HAL
open science

Activity Regions for the Specication of Discrete Event Systems

Alexandre Muzy, Luc Touraille, Hans Vangheluwe, Olivier J.J. Michel, Mamadou Kaba Traoré, David R.C. Hill

To cite this version:

Alexandre Muzy, Luc Touraille, Hans Vangheluwe, Olivier J.J. Michel, Mamadou Kaba Traoré, et al.. Activity Regions for the Specication of Discrete Event Systems. Symposium On Theory of Modeling and Simulation - DEVS Integrative M&S Symposium (DEVS'10), Apr 2010, United States. pp.176-182. hal-00465463

HAL Id: hal-00465463

https://hal.science/hal-00465463v1

Submitted on 19 Mar 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Activity Regions for the Specification of Discrete Event Systems

Alexandre Muzy[1], Luc Touraille[2], Hans Vangheluwe[3],
Olivier Michel[4], Mamadou Kaba Traoré[2], David R.C. Hill[2]

March 2, 2010

[1]LISA UMR CNRS 6240, Università di Corsica – Pasquale Paoli, Campus Grossetti, BP 52, 20250 Corti - France.

[2]LIMOS UMR CNRS 6158, Blaise Pascal University BP 10125, AUBIERE Cedex - France.

[3]Department of Mathematics and Computer Science, University of Antwerp, Middelheimlaan 1, Antwerp, Belgium.

[4]LACL - Bât P2 - 240, Université de Paris XII, Faculté des Sciences et Technologies, 61 Av. du Général de Gaulle, 94010 Créteil Cedex - France.

Abstract

The common view on modeling and simulation of dynamic systems is to focus on the specification of the state of the system and its transition function. Although some interesting challenges remain to efficiently and elegantly support this view, we consider in this paper that this problem is solved. Instead, we propose here to focus on a new point of view on dynamic system specifications: the *activity* exhibited by their discrete event simulation. We believe that such a viewpoint introduces a new way for analyzing, modeling and simulating systems. We first start with the definition of the key notion of activity for the specification of a specific class of dynamic system, namely discrete event systems. Then, we refine this notion to characterize activity regions in time, in space, in states and in hierarchical component-based models. Examples are given to illustrate and stress the importance of this notion.

1 Introduction

Complicated structures of simulation models consist of a large number of components with many intense interactions. It is not easy to extract abstractions of the dynamics of the whole system, during, before or after its simulation. The analysis of the many outputs and interactions is long and meticulous. As far as we know, no established methods exist for finding patterns of interactions in system structures, during a simulation. Some methods exist for particular domains (multi-agent systems, distributed and parallel simulations, image analysis, etc.), but except the work proposed by [10], no generic methods have been developed for this purpose.

In the simulation context, *activity* is usually used as a phase of the system under study (*e.g.*, activities of a customer in a shop are: *waiting*, *payCashier*, etc.) [9]. We do not consider this definition of activity here. Instead, activity is considered as a measure of the number of events occurring during a simulation. We believe that this *new definition of activity* can be used as a central guiding concept to construct generic structures for the analysis and specification of systems. The specification structures, driven by a measure of activity of the simulation, can be used to faithfully chart the dynamics of sub-components in time, space, and states. Inactive and active regions may also be specified. Using activity, states and components corresponding to systems can thus be dynamically, structurally and behaviorally specified. For example, one can imagine functional magnetic resonance image analysis of a brain. The detetection of neural spikes are used for an activity-based

structural determination of behavioral brain regions. These structures and behabiors are highly dynamical, according to the activity exhibited.

In this paper, the usual notions of time, space, states and components are reconsidered from an activity-based point of view for the discrete event specification of systems [11]. Our goal is to provide a new definition of activity. Benefits from using this new definition are expected to be twofold: (i) Optimizing system specifications and related simulator architectures, and (ii) Providing guidance to designers for modeling and simulating systems .

This article introduces mathematical notations for dynamic systems and how activity can be used for the analysis and the specification of these systems, using discrete events (Section 2). The new notion of activity regions is then presented (Section 3) and applied to components (Section 4) before a description of related works and a conclusion.

2 Activity tracking in discrete event system specifications

Dynamic systems can be described by mathematical structures. A discrete event specification of systems can then be achieved. Activity is restrained here to discrete event system specifications and related to event frequency.

