modeling and combination with the discrete event simulation, the ER\AOR has attracted extensive attention. However, we realize that the ER\AOR simulation is not a pure ABS because of the objects in the model. In the pure ABS, the objects which cannot be modeled as agents must belong to certain agents (attributes). In addition, because the conditional events or messages are involved, and the lifecycle of the agent is divided into many activities, the AOR simulation is the ABS with the activity-scanning worldview. As we all know, the activities-scanning worldview is not the most efficient one. It is better to choose a more efficient worldview for the ABS. Now we come to the next question: which worldview is more suitable and efficient for the ABS.

1.3 Worldview for the ABS

The event-scheduling worldview focuses on the events that instantaneously transform a system's state and schedule future events [23]. The advantage of this worldview is that periods of inactivity can be skipped over by jumping the clock from one event time to the next event time. The event-based approach is the most computationally efficient one of the three classical worldviews. The activity-scanning worldview focuses on activities and their preconditions (triggers) [23]. An activity's preconditions must be satisfied for an activity's operations. This worldview is less efficient than the event-scheduling worldview because it requires a frequent evaluation of conditions. The process-interaction worldview can be considered a combination (hybrid) of the activity-scanning worldview and the event-scheduling worldview [24]. It focuses on processes and the entities that flow through the processes and interact with resources [25]. The process-interaction worldview is more efficient than the activity-scanning worldview, but it is less efficient than the event scheduling worldview.

From the analysis above, we can see that the process-interaction worldview is the second most efficient worldview.

But besides efficiency, the choice of worldview should be made by considering other characteristics such as maintainability, modifiability, reusability, and ease of development. The process-interaction worldview is considered to be a natural way to describe models [26] and is closer to most people's mental model. In addition, the notion of "process" corresponds closely to the lifecycle of the agent and the implementation of the process-interaction worldview is very similar to the agent-based model. Moreover, if we use the event-scheduling worldview or the activity-scanning worldview, the flexibility, maintainability, and modifiability of the ABM will shrink. Therefore, we introduce the process-interaction worldview into the ABS to speed up the simulation.

2 Agent-based Simulation with the PIW

In this section, we will introduce an approach (ABS&PIW) which brings the process-interaction worldview into the agent-based simulation. First, we discuss agents in detail including its attributes, behaviors, and messages. And then some concepts in the ABS&PIW are defined. To make the approach more formal and rigorous, we formulate the approach. At last, we discuss the parallelism in the ABS&PIW.

2.1 Agents in the Agent-based Simulation

In our study, an agent has some attributes, behaviors, messages, and activation points (see Figure 1). Agents are endowed with behaviors to make independent decisions, and the messages are medium for their communications. Activation points are designed for the process interaction worldview.

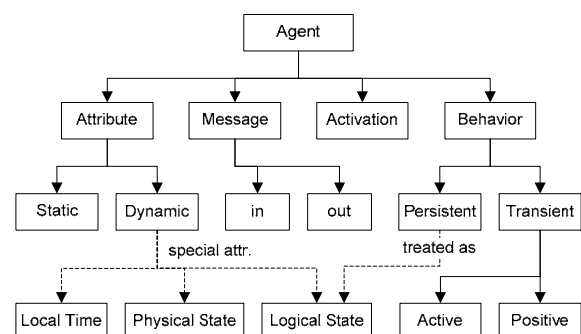


Figure 1: Attributes, behaviors, messages, and activations in an agent.

Attributes

Attributes are characteristics of the agent. An agent's attributes can be static, i.e., not changeable during the simulation, or dynamic, i.e., changeable by behaviors as the simulation progresses. For example, a static attribute is an agent's name; a dynamic attribute is an agent's memory of past interactions. The agent adapts to the environment by changing its attributes. There can be a large number of attributes in an agent, but only attributes related to the goal of the system need to be considered. The agent has three special attributes: local time, physical state, and logical state. The local time is from the inner clock in the agent, and it may be not synchronous with the simulation time. The two states will be defined in Section 2.2.

