# Definitions of Activity Measures

Alexandre Muzy

May 14, 2013

This is only a brief introduction to Activity theory for modeling and simulation. It aims at providing canonical (because very simple) definitions, in the context of dynamical and discrete event systems.

Usually, in simulation, *(qualitative) activities* of systems consist of phases, which "start from an event and end with another" [2]. Information about the dynamics of the system is embedded into phases $p \in P$ corresponding to strings ("burning", "waiting", etc.) Mathematically, an event $ev_i$ is denoted by a couple $(t_i, v_i)$, where $t_i \in \mathbb{R}^{+,*}$ is the timestamp of the event, and $v_i \in V$ is the value of the event. Therefore, usual qualitative activities have values in $P$. Each activity consists of a triple $a = (p, ev_i, ev_{i+1})$, with $v_i = p$ for $t_i \leq t < t_{i+1}$, with $i \in \mathbb{N}$. An example of qualitative activity sequence is depicted in Figure 1. The set of qualitative activities consists of: $A_q = \{(p, ev_i, ev_{i+1})\}$.

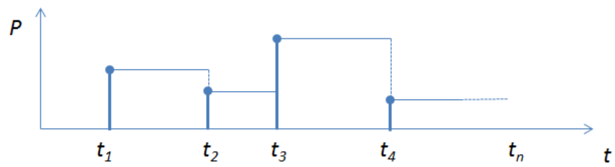


Figure 1: An example of usual qualitative activity definition.

All the definitions presented hereafter aim at providing different (*quantitative*) definitions of activity. The latter is a metrics of: continuous changes, number of transitions, number of state changes, etc.

## 1 Activity of continous segments

Considering a continuous function $\Phi(t)$ (*cf.* in Figure 2) and related extrema $m_n$, model *continuous activity* $A_c(T)$[4] of this trajectory, over a period of time $T$, consists of kind of "distance":

$$A_c(T) = \int_0^T \left| \frac{\partial \Phi(t)}{\partial t} \right| dt \simeq \Sigma_{i=1}^n |m_i - m_{i+1}|$$

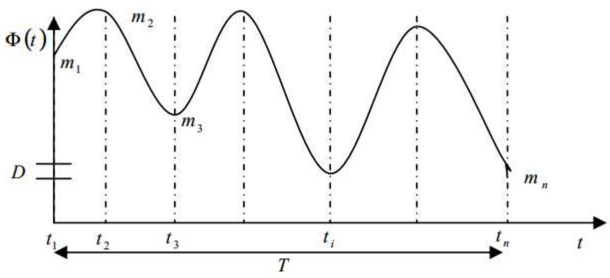


Figure 2: Continuous trajectory with extrema.

*Average continuous activity* consists then of $\overline{A_c(T)} = \frac{A_c(T)}{T}$ .

Now considering a significant change of value of size $D = \left|\Phi^{n+1} - \Phi^n\right|$, called a *quantum*, the *discretization activity* $A_d(T)$[1], corresponding to the *minimum number of transitions* necessary for discretizing/approching the trajectory of $\Phi(t)$ (*cf.* Figure 3) is:

$$A_d(T) = \frac{A_c(T)}{D}$$

*Average discretization activity* consists then of $\overline{A_d(T)} = \frac{A_s(T)}{T}$ .

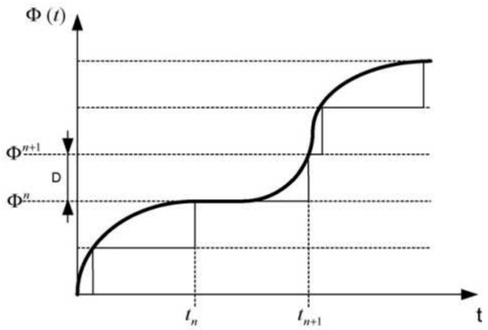


Figure 3: Continuous trajectory with extrema.

An *event set* is defined as $\xi = \{ev_i = (t_i, v_i) \mid i = 1, 2, 3, ...\}$, where a discrete event $ev_i$ is a couple of timestamp $t_i \in \mathbb{R}^{+,*}$ and value $v_i \in V$.

Considering a time interval $\langle t_0, t_n \rangle$ , an *event segment* is defined as $\omega$: $\langle \mathrm{t_0}, \mathrm{t_n} \rangle \rightarrow V \cup \{\varnothing\}$, with "$\varnothing$" corresponding to nonevents. Segment $\omega$ is an event segment if there exists a finite set of times points $t_1, t_2, t_3, ..., t_{n-1} \in \langle t_0, t_n \rangle$ such that $\omega(t_i) = v_i \in V$ for $i = 1, ..., n-1$ and $\omega(t) = \varnothing$ for all other $t \in \langle t_0, t_n \rangle$.