2.1 Dynamic system specification

A dynamic system (or DS in short) corresponds to a phenomenon that evolves over time, within some context. The phenomenon is part of a system characterized by *observables*. The observables are called the *variables* of the system (and are linked by some relations). The value of the variables evolves over time. The collection of the values of the variables that describe the system constitutes its *state*. The state of a system is an observation at a given instant. The temporal sequence of state changes is called the *state trajectory* of the system.

Let Q be the state space of a DS. We denote $q \in Q$ its current state. The transition to the next state is given by the transition function $\delta : Q \to Q$. Let q be the value of the current state (at the event time t), the value of q after the transition is $q' = \delta(q)$ [at the event time $t + \Delta$, for $t \in \mathcal{T}$, where $\mathcal{T}$ is the time base (discrete or continuous)]. In previous notation, time is implicit, to make time explicit, such a transition can be written as $q(t') = \delta\left(q\left(t\right)\right)$, where $t' = t + \Delta$.

2.2 Activity of event sets

In a discrete event simulation, the dynamics of a system is represented by a chronological sequence of events. An event affects the system at a given time and possibly carries additional information, such as a value, an operation to perform, etc. Consequently, we denote an event ev_i by a couple (t_i, v_i), where t_i is the timestamp of the event, and v_i is the information associated to the event. The event set is defined as $\xi = \{ev_i = (t_i, v_i) \mid i = 1, 2, 3, ...\}$.

Let's consider first the basic usual and transversal definitions of the notions of *activity*, *event*, and *process*. An activity "is what transforms the state of a system over time" [3]. It begins with an event and ends with another. An event is also considered to cause a change in the state of a component. A process "is a sequence of activities or events ordered in time" [3].

We do not consider here activity as a phase of a system. We define activity as a measure of the number of events in an event set. Formally, we define the event-based activity measure $\nu_H(t)$ as a function of time that provides the activity in a discrete event simulation, from t over a given time horizon H:

$$\nu_H(t) = \frac{|\{ev_i = (t_i, v_i) \in \xi \mid t \leq t_i < t + H\}|}{H}$$

Activity is a measure of the event rate, or event frequency, in an event set. The qualitative differences of influence of events on the state of the dynamic system is voluntarily neglected here. Only the quantity of events over a period of time is taken into account. For example, assuming the event trajectory depicted in Figure 1, the activity of the system corresponds to the following values for different time horizons: $\nu_{10}(t) = 0.3$, $\nu_{20}(t) = 0.15$, $\nu_{30}(t) \simeq 0.133$, $\nu_{40}(t) = 0.175$.

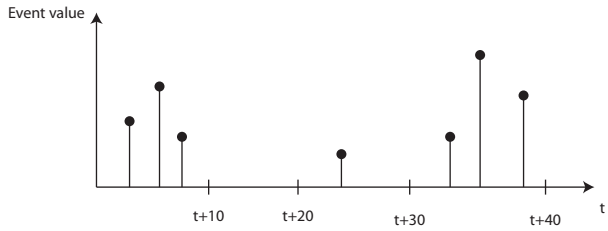

Figure 1: An example of event trajectory.

For the sake of simplicity, we will denote the activity measure $\nu(t)$ (making implicit the dependency on the time horizon H).

2.3 Activity state in discrete event system specifications

We start here with the specification of a basic activity-based DS, through discrete-events. This system is merely a model of a DS embedding an activity state based on the activity measure introduced in section 2.2. Remember that this measure merely constitutes a counter of events, without the information of events (as presented in [1], for example).

Activity states, $Q_A \subseteq Q$, can be attributed todiscrete event system specifications to encode the activity level of simulation levels, according to their reception/scheduling (or not) of discrete events. In their simplest form, activity states are: $Q_A = \{active, inactive\}$.

A mean-time activity (TA) function can be defined as: ρ_{TA}:$\mathbb{R} \rightarrow Q_A$.