Behaviors

There are two types of behaviors: persistent behaviors and transient behaviors. The persistent behaviors are equal to activities, and they will change the state of the agent. One persistent behavior is related to one of the logical states, so the persistent behaviors are treated as logical states. Transient behaviors, which are behaviors considered in the agents, can be divided into passive behaviors and active behaviors. The passive behaviors are responsible for receiving messages and updating dynamic attributes, and the active behaviors are in charge of generating and sending messages.

Messages and Activation

Agents can receive messages from the other agents and send messages to them. The message has a given format and typically contains sender, receivers, sending time, keywords, and content. The sending time is the local time of the sender agent when the message is sent. The local time is updated with the sending time of the received messages. The concept of activation will be introduced in Section 2.2

2.2 Concepts in the ABS&PIW

Concepts in the ABS&PIW

Firstly, six concepts are given. Delays and activation points come from the process interaction method but offer some improvements.

Physical state of the agent, which has two states, active and blocked, is related to the implementation. If an agent is blocked, it gives up control of the CPU. Otherwise, it occupies the CPU.

Logical state of the agent is closely connected to the application domain and is as same as the state of the entity. The logical state is a very important dynamic attribute.

State of the agent-based model has two types, ready or unready. If the physical states of all agents are blocked, the state is ready. Otherwise, it is unready.

Straggler message is a message that its sending time is earlier than the local time of the agent who receives the message. It means a later message is received before the earlier message. The straggler message will make the simulation wrong and must be avoided. There are two ways to avoid it: a conservative algorithm which does not allow the straggler message and an optimistic algorithm which allows it and then corrects it.

Delay is a period in which the logical state of the agent is not changeable. When a delay occurs, the agent will create the next activation point and become blocked.

Activation point is the time position where the agent will be after a delay ends. The agent is activated at this point and performs actions until a new delay occurs. An activation point has such a given format including activation time, activation agent, and keywords. There are two types of activation point, conditional and non-conditional. Non-conditional activation point is explicit. In contrast, the conditional activation point is uncertain in which delays of the agents does not end until the agents meet the given condition.

Relation among some Concepts

The relation among these concepts is shown in Figure 2. The agent is similar to the active entity; the life cycle of the agent is the process of the entity and is made up of a series of activation-delay-activation. The activation point is located at the time an event occurs, and the physical state is active at this point. The agent responds to the event by a transient behavior. During a delay, an activity is carrying out, and the physical state of the agent is blocked. The activity is a persistent behavior corresponding to the certain logical state.

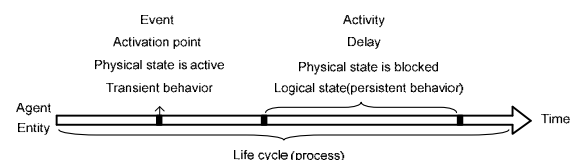


Figure 2: Relation between activation point, event, and so on.

2.3 Formulation of the ABS&PIW

Symbol Definition

- t, t_{plan} current and planned simulation time
 t_{sa} sampling interval time
 s_{mo} state of the ABM(0-ready,1-unready)
 u flag of model update (0-no update,1-update)
 R_{FAL}, R_{FAL}^0 activations in the future activation list and initial value
 R_{CoAL}, R_{CoAL}^0 activations in CoAL and initial value
 R_{CuAL} current activations in CuAL
 a an agent, $a \in A, A$ is a set of all agents
 r_a current activation point of the agent a
 c_a type of activation point(1-conditional,0-uncondt.)
 g_{r_a} flag of the condition(0-unmeet,1-meet)
 R_a a set of activation types from the agent a
 m_a a message received by the agent a
 M_a^{out} a set of message types sent by the agent a
 M_a^{in} a set of message types received by a
 τ_{m_a} timestamp of the message m
 τ_r activation time (timestamp of the activation point)
 τ_a local time of agent a updated by the time stamp
 τ_{m_a} or τ_r
 s_a physical state of the agent a
 f_a agent attribute, $f_a \in F_a$
 F_a, F_a^0 a set of all attributes and initial value
 b agent behavior
 B_a^{po}, B_a^{ac} a set of passive and active behaviors
 $H_a^1 = RS(r_a, b_a^{po}) = \{h_{r,b}^1 \mid r \in R_a, b \in B_a^{po}\}$ relationship between activations and passive behaviors. If b is related to $r, h_{r,b}^1 = 1$, otherwise $h_{r,b}^1 = 0$. The following relationships have the same rule.
 $H_a^2 = RS(m_a^{in}, b_a^{po}) = \{h_{m,b}^2 \mid m \in M_a^{in}, b \in B_a^{po}\}$
 $H_a^3 = RS(r_a, b_a^{ac}) = \{h_{r,b}^3 \mid r \in R_a, b \in B_a^{ac}\}$
 $H_a^4 = RS(m_a^{in}, b_a^{ac}) = \{h_{m,b}^4 \mid m \in M_a^{in}, b \in B_a^{ac}\}$
 $H_a^5 = RS(b_a^{po}, f_a) = \{h_{b,f}^5 \mid b \in B_a^{po}, f \in F_a\}$
 $H_a^6 = RS(b_a^{ac}, a', m_a^{out}) = \{h_{b,a',m}^6 \mid b \in B_a^{ac}, a' \in A, m \in M_a^{out}\}$

