Steps Towards a Balance between Adequacy and Time Optimization in Agent-based Simulations

A Practical Application of the Temporality Model Time Scheduling Approach

Tahina Ralitera, Nathan Aky, Denis Payet and Rémy Courdier

Laboratoire d'Informatique et de Mathématiques, University of Reunion Island, France

Keywords: Simulation Platform, Scheduler, Time Management, Temporality, Comparison, SimSKUAD.

Abstract: Simulation models are often used as decision support tools. They intend to imitate real world complex phenomena or processes. For this purpose, simulation platforms has the simulated models evolve by advancing a virtual time. Thus, users could benefit from shorter, but relevant simulations. The simulated time is managed by the scheduler and how this is done could significantly impact on the performance of the simulation platform. However, conventional time scheduling approaches are still limited in some situations. The temporality model approach was proposed as an interesting solution. It addresses a number of criteria that the classical time scheduling approaches do not fulfil. In this paper, we first describe the fundamental of the temporality model approach. Then, we implement it in a simulation model called the SKUADCityModel. Finally, we show the performance advantages of this type of approach.

1 INTRODUCTION

Agent-Based simulation models are usually used as decision support tools. They allow, for example, to evaluate the possible consequences of real projects before their implementation in the real world. For that purpose, a simulation platform has a simulation model evolve following a virtual time. In general, this virtual time progresses much faster than real time. The simulation platform part that is responsible for this virtual time management is called the scheduler. This scheduler uses different approaches among which the most commonly used are the time-stepped, the event-driven and the mixed approaches.

We are interested in applications on personal computers and in that case, conventional time scheduling approaches may be inappropriate or may have limits depending on the situation. These limits could affect the performance of the simulation platform. To address that, (Payet et al., 2006) proposed the "temporality model" approach. This approach intends to fulfil a set of criteria that the classical approaches of scheduler do not meet. This paper clarifies the fundamentals of this temporality model approach, presented in an abstract way in (Payet et al., 2006) and proposes a practical application of it.

A set of requirements that we think an agent-based simulation scheduler should meet are listed in the next section. Then, we talk about the most commonly used time scheduling approaches and their limits. Next, we describe the temporality model approach and the performance advantages of this type of scheduler approach. Finally, we end with a practical implementation.

2 RELATED WORK

2.1 Requirements

(Payet et al., 2006) list a set of requirements that a scheduler should meet to be reasonably fit for multi-agent simulations:

1. **Take into Account the Specificities of the Simulated Model** (Helleboogh et al., 2004)**:**
   Models can take various and complex forms. It would be unrealistic to believe that a time management approach could work properly without any consideration of these simulated model features.
2. **Take into Account the Experimental Constraints:**
   A simulation often implies a shorter execution time than real time. Thus, depending on the complexity of the simulated model, the user can be

Ralitera, T., Aky, N., Payet, D. and Courdier, R.
Steps Towards a Balance between Adequacy and Time Optimization in Agent-based Simulations - A Practical Application of the Temporality Model Time Scheduling Approach.
DOI: 10.5220/0006904601600166
In *Proceedings of 8th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2018)*, pages 160-166
ISBN: 978-989-758-323-0

forced to reduce the simulation execution time by constraining the simulation platform. The more these constraints are stronger the less precise the results are. However, it is better than not getting any results at all.

3. **Establish a Homogeneous Management of Time:**
   Sometimes the user needs a flexible time scheduling approach that could adapt to time-stepped and to pure event-driven simulation contexts. In that way, all the results are obtained on the same experimental basis and can be compared (Axtell, 2000).

4. **Handle a Cumulative Characterization of Time:**
   Multidisciplinary complex phenomena simulation models could be developed by several modelers. Most of the time, none of these modelers have the full control over the model. Thus, it will be interesting to have a cumulative time scheduling that can be partially defined by each modeler, while remaining coherent.

5. **Handle an Incremental Complexity:**
   Currently, we manage to obtain complex structural and behavioural description models. However the time management is still basic. Consequently, the simulation results seem to be limited at a level that time dimension does not allow to cross (Amblard and Dumoulin, 2004). Thus, it could be interesting to have a time scheduling mechanism that could handle an incremental complexity.