More precisely we have:

$$\begin{cases} q_A(t) & = & \rho_{TA}(\nu(t)) & = & inactive & \text{if } \nu(\text{t})=0 \\ q_A(t) & = & \rho_{TA}(\nu(t)) & = & active & \text{otherwise} \end{cases}$$

2.4 Activity for discrete event system specifications in Cartesian coordinates

The Cartesian coordinate space is defined as a set of references: $\mathcal{P} = \{(x_1, \ldots, x_n) \mid x_i \in \mathbb{R},\ i \in \mathbb{N}\}$. A spatial state is thus defined as $q\,(p) \in Q \times \mathcal{P}$. Spatially referenced states can be considered as a refinement of the set of states Q. Interactions can be noted as: $q(p_i) = \delta\,(q\,(p_{j \in N_i}))$, where N_i corresponds to the set of neighborhood positions of i (possibly including the self-position i): $N_i \subset \mathbb{N}$. A state in space and time is defined as $q\,(p,t) \in Q \times \mathcal{P} \times \mathcal{T}$. Notice that, considering a single self-neighborhood: $N_i = \{i\}$, leads to the following simplification: $q(p_i, t') = \delta\,(q\,(p_i, t))$ and $q(t') = \delta\,(q\,(t))$. That is, our spatiotemporal notation is consistent with the temporal one.

A mean-space activity (SA) function can be defined as: ρ_{SA}:$\mathbb{R} \rightarrow Q_A$.

More precisely we have:

$$\begin{cases} q_A(p,t) & = & \rho_{SA}(\nu^p(t)) & = & inactive & \text{if } \nu^p(\text{t})=0 \\ q_A(p,t) & = & \rho_{SA}(\nu^p(t)) & = & active & \text{otherwise} \end{cases}$$

Figure 2 depicts the different activity regions in space.

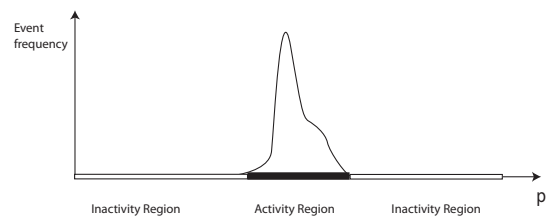

Figure 2: Activity in space.

A refinement of the activity structures definition can be achieved through the notion of activity regions.

3 Definition of activity regions

A formalization of the activity notion must be provided before being able to study it thoroughly. In this section, we propose several mathematical structures for describing the activity of systems, going from particular cases to more general notations. From the modeler's perspective, the notion of activity as such is not usually explicitly described. Most of the time, we want to know which parts of the system are active and which parts are not. Therefore, *activity regions* can be used at a high level of abstraction to describe elements of a discrete event system specification as active or inactive.

3.1 Activity regions in time

The activity measure is used to determine the subregions of the time base $\mathcal{T}$ through:

- Activity region in time:

$$\mathcal{AR}^{\mathcal{T}} = \{t \in \mathcal{T} \ \mid \nu(t) > 0\}$$

- Inactivity region in time:

$$\overline{\mathcal{AR}^{\mathcal{T}}} = \{t \in \mathcal{T} \ \mid \nu(t) = 0\}$$

Considering the chronological nature of time and that every element of the time-base can be defined as active or inactive, an activity-based partitioning of time base $\mathcal{T}$ is thus achieved: $\mathcal{T} = \mathcal{AR}^{\mathcal{T}} \cup \overline{\mathcal{AR}^{\mathcal{T}}}$.

3.2 Activity regions in states

The activity measure is used to determine the subregions of the state set Q:

- Activity region in states:

$$\mathcal{AR}^{Q}(t) = \{q \in Q \ \mid \nu(t) > 0\}$$

- Inactivity region in states:

$$\overline{\mathcal{AR}^{Q}}(t) = \{q \in Q \ \mid \nu(t) = 0\}$$

We consider now the function of reachable states in time as $q : \mathcal{T} \to Q$. We can define now the set of all reachable states in the state set Q, through time, named the *universe* and noted $\mathcal{U} = \{q\,(t) \subseteq Q \mid t \in \mathcal{T}\}$. Considering that all reachable states in time can be active or inactive, an activity-based partitioning of the state set Q can be achieved: $Q = \mathcal{AR}^{Q} \cup \overline{\mathcal{AR}^{Q}}$.

3.3 Activity regions in Cartesian coordinates

The activity measure is used to determine the subregions of the Cartesian coordinate space (as defined in 2.4) through:

- Activity region in space:

$$\mathcal{AR}^{\mathcal{P}}(t) = \{p \in \mathcal{P} \ \mid \nu^{p}(t) > 0\}$$

- Inactivity region in space:

$$\overline{\mathcal{AR}^{\mathcal{P}}(t)} = \{p \in \mathcal{P} \ \mid \nu^{p}(t) = 0\}$$

We consider now the function of reachable states in time and space as $q : \mathcal{P} \times \mathcal{T} \to Q$. We can define now the set of all reachable states in the state set Q, through time and space, through the *universe* $\mathcal{U} = \{q\,(p,t) \subseteq Q \mid p \in \mathcal{P}, t \in \mathcal{T}\}$.