$$H_a^7 = RS(b_a^{ac}, r_a) = \{h_{b,r}^7 \mid b \in B_a^{ac}, r \in R_a\}$$

Four-tuple of the ABS&PIW

We provide a mathematical framework for the ABS following the process interaction worldview. The simulation is specified as a four-tuple

$$SIM = (I, TM, ABM, O).$$

I is a set of inputs

$$I = (R_{FAL}^0, R_{CoAL}^0, \{F_a^0 \mid a \in A\})$$

including initial activation points and initial attributes of all agents. O is a set of outputs

$$O = (\{(t, F_a^t) \mid a \in A, 0 < t < t_{aim}, \text{mod}(t, t_{sa}) = 0\}),$$

which is made up by attributes of all agents at each sample point. TM is a time manager

$$TM = (t, t_{plan}, R_{FAL}, R_{CoAL}, R_{CuAL})$$

who is in charge of the simulation time and manages all of the activation points created by agents. The activation points are grouped into three lists: conditional activation list (CoAL), future activation list (FAL) and current activation list (CuAL). FAL and CoAL are direct lists in which the activation points come from agents. CuAL is an indirect list in which the activation points are moved in from FAL. The planned time t_{plan} is the maximal simulation time. The simulation will end when the simulation time t reaches the planned time. Note that there is no TM in the general real-time ABS. ABM is an agent-based model described as

$$ABM = (s_{mo}, u, \{(\tau_a, s_a, F_a, B_a^{po}, B_a^{ac}, M_a^{in}, M_a^{out}, R_a, RS_a) \mid a \in A\}).$$

The model state s_{mo} and the updated flag u are used by the TM to advance the simulation. The simulation clock advances whenever the model state is ready and the model does not update anymore. In this way, we can avoid the straggler messages and ensure that all conditional activation points which meet the corresponding conditions are activated as soon as possible. This is a conservative synchronization algorithm. The ABM is different from the general one which contains three elements, agents, relationship, and messages [11]. In a complex system, the relationship changes dynamically and there are massive situations. It is difficult to express the relationship among all agents by a two-axis matrix. But for the individual agent, the situations of relationship with other agents are countable.

So, in our ABM, the relationship and messages are specified in the individual agents and the target agents with corresponding messages can be got by some simple IF-THEN rules. RS_a is a set of the relationship sets, $RS_a = (H_a^1, H_a^2, H_a^3, H_a^4, H_a^5, H_a^6, H_a^7)$, which contains seven relationships in the agent such as the relationship between received messages (in) and passive behaviors, as well as the relationship between active behaviors, target agents and corresponding messages (out). Local time and physical state are two special attributes and play a great role in the simulation. We extract them from the attributes and consider separately. The physical state is used to determine the model state.

Procedure of the ABS&PIW

The following is a procedure for the simulation. Simulation initialization (1), advancing time (2), and activating agents (3) are executed by the *TM*. The simulation is initialized with the inputs. Simulation clock t advances according to the time of the earliest activation points. All concurrent activation points, including both current activation points and conditional activation points, are activated at a time.