6. **Minimize the Impact on the Execution Performance:**
   In our case, a good scheduling mechanism should work properly with the available computing power on personal computers.

In the following subsections, we describe the three most used types of scheduler approaches and how they fail to address some of these requirements.

2.2 The Time-stepped Approach

Because of its ease of implementation, the time-stepped approach is the most used approach in the agent-based simulations. In this approach, the scheduler advances the simulation time by incrementing its value by a fixed duration Δt called time-step (Fujimoto, 1998). The simulated time can be represented by an axis that is discretized by fixed intervals (figure 1). With each time-step, all the simulation activities (agent cycle and possibly objects simulation) are completed before advancing to the next step.

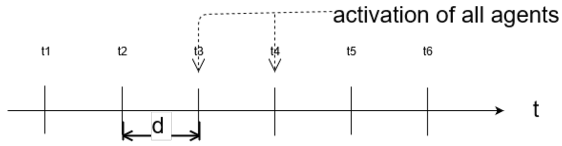

Figure 1: Time axis for time-stepped approaches.

This approach is usually easy to set up. Also, it is convenient for one specific model composed of agents that have homogeneous behaviour and the same activation frequency. However, it becomes unsatisfactory in the case of highly heterogeneous agents' behaviour. Indeed, using an inappropriate time step value can lead to lethargic or overactive agents. On one hand, if the agent is activated too infrequently, his actions seem slowed. On the other hand, if the agent is activated too often, his actions seem accelerated. The both cases may lead to erroneous simulation results.

To address that, a solution consists in setting a time step value that is equal to the smallest time interval required. Then, to avoid hyperactivity, the agents that require a bigger time step value have to explicitly become inactive during the intermediate time steps that are not relevant for them. The opposite approach is not possible. Indeed, an agent can slow down his activity, but he does not have the possibility of acting at a smaller granularity than that imposed by the scheduler. Consequently, when only a very small number of agents need a small time step value, the majority of the agents spend most of their time to become inactive. (Michel, 2004) concludes that using a regular discretization of time is unsatisfactory when the simulated model needs to take highly heterogeneous agent actions (from the frequency point of view) into account.

To summarize, this approach does not take the specificity of any simulated model into account.

2.3 The Event-driven Approach

In this approach, the simulation axis is continuous but the state of the system changes discretely at precise time called events (Anagnostou et al.,). An event can be defined as the description of the agents' behaviour activation conditions at a particular time. Its release date can be calculated depending on the nature of these conditions. Thus, the simulation consists in executing an orderly list of events. The time axis can be represented by a chained event list that are not equitably spaced (figure 2).

This approach is suitable in case of highly heterogeneous agents. However, the user of the platform does not have any control over the simulated time. He is not able to force the simulator to reduce the simulation execution time. However, for large-scale simu-

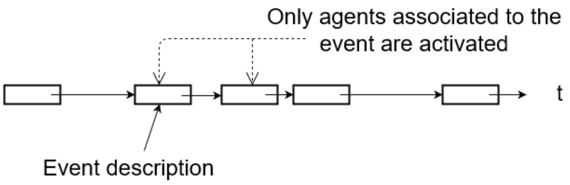

Figure 2: Time axis for event-driven approaches.

lations, this possibility of making compromises is essential.

Moreover, the simulated time can take very complex forms. Thus, the calculations made by the simulator can be rather substantial. Consequently, it could require lots of computational power.

2.4 The Mixed Approach

The mixed approaches propose to split the simulated model into sub-models. To each sub-model is associated the most appropriate type of scheduler. The resulting time management system is the combination of the different chosen schedulers. Consequently, there is no global time axis (figure 3).