Considering that all reachable states in time and space can be active or inactive, an activity-based partitioning of $\mathcal{P}$ can be achieved: $\forall t \in \mathcal{T}$, $\mathcal{P} = \mathcal{AR}^{\mathcal{P}}(t) \cup \overline{\mathcal{AR}^{\mathcal{P}}(t)}$.

Figure 3 depicts activity values for two-dimensional Cartesian coordinates $X \times Y$. This is a neutral example, which can represent whatever activity measures in a Cartesian space (fire spread, brain activity, etc.)

3.4 Activity referenced states

For the set of states Q, we consider here that:

$$Q = \prod_{i=0\ldots n} E_i$$

where E_i can be any set, and n is the number of sets. For example, the model of a leaf could include its area in cm^2 (a real number), its age in days (a natural number) and the amount of energy received from sunlight in Watts per meter (a real number). Hence, the state set of this model would be $S = \mathbb{R} \times \mathbb{N} \times \mathbb{R}$, and a possible state would be $s = (68.2, 20, 381.5)$.

Now, we reference states through activity. Activity references constitute a viewpoint of the state set where only the variables *relevant* for activity are considered.

Formally, we define the set of activity referenced states $\mathcal{G}_I$ as a projection of the state space Q onto indexes $I \subseteq \{1, ..., n\}$:

$$\mathcal{G}_I = \pi_I(Q) = \prod_{i \in I} E_i$$

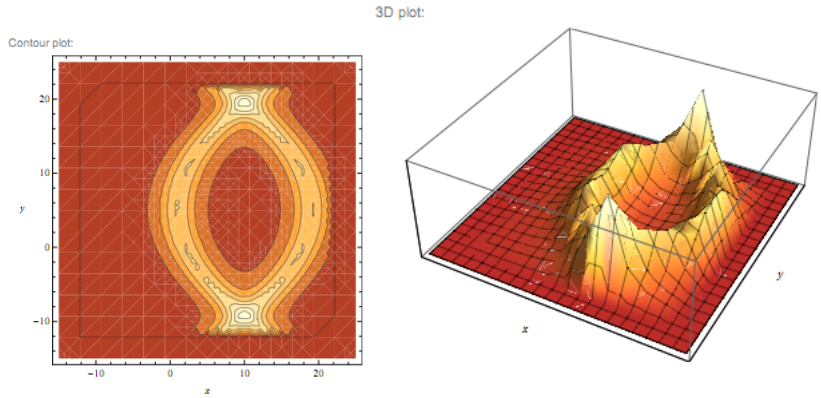

Figure 3: 2D and 3D visualization of activity level in a 2D space. x and y represent Cartesian coordinates. The activity amplitude (real value), of each coordinate, is represented in the third dimension.

The projection operator π is used to "select" a subset of the state elements[1]. $I \subseteq \{1, ..., n\}$ is the set of indexes denoting the elements of interest. For a given model, the set of activity referenced states can vary depending on which states are selected for activity indexing. In the previous leaf example, active leaves can be defined as being the ones that are younger than *100* days. In this case, the only activity referenced state of interest is the age of the leaf. Therefore, the following set of activity referenced states are used: $\mathcal{G}_2 = \pi_2(Q) = \mathbb{N}$. However, active leaves are defined as being the ones that receive enough energy to grow (depending on their area and the energy received), the set of activity referenced states will be $\mathcal{G}_{1,3} = \pi_{1,3}(Q) = \mathbb{R} \times \mathbb{R}$. The activity measure is used to determine the sub-regions of the generalized activity regions through:

- Activity region in activity referenced states:

$$\mathcal{AR}^{\mathcal{G}_I}(t) = \{g \in \mathcal{G}_I \mid \nu^g(t) > 0\}$$

- Inactivity region in activity referenced states:

$$\overline{\mathcal{AR}^{\mathcal{G}_I}}(t) = \{g \in \mathcal{G}_I \mid \nu^g(t) = 0\}$$

[1]In the context of relational algebra, the projection could be defined using attribute names instead of indexes.