(1) Initialize

$$t = 0, R_{FAL} = R_{FAL}^0, R_{CoAL} = R_{CoAL}^0$$

$$F_a = F_a^0, s_a = 0, \text{ where } a \in A$$

$$s_{mo} = 1$$

(2) Advance time

if $R_{FAL} = \phi$ or $t > t_{plan}$ then simulation ends

$$\tau_{min} = \min(\{\tau_r \mid r \in R_{FAL}\})$$

$$R_{CuAL} = \{r \mid \tau_r = \tau_{min}, r \in R_{FAL}\}$$

$$R_{FAL} = R_{FAL} - R_{CuAL}$$

$$t = \tau_{min}$$

(3) Activate

$$R' = \{R_{CoAL}, R_{CuAL}\}, R_{CuAL} = \phi$$

if $R' = \phi$ then go to (2)

$$A' = \{a' \mid r_{a'} \in R'\}$$

$$S_{A'} = 1, \text{ where } S_{A'} = \{s_{a'} \mid a' \in A'\}$$

TM activates A' at R'

(4) When an agent $a' \in A'$ is activated at $r_{a'} \in R'$

$$\tau_{a'} = \tau_{r_{a'}}$$

Get passive behavior b_1 by $H_{a'}^1$,

which satisfies $h_{r_{a'}, b_1}^1 = 1, h_{r_{a'}, b_1}^1 \in H_{a'}^1$

if $c_{r_{a'}} = 1$ then

$g_{r_{a'}} = g_{r_{a'}}^{new}$, where $g_{r_{a'}}^{new}$ is new one from b_1

If $g_{r_{a'}} = 0$ then $s_{a'} = 0$ and go to (6)

end if

if $F_{a'} \neq \phi$ then $F_{a'} = F_{a'}^{n'}$,

where $F_{a'}^{n'}$ are new values calculated by b_1

and $F_{a'} = \{f \mid h_{b_1, f}^5 = 1, f \in F_{a'}, h_{b_1, f}^5 \in H_{a'}^5\}$

$u=1$

Get active behavior b_2 by $H_{a'}^3$,

which satisfies $h_{r_{a'}, b_2}^3 = 1, h_{r_{a'}, b_2}^3 \in H_{a'}^3$

if $\exists r_{a'} \in R_{a'} : h_{b_2, r_{a'}}^7 = 1, h_{b_2, r_{a'}}^7 \in H_{a'}^7$ then

if $c_{r_{a'}} = 0$ then $R_{FAL} = R_{FAL} + \{r_{a'}\}$

if $n c_{r_{a'}} = 1$ then $R_{CoAL} = R_{CoAL} + \{r_{a'}\}$

end if

$(A'', M_{a'}^{out'}) = \{(a'', m) \mid h_{b_2, a'' m}^6 = 1, h_{b_2, a'' m}^6 \in H_{a'}^6\}$ if

$A'' \neq \phi$ then

$S_{A''} = 1$, where $S_{A''} = \{s_{a''} \mid a'' \in A''\}$

send messages $M_{a'}^{out'}$ to A''

end if

if $c_{r_{a'}} = 1$ then $R_{CoAL} = R_{CoAL} - \{r_{a'}\}$

$s_{a'} = 0$

go to (6)

(5) When an agent $a'' \in A''$ receives the message $m_{a''}$ from the agent a'

It is similar to step (4) and just needs the following replacements and to ignore condition activation:

$$a' \rightarrow a'', r_{a'} \rightarrow m_{a''}, H_{a'}^1 \rightarrow H_{a''}^2, H_{a'}^3 \rightarrow H_{a''}^4$$

(6) When an agent is blocked

if $\forall a \in A : s_a = 0$ then $s_{mo} = 0$

if $s_{mo} = 0$ then

if $u = 1$ then $u = 0$ and go to (3)

if $u = 0$ then go to (2)

end if

In steps (4, 5), when an agent becomes active or receives messages, passive behaviors handle the received messages or activations and update its attributes. Active behaviors create new activation points and communicate with others. Decisions on the timing of advancing the time and quitting repeat of the conditional activations (6) are made by ABM according to the model state and the updated flag whenever the physical state of one agent becomes blocked.