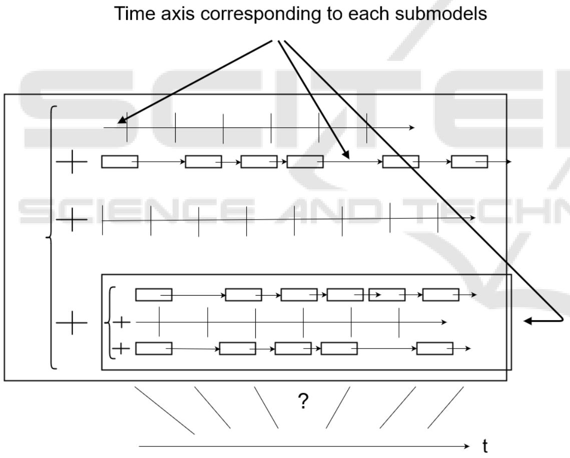

Figure 3: Time axis for mixed approaches.

This lack of global vision is a limit if we want to make analysis of the simulation time. Moreover, any attempt to influence the simulated time structure in order to reduce the execution time duration is prohibited.

2.5 The Temporality Model Approach

In multi-agent systems, an agent typically has some autonomy over the characterization of its state and its behaviour. In the same way, in the temporality model approach, the agent describes its own activation times itself. This kind of scheduler is focused on needs which are directly expressed by the agents.

The agents needs are expressed using a data structure called “the temporality”.

A temporality t specifies a point on the time axis where an agent wants to be activated. t can be defined by the tuple (Payet et al., 2006):

$$t = \{id, d, f, p, v\} \tag{1}$$

where:

- id is the identifier of the temporality.
- $[d, f]$ is the time interval during which the temporality can be activated.
- $[p]$ is the time period, i.e. the time interval between two executions of this temporality ($p = 0$ if the action is only executed once).
- v is the variability. It defines the accuracy below which the temporal occurence remains valid.

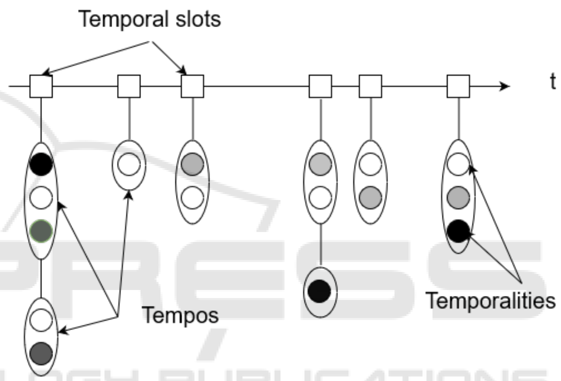

Figure 4: Time axis for the temporality model approach.

The agents behaviour activation time is equal to $x = d + p * k$, where k is an integer such as $0 \leq k \leq n$ and n is the biggest integer that verifies $(d + p * n) = f$ (Payet et al., 2006).

The agents define their temporalities during the simulation initialization. Afterward, they will be able to redefine or create new temporalities any time. In that case, the scheduler immediately processes the creations and modifications, then updates the time axis.

One particularity of the temporality model approach is the **ability to allow the user to influence the simulated time structure by adding time constraints**. These time constraints are the minimum time-step, the default time period and the variability.

- The minimum time-step Δt_{min}, indicates that two distinct activation dates should be separated by a duration at least equal to the value of Δt_{min}. If such a situation occurs during the analysis of the temporalities, the scheduler will use the variability parameter v of each of the temporalities to determine if they should be separated from each other or grouped together on the same date.

- The default time period value that is used by agents that do not define any temporalities during the initialization of the simulation.

The scheduler processes the temporalities using two data structures (see figure 4):

- The **temporal slot**: a precise time when the scheduler must activate the agent's behaviour.
- The **tempo**: a set of temporalities which are located on the same time slot and which have the same time period. This time period characterizes the tempo. It means that all the temporalities it contains have the same activation rhythm. The scheduler advances the simulated time from a time slot to another. At each time slot, all the tempos are processed. That produces the execution of all the behaviours associated with the temporalities they contain.