Considering that all reachable states in the set of activity referenced states $\mathcal{G}_I$ are active or inactive, and that all unreachable states are inactive, an activity-based partitioning of $\mathcal{G}_I$ can be achieved: $\forall t \in \mathcal{T}$, $\mathcal{G}_I = \mathcal{AR}^{\mathcal{G}_I}(t) \cup \overline{\mathcal{AR}^{\mathcal{G}_I}}(t)$.

The computation of activity referenced states can be automated through the following steps: (i) Select all states $q \in Q$ *relevant* for activity, (ii) Copy these new states in the set of activity referenced states, and (iii) Compute the activity regions for every activity referenced state $g \in \mathcal{G}_I$, *i.e.*, those satisfying $\nu^g(t) > 0$.

By restricting the states of the model to activity referenced states, the specification of activity regions becomes straightforward. Activity regions can be used to map the activity of the real system. Besides, an hypothetical "activity-aware simulator", more efficient, can be developed to track and focus computations on active states.

We end up here with an universe of elements of reachable activity referenced states $g \in \mathcal{G}_I$: $\mathcal{U}_{\mathcal{A}} = \{g \in \mathcal{G}_I \mid \mathcal{G}_I = \pi_I(Q), I \subseteq \{1, ..., n\}\}$. It can be noticed that the definition of activity regions in states given in 3.2 corresponds to a particular case where $I = \{1, ..., n\}$.

Let's consider now a simple application example of fire spreading. Using activity referenced states, we can model very simply the activity regions of a fire

spreading. Assume the fire model describes the state of a cell with the following states:

- $x \in \mathbb{R}$ and $y \in \mathbb{R}$;
- $status \in \{burnt, burning, safe\}$;
- $type \in \{tree, bush, water, road\}$;
- $heat \in \mathbb{R}$.

A simple model of the activity regions can involve the status and the type of the cell. Formally, the set of activity referenced states would be $\mathcal{G}_{2,3}$. Assuming the activity map depicted in Figure 4, the resulting activity region specification would be:

$$AR^{G_{2,3}}(t) = \{\{burning, safe\} \times \{tree, bush\}\}, \forall t \in \mathcal{T}$$

Figure 4: Activity measures based on two activity references (status and type). *E.g.*, *Bushes* in a *burning* state have an activity of *0.8*.

4 Activity in component-based models

The modeling of a system can often be eased by breaking it down into several subsystems (top-down approach), or by assembling existing subsystem models into a larger one (bottom-up approach). This leads to *component-based* models that describe systems as sets of components, along with the way they interact with each other. We propose in the following an extension of the notion of activity to this type of hierarchical models.

4.1 Activity in a single composite model

The activity of a composite model depends on the activity of its components, but also on the interactions between the components. In the case of simple systems, the activity of a model $\mathcal{M}$ composed of components $C = \{c_1, \ldots, c_n\}$ might be approximated by summing the c_i activity measures:

$$\nu_{\mathcal{M}}(t) = \sum_{c \in \mathcal{C}} \nu_c(t)$$

4.2 Activity regions in composite models

In previous sections, we successively defined activity regions as sets of instants, sets of spatial (Cartesian) coordinates, states, and as sets of activity referenced states being a projection of the set of states in which activity is suppused to occur. These mathematical structures are useful to model activity in simple — not composed — models. In composite models, we must take into account that each subsystem is itself a model. We provide here a new definition of activity regions in composite models, noted $AR_H^{\mathcal{C}}(t)$, which is the set of sub-components whose level of activity is larger than zero, at a given time t and for a given horizon H[2]:

$$\mathcal{AR}^{\mathcal{C}}(t) = \{c \in \mathcal{C} \mid \nu_c(t) > 0\}$$

Once again, we will extend this definition to ease the specification of activity regions.

4.3 Activity regions in time for composite models

Over time, a component can be active or inactive. The periods of time for which a component c is active is specified using an activity region in time $\mathcal{AR}_c^{\mathcal{T}}$,

[2]For the sake of brevity, we omit in this section the definition of inactivity regions, which can be easily deduced from the active ones.

as presented in sub-section 3.1. Using the activity regions of the components, we define the overall activity region of the composite model as:

$$\mathcal{AR}^{\mathcal{C}}(t) = \left\{c \in C \mid t \in \mathcal{AR}_c^{\mathcal{T}}\right\}$$

In other words, the activity region of the hierarchical model at time t is the set of components whose activity region in time contains t.