Parallelism in the ABS&PIW

In the agent-based simulation, agents run in parallel. While concurrent activation points are activated simultaneously, associated agents will respond in parallel. The parallelism in ABS is shown in Figure 3. To avoid the straggler messages mentioned above, we adopt the conservative synchronization algorithms to ensure the correct local time (see step 6). Even though it cannot fully take advantage of parallelism, it can prevent the straggler messages from appearing at all and save the rollback time in the optimistic algorithm.

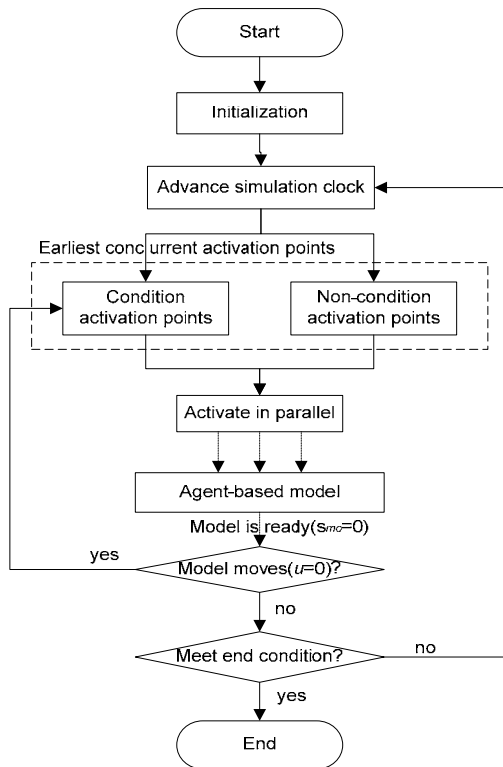


Figure 3: Parallelism in the agent-based model.

3 Experiments

A queuing system $M/M^r/1$ with a batch service is one of the classical discrete event systems. We use it to validate the proposed approach. An ABM of the system is built for the queue system, and the simulation result is compared with the theoretical value. We also compare the efficiency of the approach with the real-time ABS by using the built model.

3.1 Queuing System $M/M^r/1$

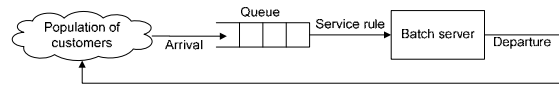


Figure 4: A Queuing system.

The queuing system $M/M^r/1$, shown in Figure 4, consists of an infinite population of customers, an infinite queue with FCFS (First Come First Serve) dispatching rule, and one batch server. The batch server provides service in batches (of size r) for arrived customers based on the rule. Customers who arrive and find the server busy join in the queue. Customers in a batch start service at the same time and depart together after served. The interval arrival and service times follow exponential distributions.

3.2 Agent-based Model of the $M/M^r/1$ Queuing System

Three types of agent are abstracted from the queuing system: customer source, customer, and server. Because a server is also an active entity, activation points of the whole system are simplified to two types: customer arrival and service completion. We build a single group ABM for the queuing system. The customer source generates customers according to a certain time distribution. The behaviors of the customers are requesting service, joining the queue to wait, accepting service, and leaving the system. The behaviors of the server include handling customer messages and providing service. The queue is part of the server agent. After a customer is served, the queue will use the given rule to choose new customers to begin service.

3.3 Correctness Verification

Assuming that customers arrive one by one; arrival rate λ is 9.76 per hour; the service batch r is 3; service rate μ is 5 per hour. Measures of performance in the steady state can be calculated with the following formulas [27].

The probability that n customers are in the system,

$$P_n = \begin{cases} (1 - s_0^{-n-1}) / r & 0 \leq n < r \\ \rho(s_0 - 1)s_0^{r-n-1} & r \leq n \end{cases}$$