An *activity segment* (cf. Figure 4) of a continuous function $\Phi(t)$ is defined as an event segment such that $\omega(t_i) = \frac{m_i}{t_i - t_{i-1}}$ for $i = 1, ..., n-1$ and $t_0 = 0$.

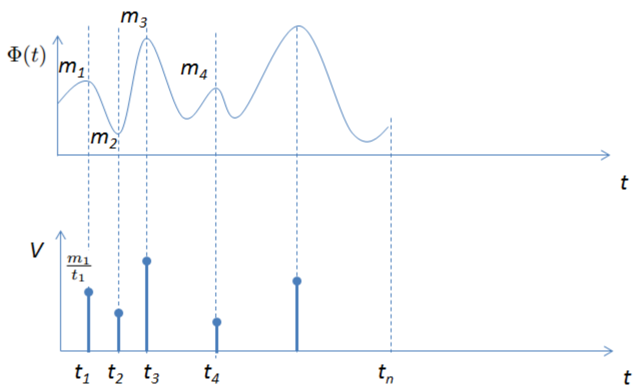


Figure 4: Continuous trajectory with extrema.

# 2 Envent-based activity

We consider here the activity as a measure of the number of events in an event set $\xi = \{ev_i = (t_i, v_i) \mid i = 1, 2, 3, ...\}$, for $0 \leq t_i < T$.

## 2.1 Activity in an discrete event set

*Event-based activity* $A_\xi(T)$ [5] consists of :

$$A_\xi(T) = |\{ev_i = (t_i, v_i) \in \xi \mid 0 \leq t_i < T\}|$$

*Average event-based activity* consists then of $\overline{A_\xi(T)} = \frac{A_\xi(T)}{T}$ .

For example, assuming the event trajectory depicted in Figure 5, the average event-based activity of the system corresponds to the following values for different time periods: $\overline{A_\xi(10)} = 0.3$, $\overline{A_\xi(20)} = 0.15$, $\overline{A_\xi(30)} \simeq 0.133$, $\overline{A_\xi(40)} = 0.175$.

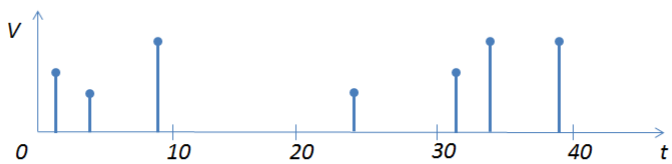


Figure 5: An example of event trajectory.

## 2.2 Event-based activity in a Cartesian space

*Activation* and *non-activation* can be used to partition the set of positions $p \in \mathcal{P}$ in a Cartesian space. *Activation* is simply defined as an *event-based activity*

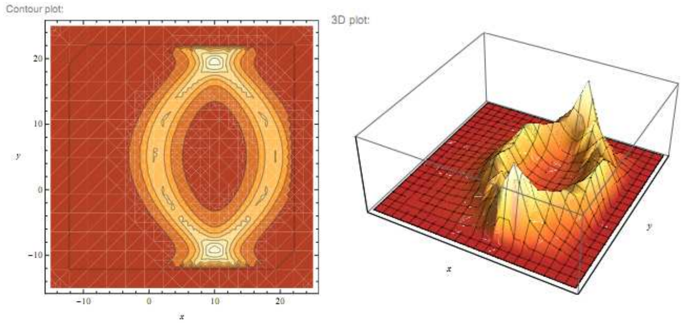


Figure 6: 2D and 3D visualization of event-based activity in a 2D space. $x$ and $y$ represent Cartesian coordinates. The event-based activity of each coordinate is represented in the third dimension.

$A_\xi(T) > 0$ while *non-activation* is defined as an *event-based activity* $A_\xi(T) = 0$. Related partitions are called *activity* and *inactivity regions* [5]:

- Activity region in space:

$$\mathcal{AR}^{\mathcal{P}}(T) = \{p \in \mathcal{P} \mid A_{\xi,p}(T) > 0\}$$

  where $A_{\xi,p}(T)$ corresponds to the event-based activity at position $p \in \mathcal{P}$ .

- Inactivity region in space:

$$\overline{\mathcal{AR}^{\mathcal{P}}(T)} = \{p \in \mathcal{P} \mid A_{\xi,p}(T)= 0\}$$

A function of reachable states can be considered in time and space as $r : \mathcal{P} \times \mathcal{T} \rightarrow Q$, where $Q$ is the set of states of the system. The set of all reachable states in the state set $Q$, through time and space, can be defined *universe* $\mathcal{U} = \{r\,(p, t) \subseteq Q \mid p \in \mathcal{P}, t \in \mathcal{T}\}$. Considering that all reachable states in time and space can be active or inactive, an activity-based partitioning of space $\mathcal{P}$ can be achieved: $\forall t \in \mathcal{T}$, $\mathcal{P} = \mathcal{AR}^{\mathcal{P}}(T) \cup \overline{\mathcal{AR}^{\mathcal{P}}(T)}$.