4.4 Activity regions in Cartesian coordinates for composite models

In spatialized models[3] components are localized into a Cartesian coordinate space $\mathcal{P}$. Each component c is assigned to a position $c_p \in \mathcal{P}$. Applying the definition of activity regions in space (presented in section 3.1) to components, we obtain:

$$\mathcal{AR}^{\mathcal{C}}(t) = \left\{c \in \mathcal{C} \mid c_p \in \mathcal{AR}^{\mathcal{P}}(t)\right\}$$

$\mathcal{AR}^{\mathcal{P}}(t)$ specifies the coordinates where activity occurs. Consequently, active components correspond to the components localized at positions p.

4.5 Activity state references for composite models

Denoting activity regions only through spatial coordinates can be rather restrictive: Not all models are spatialized, far from it. Moreover, even in spatialized contexts, active components can often be identified using the states of the components but not using their position. As we did previously in section 3.4, we broaden the notion of activity regions to the entire state set. Each component in the model has a state $q_c \in Q$. For the component to be active, this state must match one of the elements of the activity region $\mathcal{AR}^{\mathcal{G}_{\mathcal{I}}}$, meaning that the elements of the state that are activity referenced states must belong to $\mathcal{AR}^{\mathcal{G}_{\mathcal{I}}}$. Formally, we obtain the following definition:

$$\mathcal{AR}^{\mathcal{C}}(t) = \left\{c \in \mathcal{C} \mid \pi_I(q_c) \in \mathcal{AR}^{\mathcal{G}_{\mathcal{I}}}(t)\right\}$$

[3] A model is said to be *spatialized* when the phenomenon under study has a spatial extension. This requires that states have a richer structure than just scalar values to cope with the discretization of a spatially embedded phenomenon. Examples of spatialized models include cellular automata and L-systems.

4.6 Extension to the component types

In a composite model, all components do not necessarily have the same type. An hypothetical plant model can be composed of leaf, stem and root models. To allow the activation or deactivation of heterogeneous components, we need to take their types into consideration in the definition of activity regions. A composite model with heterogeneous sub-models aggregates a set of components $\{c_{11}, c_{12}, \ldots, c_{1k}, c_{21}, c_{22}, \ldots, c_{ij}\}$ where $c_{m1}, \ldots, c_{mn}$ are of type T_m. Components of different types have different state sets. Therefore, separate activity referenced states must be provided for every component type. To reflect this, we generalize the previous definition of the activity region of a composite model to:

$$\mathcal{AR}^{\mathcal{C}}(t) = \left\{c_{mk} \in \mathcal{C} \mid \pi_I^{T_m}(q_{c_{mk}}) \in \mathcal{AR}_{T_m}^{\mathcal{G}_{\mathcal{I}}}(t)\right\}$$

By using separate activity regions for each type of components, an entire type set of components can be deactivated. For example, if the leaves components have to be deactivated during the night (because they do not receive any energy from the sun), we can specify $\mathcal{AR}_{leaves}^{\mathcal{G}_{\mathcal{I}}}(t) = \emptyset$ when t belongs to the nighttime (remember that activity regions are functions of time, and therefore can be dynamic).

5 Related works

As pointed out in [2], the notion of *activity* presented here is a very generic term which can be applied to a variety of different topics in computer science. This notion of activity is different from the notion used usually in simulation. The usual activity notion can be found in Tocher [9], who also first described the three phase approach, as an optimization of an activity-based simulation. In [3], Balci presents the concept of *activity* as a possible approach to drive the implementation of a discrete event simulation kernel. An object-oriented variant of the three phase approach was introduced by Pidd [14].

In many fields, the notion of *activity* can be found. For example, it is a fundamental issue in *computer graphics*, from Z-buffers [4], to current work required

for fast rendering of different level of details [5, 6] in complex scenes or multiresolution modeling in game engine [15]. In *autonomic systems* [8, 16], ensuring the persistance of the self-$*$ properties requires a feedback loop based on tracking certain variables that account for activity changes in the system, from the level of the operating system (*e.g.*, in Solaris 10) to the level of large cloud-based systems. In *everyware/ambiant/pervasive/ubiquitous systems* [17], the key issue is to track the activity/location of a user to adapt local devices to the presence/absence and movement of the user's activity. Nowadays, any parallel system copes with dynamic requirements for resources using *load-balancing* [12] algorithms to track the activity taking place in each computing sites to reallocate and reschedule tasks according to changes in both the demands and the availability of resources. In *dynamic systems*, the notion of activity is a key notion since, which, in some contexts, can lead to structure changes of the state space as coined by [7], with the notion of *dynamic systems embedding a dynamic structure*. An attempt to quantify and formalize a simulated system activity has been proposed in [13] for model exploration. Using a thermodynamical approach functions characterized the activity and the speed of evolution of a system. This approach enhances the analysis of the trajectories of a system, facilitating the identification of cyclic, stationary or chaotic behaviors.