where $\rho = \lambda / (r\mu)$, s_0 satisfies $|s_0| > 1$ and

$$r\rho s_0^{r+1} - (1 + r\rho)s_0^r + 1 = 0$$

The average number of customers in the system,

$$\bar{N} = (r-1)/2 + 1/(s_0 - 1)$$

The average number of customers in the queue,

$$\bar{N}' = (r-1)/2 + 1/(s_0-1) - r\rho$$

The average waiting time of customers,

$$\bar{W} = (r-1)/(2\lambda) + 1/(r\mu(s_0-1))$$

The average time of customers in the system,

$$\bar{T} = (r-1)/(2\lambda) + 1/(r\mu(s_0-1)) + 1/\mu$$

| Results | p_0 | $p(n>0)$ | \bar{N}' | \bar{N} | \bar{W} (min) | \bar{T} (min) |
|-------------|-------|----------|------------|-----------|--------------------|--------------------|
| Theoretical | 0.067 | 0.933 | 3.05 | 5.00 | 0.37 | 0.57 |
| ABS&PIW | 0.067 | 0.933 | 3.01 | 4.96 | 0.31 | 0.51 |

Table 1: Comparison between theoretical result and simulation result.

3.4 Comparison with the Real-time ABS (Time Scale)

In order to prove the less efficiency and precision, a brief simulator for the real-time ABS with timescale is developed by using JADE (Java Agent Development Environment). In JADE, the messages are not in sync sent and received (asynchronous communication). To achieve the simulation with the timescale, we improved it to the synchronous communication. After improvements, an agent (1) who just sends a message (a) will move on only after the message (a) is received and handled by its receiver agent (2). If the receiver (2) needs to send another message (b) in the message (a)'s handling process, the agent (1) has to wait for the receiver (2) until its message (b) is received and handled by another receiver. In addition, when a delay occurs in the agent, the agent will be blocked until the delay ends. The timescale is used in such delays to decrease the delay time so as to speed up the simulation.

We still use the queuing system but with the constant arrival rate (3 per hour) and service rate (3 per hour) to avoid the stochastic influences. The theoretical values can be got easily, shown in Table 2. The simulation runs ten days, and the simulation results from our approach and the real-time ABS are shown in Table 2 too.

We can see that the results of our approach are as same as the theoretical values. It takes only 0.374 seconds (the configuration of hardware is Intel i3-330M 2.13GHz CPU and 2GB memory). However, for the ABS with the timescale, the error is very big, and it also took the longer time.

4 Conclusions

Because the PIW is more natural and closer to the mental model, it is combined with the ABS to speed up the simulation. The ABS&PIW approach is proposed on the basis of the agent-based model. We provide a four-tuple $SIM = (I, TM, ABM, O)$ with elements, inputs (I), time manager (TM), ABM, and outputs (O) to describe the approach strictly. The procedure of the approach is presented mathematically in which the simulation clock advances in a sequence of activation points and all concurrent activations are activated at a time, and associated agents respond in parallel. A conservative algorithm is adopted to avoid straggler messages while the simulation is running. The result from an application to the queuing system $M/M^r/1$ shows the validity of the proposed approach. Comparing the efficiency with the real-time ABS, it performs more efficiently. Besides the advantages mentioned above, the ABM can be naturally combined with the process interaction worldview. The flexibility, maintainability, and modifiability of the ABM are also enhanced in this way.

| Results | p_0 | $p(n>0)$ | \bar{N}' | \bar{N} | \bar{W} (min) | \bar{T} (min) | Time spent (s) |
|---------------------------|-------|----------|------------|-----------|-----------------|-----------------|----------------|
| Theoretical | 0.000 | 1.000 | 1.00 | 4.00 | 20.0 | 80.0 | - |
| ABS&PIW | 0.000 | 1.000 | 1.00 | 4.00 | 20.0 | 80.0 | 0.374 |
| ABS (scale 70000) | 0.002 | 0.998 | 1.07 | 3.20 | 24.6 | 86.4 | 12.300 |
| ABS (scale 500000) | 0.041 | 0.959 | 1.05 | 1.85 | 147.5 | 300.4 | 1.743 |

Table 2: Comparisons with real-time ABS (time scale).