(Payet et al., 2006) propose the temporality model as a solution that fulfil the criteria cited in section 2.1:

1. By definition, the temporality model approach is based on the specificity of the simulated model.
2. The temporality model approach allows the user to influence the simulated time structure by adding time constraints. This is done by varying the minimum time step, the default period and the variability.
3. This approach allows for fine management of complex agents in term of diversity of activation rhythm, but does not complicate the simple agents management. The simple agents do not have to define any temporality and the system will automatically assign them a default one (defined at the beginning by the user). Thus, in the extreme cases, on one hand, if no agent defines a temporality, we automatically fall back into a time-stepped approach. On the other hand, if all the agents are complex and express a large number of temporalities, we end up with an event-driven type of scheduling.
4. For complex agents, the writing of the behavior can be done by different developers (Each specialist in his field). How then determine the global temporal operating mode of the agent? Who has the competence? The granularity of the temporality model approach is the activities. Consequently, each developer participates in the definition of the temporality of the agent as for the definition of its behavior.
5. In this approach, the agents temporal needs are expressed at the level of the agents' activities. It is possible to relate a specific temporality with each distinct activity. Also, the complexity of these temporalities can be accentuated at the same time that the behavior to which they are attached is made even more complex. Thus, this approach supports an incremental complexification in the same way as that which can be achieved at the level of writing the agents' behavior.
6. A way to implement a temporal model is the use of time slots and tempos. These can take the form of a linked list that manages the simulated time and that determine the agents' activation time. As we have experienced, and as has been shown by (Lawson and Park, 2000), such a data structure can be optimized with the algorithms of (Henriksen, 1983) so the resulting execution time remains around that of the time stepped approach.

(Payet et al., 2006) use assumptions and illustrative examples to demonstrate the advantages of the use of the temporality model approach. They made a comparative study between the different classical approaches and the temporality model. In this article, we support their demonstration with a practical application.

3 CASE STUDY

The proposed case study is about agent-based simulation of individual transport. This is done for the city of Saint-Denis, capital of Reunion Island, a French island in the Indian Ocean.

In this simulation model, agents are vehicle owners. The model uses activity-based approach so agents move from a place to another following their activity schedule. Thus, travels result from personal activities (work, shopping, leisure, going home) that individuals need or wish to perform.

3.1 The Agent Model

In our simulation model, an agent is a vehicle owner. His goal is to achieve all his activities using the vehicle at his disposal. The activity profile AP_i for each group i of vehicle owners is defined with a list of 3-tuples:

$$AP_i = (ACT_j, MDT_j, PD_j) \quad (2)$$

Where ACT_j represents the activity j, MDT_j the mean departure time, and PD_j the probability of departure for the activity j.

3.2 The Environment Modeling Elements

The environment in which the agents are situated and act is represented by a collection of entities. These entities represent fragments of the real environment. Its definition is done using Geographic Information System (GIS) composed of the following layers:

- The building layer that is a set of polygon entities and that represents administrative boundaries.
- The road layer that is composed of polyline entities and that represents the road network along which the agents move.
- The areas of interests layers, represented by polygon or point entities. They represent the location of the home, the workplace, the leisure, the commercial or the industrial places.

The simulation model takes shape files and statistical data as input. They are used at the different level of the simulation model such as for environment modelling or for the calculation of the population distribution.

In the following sections we will show how we built this simulation model upon the SimSKUAD simulation platform.

4 THE SKUADCityModel

4.1 SKUAD

SKUAD stands for “Software Kit for Ubiquitous Agent Development”. This free multi-platform toolkit is still being developed. It allows us to create multi-agent systems using Java language. It has been developed since 2013 by the Collective Adaptive Systems Research Group in the Laboratory of Mathematics and Computer Science (LIM) at the University of Reunion Island. The idea comes from the observation of the outburst of the Internet of Things, and the necessity to have a software that can make this mass of objects more consistent. For that purpose, SKUAD can operate in an ambient (agents can operate in our real environment) and in a simulated mode.

4.2 The SimSKUAD Simulation Platform

The SimSKUAD is the simulated operating mode of SKUAD. In this mode, the agents evolve following a simulated time. Devices constituting the agent’s environment are virtual. The SimSKUAD uses the temporality model as a default time scheduling approach. However, its architecture is flexible enough to allow us to easily replace the scheduler. Thus, we were also able to implement the time-stepped approach.