While the concept of activity is found in many fields, very few address activity explicitly, as we did here, through a precise definition. Even if we believe that the work started here still needs to be worked-out.

6 Conclusions

In this paper, we have introduced a new definition of the notion of (simulation) *activity*. All definitions constitute new aspects of dynamic systems through a discrete event specification. These aspects focus on the possibly "changing" elements of a system. Possible changes correspond to the occurrence of discrete events, *i.e.*, whithout considering the effective impact of event occurences for state changes. It is expected that, for correct implementations and models, these discrete event occurrences represent an efficient mapping of a real-world dynamic phenomenon. Several region-based extensions of activity have been proposed through several elements: time, space, activity references and components. We hope that this first work sketches many perspectives to deal with activity (the definition of activity rates, levels, changes in state, etc.)

Acknowledgements

The authors would like to thank all participants of the Cargese seminar (http://msdl.cs.mcgill.ca/conferences/Cargese/2009/): Bernie Zeigler, Xiaolin Hu, Jean-Pierre Briot, Patrick Coquillard, Franck Varenne, Levent Yilmaz, Philippe Caillou, and Guillaume Deffuant. Many ideas emerged from our fruitful discussions!

This research is supported by the CNRS.

References

[1] A. Muzy, J.J. Nutaro, B.P. Zeigler, P. Coquillard. Modeling and simulation of fire spreading through the activity tracking paradigm. *Ecological Modelling*, 219(1):212 – 225, 2008.

[2] Salil R Akerkar. Analysis and visualization of time-varying data using the concept of 'activity modeling'. Master's thesis, Dept. of Electrical and computer engineering - University of Arizona, 2004.

[3] Osman Balci. The implementation of four conceptual frameworks for simulation modeling in high-level languages. In *WSC '88: Proceedings of the 20th conference on Winter simulation*, pages 287–295, New York, NY, USA, 1988. ACM.

[4] Edwin E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. PhD thesis, Dept. of CS, U. of Utah, December 1974.

[5] J. Clark. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 19(10):547–554, 1976.

[6] T. A. Funkhouser and C. H. Sequin. Adaptive display algorithm for interactive frame rates during visualisation of complex virtual environments. *Proceedings of SIGGRAPH'93*, pages 247–254, 1993.

[7] J.-L. Giavitto, C. Godin, O. Michel, and P. Prusinkiewicz. *Modelling and Simulation of biological processes in the context of genomics*, chapter "Computational Models for Integrative and Developmental Biology". Hermes, July 2002.

[8] Paul Horn. Autonomic computing: IBM's perspective on the state of information technology. Manifesto, IBM Research, October 2001.

[9] J. von Neumann. *The art of Simulation.* English University Press, 1963.

[10] J.J. Shi . Activity-based construction (ABC)Modeling and Simulation Method. *J. Constr. Engrg. and Mgmt.*, 125(5):354–360, September 1999.

[11] L. von Bertalanffy. *General System Theory. Foundations, development, applications.* 1968.

[12] W. E. Leland and T. J. Ott. Load balancing heuristics and process behavior. *ACM Performance Evaluation Review*, 14(1):54–69, May 1986.

[13] D.R.C. Hill P. Coquillard. *Modélisation et Simulation des Ecosystèmes.* Masson, 1997.

[14] M. Pidd. Object-orientation and three phase simulation. In *Proc. of Winter Simulation Conference*, pages 689 – 693, 1992.

[15] J. Francisco Ramos and Miguel Chover. Level of detail modelling in a computer game engine. In Matthias Rauterberg, editor, *ICEC*, volume 3166 of *Lecture Notes in Computer Science*, pages 451–454. Springer, 2004.

[16] Roy Sterritt, Manish Parashar, Huaglory Tianfield, and Rainer Unland. A concise introduction to autonomic computing. *Advanced Engineering Informatics*, 19(3):181–187, 2005.

[17] Mark Weiser. Ubiquitous computing (abstract). In *ACM Conference on Computer Science*, page 418, 1994.