References

- [1] Sugumaran V. Distributed Artificial Intelligence, Agent Technology, And Collaborative Applications. 2008: Information Science Reference.
- [2] Lian J, Shatz SM, He X. Flexible coordinator design for modeling resource sharing in multi-agent systems. *Journal of Systems and Software*, 2009. 82(10): p. 1709-1729.
- [3] Tapia DI, et al. Integrating hardware agents into an enhanced multi-agent architecture for Ambient Intelligence systems. *Information Sciences*, (0).
- [4] García C, et al. Design and modeling of the multi-agent robotic system: SMART. *Robotics and Autonomous Systems*, 2012. 60(2): p. 143-153.
- [5] Namatame A. Collective Intelligence of Networked Agents, in *Emergent Intelligence of Networked Agents*, A. Namatame, S. Kurihara, and H. Nakashima, Editors. 2007, Springer Berlin Heidelberg. p. 159-176.
- [6] Bonabeau E. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 2002. 99(Suppl 3): p. 7280-7287.
- [7] Wolfram S. *Cellular automata and complexity*. 1994: Addison-Wesley Pub. Co.
- [8] Vánca J, Márkus A. An agent model for incentive-based production scheduling. *Computers in Industry*, 2000. 43(2): p. 173-187.
- [9] Oliva G, Panzieri S, Setola R. Agent-based input-output interdependency model. *International Journal of Critical Infrastructure Protection*, 2010. 3(2): p. 76-82.
- [10] McLane AJ, et al. The role of agent-based models in wildlife ecology and management. *Ecological Modelling*, 2011. 222(8): p. 1544-1556.
- [11] Macal CM, North MJ. Tutorial on agent-based modelling and simulation. *J of Sim*, 2010. 4(3): p. 151-162.
- [12] Davidsson P. Multi Agent Based Simulation: Beyond Social Simulation, in *Proceedings of the Second International Workshop on Multi-Agent-Based Simulation-Revised and Additional Papers*. 2001, Springer-Verlag. p. 97-107.
- [13] Shannon RE. *Systems simulation: the art and science*. 1975: Prentice-Hall.
- [14] Botta N, et al. A functional framework for agent-based models of exchange. *Applied Mathematics and Computation*, 2011. 218(8): p. 4025-4040.
- [15] Fortino G, Russo W. ELDAMeth: An agent-oriented methodology for simulation-based prototyping of distributed agent systems. *Information and Software Technology*, 2012. 54(6): p. 608-624.
- [16] Grimm V, Berger U, Bastiansen F. A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, 2006. 198(1-2): p. 115-126.
- [17] Chen L. Agent-based modeling in urban and architectural research: A brief literature review. *Frontiers of Architectural Research*, 2012. 1(2): p. 166-177.
- [18] Rebaudo F, Dangles O. An agent-based modeling framework for integrated pest management dissemination programs. *Environmental Modelling & Software*, (0).
- [19] Cao Y, Chen X-h. An agent-based simulation model of enterprises financial distress for the enterprise of different life cycle stage. *Simulation Modelling Practice and Theory*, 2012. 20(1): p. 70-88.
- [20] Pawlaszczyk D, Timm IJ. A hybrid time management approach to agent-based simulation, in *Proceedings of the 29th annual German conference on Artificial intelligence*. 2007, Springer-Verlag: Bremen, Germany. p. 374-388.
- [21] Siebers PO, et al. Discrete-event simulation is dead, long live agent-based simulation! *J of Sim*, 2010. 4(3): p. 204-210.
- [22] Wagner G. AOR Modelling and Simulation: Towards a General Architecture for Agent-Based Discrete Event Simulation, in *Agent-Oriented Information Systems*, P. Giorgini, B. Henderson-Sellers, and M. Winikoff, Editors. 2004, Springer Berlin / Heidelberg. p. 174-188.
- [23] Miller JA, et al. Investigating Ontologies for Simulation Modeling, in *Proceedings of the 37th annual symposium on Simulation*. 2004, IEEE Computer Society. p. 55.
- [24] Zeigler B, Praehofer H, Kim T. *Theory of Modeling and Simulation*, Second Edition. 2000: Academic Press.
- [25] Banks J, Carson JS. Process-interaction simulation languages. *SIMULATION*, 1985. 44(5): p. 225-234.
- [26] Franta WR, Maly K. An efficient data structure for the simulation event set. *Commun. ACM*, 1977. 20(8): p. 596-602.
- [27] Gelenbe E, Pujolle G. *Introduction to queueing networks*. b, ed. a. Vol. c. 1987, New York: John Wiley & Sons. 900.