Figure 6 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.)

In spatialized models (cellular automata, L-systems,...), components are localized at Cartesian coordinates in space $\mathcal{P}$. Each component $c$ is assigned to a position $c_p \in \mathcal{P}$.

Applying the definition of activity regions in space 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 event-based activity occurs. Consequently, active components, over time period $T$, correspond to the components localized at positions $p$, with $A_{\xi,p}(T) > 0$, while inactive components have a null event-based activity $A_{\xi,p}(T) = 0$.

# 3 Activity in Discrete Event System Specification (DEVS)

DEVS allows separating model and simulator (called the *abstract simulator*). The latter is in charge of *activating* the transitions of the model. This allows counting the number of transition executions (activations). This measure is the *simulation activity[6]*. Each transition can be also *weighted [3]*.

## 3.1 Model

The dynamics of a component can be further described using a Discrete Event System Specification (DEVS). The latter is a tuple, denoted by $DEVS =< X, Y, S, \delta, \lambda, \tau >$, where $X$ is the *set of input values*, $Y$ is the *set of output values*, $S$ is the *set of partial sequential states*, $\delta : Q \times (X \cup \{\emptyset\}) \rightarrow S$ is the *transition function*, where $Q = \{(s,e)\,|s \in S, 0 \leq e \leq \tau(s)\}$ is the *total state set*, $e$ is the *time elapsed* since the last transition, $\emptyset$ is the *null input value*, $\lambda : S \rightarrow Y$ is the *output function*, $\tau : S \rightarrow \mathbb{R}^{+}_{0,\infty}$ is the *time advance function.*

If no event occurs in the system, the latter remains in partial sequential state $s$ for time $\tau(s)$. When $e = \tau(s)$, the system produces an output $\lambda(s)$, then it changes to state $(\delta(s,e,x), e) = \big(\delta(s, \tau(s), \emptyset), 0\big)$, which is defined as an *internal transition* $\delta_{int}(s)$. If an external event, $x \in X$, arrives when the system is in state $(s,e)$, it will change to state $(\delta(s, \tau(s), x), 0)$, which is defined as an *external transition* $\delta_{ext}(s,e,x)$.

## 3.2 Activity-based abstract simulator

Modifications of usual abstract simulator for atomic models *[6]* is presented here:

- *External activity* $A_{ext}$, related to the counting $n_{ext}$ of *external transitions* $\delta_{ext}(s,x) = (\delta(s, \tau(s), x), 0)$, over a time period $[t, t']$:

$$\begin{cases} s' = \delta_{ext}(s,e,x) \Rightarrow n'_{ext} = n_{ext} + 1 \\ A_{ext}(t'-t) = \frac{n_{ext}}{t'-t} \end{cases}$$

- *Internal activity* $A_{int}$, related to the counting $n_{int}$ of *internal transitions* $\delta_{int}(s) = \big(\delta(s, \tau(s), \emptyset), 0\big)$, over a time period $[t, t']$:

$$\begin{cases} s' = \delta_{int}(s,e) \Rightarrow n'_{int} = n_{int} + 1 \\ A_{int}(t'-t) = \frac{n_{int}}{t'-t} \end{cases}$$

**Algorithm 1** Modified abstract simulator for weighted activity

**variables**
  $parent$ — parent coordinator
  $tl$ — time of last event
  $tn$ — time of next event
  $DEVS$ — associated model with total state $(s, e)$
  $y$ — output message bag
  $n_{int}$ — number of internal transitions
  $n_{ext}$ — number of external transitions

**when** receive i-message $(i, t)$ at time $t$
  $tl = t–e$
  $tn = tl + ta(s)$
**when** receive *-message $(*, t)$at time $t$
  **if** $(t = tn)$ **then**
    $y = \lambda(s)$
    send y-message $(y, t)$ to parent coordinator
    $s = \delta_{int}(s)$
    $n'_{int} = n_{int} + 1$
**when** receive x-message (x, t)
  **if** $(x \neq \oslash$ and $tl \leq t \leq tn)$ **then**
    $s = \delta_{ext}(s, x, e)$
    $n'_{ext} = n_{ext} + 1$
$tl = t$
$tn = tl + \tau(s)$

- *Simulation (total) activity* $A_s(t' - t)$ is equal to:

$$A_s(t' - t) = A_{ext}(t' - t) + A_{int}(t' - t)$$

- *Average simulation activity* $\overline{A_s(t' - t)}$ is equal to:

$$\overline{A_s(t' - t)} = \frac{A_{ext}(t' - t) + A_{int}(t' - t)}{t' - t}$$

Here simulation activity is simply a counter of the number of events. However, when events have different impacts, weighted activity is introduced.