Different optional modules have already been developed for SimSKUAD. Examples are Mod2D allowing agents simulations in a continuous space environment or ModGrid which allowing agents simulations in a discrete space environment. More details can be found on the SKUAD website (Payet, 2018).

In this paper, we focus in a module called **ModGIS** that allows agents to evolve in Geographical Information System (GIS) kind of environment. For that, ModGIS uses the GeoTools library (Turton, 2008).

The SKUADCityModel is built upon the ModGIS module of the SimSKUAD simulation platform. In our experiments, we implemented two different types of scheduler: a time-stepped approach and a temporality model approach. In the next section, we compare the experimental results obtained from these two types of approaches.

5 COMPARISON OF THE DIFFERENT APPROACHES

In this section, we illustrate the advantages of the temporality model approach against classical approaches by experiments on the SKUADCityModel. For that purpose, we make two types of manipulations:

- The first manipulation is to **vary the experimental constraints (time step duration and the variability)** from 0% to 100%. In this way, we show how the simulation duration can be reduced depending on the used time scheduling approach.
- The second one is **a scaling up test**. For that purpose, we vary the number of agents from 1000 to 10,000 moving moving over simulated 12 hours. In this way, we show how the simulation model can scale up depending on the used time scheduler approach.

Remark: We chose to not make implementation of the event-driven or the mixed approaches because they are not appropriate if we want the users to have control over the simulated time (see section 2.1).

For this experiment, we run the two simulation models on a personal computer with the following configuration : fifth generation Intel core i5, 16 gigabytes of RAM, Solid State Drive.

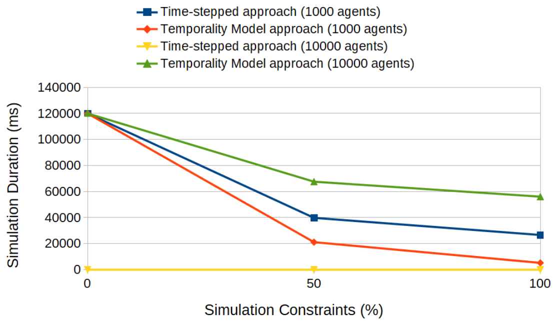

Figure 5: Execution duration performance of each scheduling mode for 1000 then 10000 agents.

Figure 5 shows the results of the first experiments we carried.

First, in the case where no experimental constraint is applied (0%) and for 1000 agents, we can see that the simulation execution durations are almost the same for the two approaches. However, when we vary the experimental constraints (50%, 100%), we can see a sharp decrease of the simulation execution duration. This decrease is greater in case of the use of the temporality model approach.

Second, we note a difficulty of scaling up in case of the time-stepped approach. Indeed, the results show that when using the time-stepped approach, the maximum number of agents supported by the SKUADCityModel does not exceed 10,000. In the figure 5, the yellow curve on 0ms indicates a crash due to an out of memory error. This is a problem we do not encounter, at least up to 10,000 agents, when we use the temporality model approach.

Finally, the performances remain acceptable and the simulation execution times are less than five minutes. These results are in line with the demonstrations made in (Payet et al., 2006). That shows the performance advantages of this type of scheduler approach. Especially, it allows the user to have control over the simulated time. Moreover, it allows to scale while maintaining acceptable performance.

6 CONCLUSION AND FURTHER WORK

Agents scheduling management is a critical process in simulation platforms. Unfortunately, all the conventional approaches have limits in some situations. From case to case, these limits could be very restrictive. (Payet et al., 2006) proposes the temporality model approach as a possible solution that addresses some of these limits. They demonstrate that by illustrative examples based on theoretical assumptions.

In this paper, we show that the temporality model meets requirements that we think agent-based simulation of urban and transport system should fulfil:

- It **takes into account the specificities of the simulated model** and can automatically adapt to them. Indeed, in the case of homogeneous agents, the temporality model works like a time-stepped approach. In the case of highly heterogeneous agents, it works like an event-driven approach.
- It **takes into account the experimental constraints** as it allows to vary a set of parameters in order to shorten the simulation execution duration for example.
- It ensures a **homogeneous management** and a**cumulative characterization of time** because all the agents' actions are based on the same virtual timeline.
- It activates agents who need to be awake only when they need to be. Consequently, it **handles an incremental complexity** and **minimizes the impact on the execution performance**.