### 3.3 Abstract simulator for weighted activity

*Weighted simulation activity* $A_w(T)$ has been defined in [3]. It is related to a modified abstract simulator:

- *External weighted activity* $A_{ext,w}$, related to the counting $n_{ext,w}$ of *external transitions* $\delta_{ext}(s, x) = (\delta(s, \tau(s), x), 0)$, over a time period $[t, t']$:

**Algorithm 2** Modified abstract simulator for weighted activity

**variables**
  $parent$ — parent coordinator
  $tl$ — time of last event
  $tn$ — time of next event
  $DEVS$ — associated model with total state $(s, e)$
  $y$ — output message bag
  $n_{int}$ — number of internal transitions
  $n_{ext}$ — number of external transitions

**when** receive i-message $(i, t)$ at time $t$
  $tl = t–e$
  $tn = tl + ta(s)$
**when** receive *-message $(*, t)$at time $t$
  **if** $(t = tn)$ **then**
    $y = \lambda(s)$
    send y-message $(y, t)$ to parent coordinator
    $s = \delta_{int}(s)$
    $n'_{int,w} = n_{int,w} + wt_{int}(s)$
**when** receive x-message (x, t)
  **if** $(x = \oslash$ and $tl \leq t \leq tn)$ **then**
    $s = \delta_{ext}(s, x, e)$
    $n'_{ext,w} = n_{ext,w} + wt_{ext}(s, e, x)$
$tl = t$
$tn = tl + \tau(s)$

$$\begin{cases} wt_{ext} : X \times Q \to \mathbb{N}^0 \\ s' = \delta_{ext}(s, e, x) \Rightarrow n'_{ext,w} = n_{ext,w} + wt_{ext}(s, e, x) \\ A_{ext,w}(t' - t) = \frac{n_{ext,w}}{t'-t} \end{cases}$$

- *Internal weighted activity* $A_{int,w}$, related to the counting $n_{int,w}$ of *internal transitions* $\delta_{int}(s) = \left(\delta(s, \tau(s), \emptyset), 0\right)$, over a time period $[t, t']$:

$$\begin{cases} wt_{int} : S \to \mathbb{N}^0 \\ s' = \delta_{int}(s, e) \Rightarrow n'_{int,w} = n_{int,w} + wt_{int}(s) \\ A_{int,w}(t' - t) = \frac{n_{int,w}}{t'-t} \end{cases}$$

- *Simulation (total) weighted simulation activity* $A_w(t' - t)$ is equal to:

$$A_w(t' - t) = A_{ext,w}(t' - t) + A_{int,w}(t' - t)$$

- *Average weighted simulation activity* $\overline{A_w(t' - t)}$ is equal to:

$$\overline{A_w(t' - t)} = \frac{A_{ext,w}(t' - t) + A_{int,w}(t' - t)}{t' - t}$$

# 4 Open Research

Quantification of component activity opens new research directions, *e.g.*, in:

- Machine Learning, where activity corresponds to the *usage* of components in the search space and can be correlated to the payoff of component compositions,
- Networks, where activity provides an indication of the frequency of node accesses as well as *information paths,*
- Systems-theory, where activity could be used for the specification of dynamic systems (from input-output behaviors to internal structures),
- ...

In relation to these theoretical directions, many application domains can be considered:

- In Neurosciences, through the mapping between the activity of components/networks and neurons/brain regions,
- In Ecology, through the analogy between activity and the energy used by organisms to survive and evolve,
- In Economics, through the comparison of decision paths, characterized through their activity.
- In propagation processes, activatability and activity can be used for optimization through activity tracking at run-time, or for activatability preprocessing (*e.g.*, in fire spread, where the vegetation is expected to burn, etc.)
- ...

# 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] 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.

[3] X. Hu and B.P. Zeigler. Linking information and energy - activity-based energy-aware information processing. *SIMULATION*, page to appear, 2013.

[4] Alexandre Muzy, Rajanikanth Jammalamadaka, Bernard P. Zeigler, and James J. Nutaro. The activity-tracking paradigm in discrete-event modeling and simulation: The case of spatially continuous distributed systems. *Simulation*, 87(5):449–464, 2011.

[5] Alexandre Muzy, Luc Touraille, Hans Vangheluwe, Olivier Michel, Mamadou Kaba Traoré, and David R. C. Hill. Activity regions for the specication of discrete event systems. In *Spring Simulation Multi-Conference Symposium On Theory of Modeling and Simulation (DEVS)*, pages 176–182, 2010.

[6] Alexandre Muzy and Bernard P. Zeigler. Activity-based Credit Assignment (ACA) in Hierarchical Simulation. In *IEEE/ACM/SCS: SpringSim Multi-conference, Symposium On Theory of Modeling and Simulation*, page accepted, Orlando, USA, March 2012.