We support these assumptions with practical applications. For that, we implemented the time-stepped approach and the temporality approach in a same simulation model called the SKUADCityModel. Then, we compared the two implementations based on the execution time performance and the ability of the simulation model to scale.

The results are in agreement with the theoretical demonstrations that have already been made in (Payet et al., 2006). In our experiments, we used the SimSKUAD simulation platform, that is still under development and that has an architecture that seems to be different compared to classic agent-based simulation platform. As further work, it could be interesting to see if the architecture of the SimSKUAD simulation platform also affects the performance of the simulation.

The demonstration done in this paper is limited to a comparison with the classical time management approaches. However, the relevance of the temporality model approach should be further assessed with more advanced mechanisms such as the time-stepped load balancing approach described in (Wu et al., 2015) In addition, it would be interesting to increase the performance already obtained using the GPU calculation, as proposed in the paper (Song et al., 2017).

ACKNOWLEDGEMENTS

The research in this paper was supported by the Région Réunion, the L'Oréal-UNESCO for women in science fellowship and the town of Saint-Denis. The authors thank the reviewers for their comments.

REFERENCES

Amblard, F. and Dumoulin, N. (2004). Mieux prendre en compte le temps dans les simulations individus-centrées. *11èmes Journées de Rochebrune*, pages 13–26.

Anagnostou, A., Meskarian, R., Robertson, D., and Fakhimi, M. Introduction to discrete-event simulation: how it works. In *Welcome to the 2018 Operational Research Society Simulation Workshop (SW18)*, page 13.

Axtell, R. L. (2000). Effects of interaction topology and activation regime in several multi-agent systems. In *Multi-Agent-Based Simulation, Second International Workshop, MABS 2000, Boston, MA, USA, July, 2000, Revised and Additional Papers*, pages 33–48.

Fujimoto, R. M. (1998). Time management in the high level architecture. *Simulation*, 71(6):388–400.

Helleboogh, A., Holvoet, T., Weyns, D., and Berbers, Y. (2004). Extending time management support for multi-agent systems. In *Multi-Agent and Multi-Agent-Based Simulation, Joint Workshop MABS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*, pages 37–48.

Henriksen, J. O. (1983). Event list management - a tutorial. In *Proceedings of the 15th conference on Winter simulation, WSC 1983, Arlington, VA, USA, December 12-14, 1983*, pages 543–551.

Lawson, B. G. and Park, S. (2000). Asynchronous time evolution in an artificial society model. *J. Artificial Societies and Social Simulation*, 3(1).

Michel, F. (2004). *Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents. (Formalism, tools and methodological elements for the modeling and simulation of multi-agents systems)*. PhD thesis, Montpellier 2 University, France.

Payet, D. (2018). Official website of skuad. http://skuad.onover.top/, Last accessed on 2018-15-07.

Payet, D., Courdier, R., Ralambondrainy, T., and Sébastien, N. (2006). Le modèle à temporalité: pour un équilibre entre adéquation et optimisation du temps dans les simulations agent. In *Systemes Multi-Agents, Articulation entre l'individuel et le collectif - JFSMA 2006 - Quatorzieme journees francophones sur les systemes multi-agents, Annecy, France, October 18-20, 2006*, pages 63–76.

Song, X., Xie, Z., Xu, Y., Tan, G., Tang, W., Bi, J., and Li, X. (2017). Supporting real-world network-oriented mesoscopic traffic simulation on GPU. *Simulation Modelling Practice and Theory*, 74:46–63.

Turton, I. (2008). *Open Source Approaches in Spatial Data Handling*. Springer, Berlin, Heidelberg.

Wu, Y., Song, X., and Gong, G. (2015). Real-time load balancing scheduling algorithm for periodic simulation models. *Simulation Modelling Practice and Theory*, 52:123–134.