TIME SATISFACTION IN COORDINATED MULTI-AGENT SYSTEMS WITH RELATIVE TIME RATES

Said Layadi
MISC Laboratory
Constantine 2 University
25000, Constantine, Algeria
E-mail: layadi@misc-umc.org

Jean-Michel Ilié
Universities of Paris UPMC
and Paris Descartes
4 place Jussieu, 75005, Paris, France
E-mail: jean-michel.ilie@upmc.fr

Djamel-Eddine Saidouni
MISC Laboratory
Constantine 2 University
25000, Constantine, Algeria
E-mail: saidouni@misc-umc.org

KEYWORDS

Relative time rates, Region graph, Multi-agent systems, Ambient systems, Timed transport fleets.

ABSTRACT

In this paper, we aim at checking time satisfaction for multi agent systems, the agents of which must coordinate. As in real systems, agents can behave in true concurrency and may have relative time rates. In this context we show that the problem can be reduced to the searching over a specific time region graph.

INTRODUCTION

The coordination problem is a recurrent problem in muti-agent systems. Mainly, this consists in maintaining the synchronization of the agent plans with respect to some objective (Guivarch et al. 2012; Pi et al. 2013).

In this paper, we consider real time application wherein the agent plans refers to timed actions whose durations are known. Agents are assumed to be able to communicate via reliable protocols, in order to achieve some plan called coordinated plan. We aim at providing a technique for the agents to check the satisfaction of temporal properties over such a plan.

As far as time is concerned, the first difficulty to cope with is the heterogeneity of the agent performances. Hybrid Timed automata are often used to represent timed behavioral problems with different speeds, since the represented clocks can be associated with distinguished speeds. In our approach, we propose to model plan in a more attractive way using the standard algebraic language based on LOTOS, seeing plans as the execution of concurrent processes (Chaouche et al. 2014). In fact, LOTOS specifications supplies modular concepts useful here to part some plan over several agents.

Moreover, LOTOS-basic specification are known to be translated in a specific class of timed automata called Duration Actions Timed Automata (DATA for short), such that DATA is demonstrated to be the most compacted way to represent the true concurrency of processes (Belala and Saidouni 2005). DATA compactly represents the possibly infinite behaviors of the coordinated plans. In this paper, we show how to build a (finite) region graph structure from a DATA specification, in order to check the timing constraints defined over the coordinated plans, taking into account relative time rates that distinguishes the speeds of the agents in coordination.

The schedule of the paper is the following: In Section II, the specification of a coordinated multi-agent system is presented, based on LOTOS concepts, followed by the semantical DATA representation model. In Section III, we show how to extend the operational rules to build a DATA, to take into account both coordination and heterogeneity of the agents. In Section IV, a specific algorithm is defined to build a region graph, while taking the relative time rates into account. The next section is our conclusions, including some of our immediate perspectives.

COORDINATED MAS SPECIFICATION

In this paper, a multi agent system is a tuple $MAS=(Ag, Plan, Act, \tau, \gamma)$ where Ag is set of agents; $Plan$ is a set of agent plans, called *coordinated plans* since some of them are realized by several agents, Act is the set of actions mentioned in these plans. $\tau=(\tau_p)_{p\in Ag}$ is a rate mapping associating a relative time rate with each agent characterizing the speed of the agent to execute their actions, γ is a duration mapping assigning a *global duration value* to each action of Act, evaluated in a number of execution cycles (called *time units*).

In the following, we describe the specification of coordinated plans as an extension of the language Basic LOTOS, precising which parts of the plans are dedicated to each agent.

The Relative Time Model

In the MAS systems, agents are assumed to have a notion of clocks to achieve their actions. Since agents can have different speeds, we assume that the speeds are relative according to a global time-scale (denoted absolute time), thus the duration performance of some action can be more or less important, depending on the agent considered to execute the action.

Proceedings 29th European Conference on Modelling and Simulation ©ECMS Valeri M. Mladenov, Petia Georgieva, Grisha Spasov, Galidiya Petrova (Editors)
ISBN: 978-0-9932440-0-1 / ISBN: 978-0-9932440-1-8 (CD)

In Figure 1, the action a of global duration 2, is achieved two times faster by the agent q than by agent p.

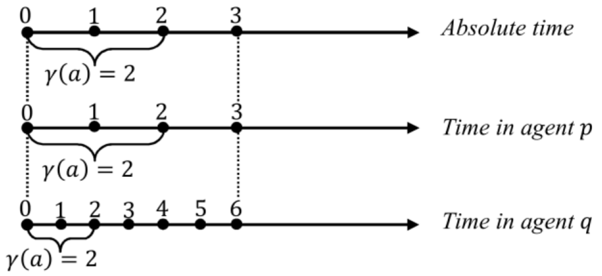


Figure 1: Relative Time over Two Agents

For instance in Figure 1, the action a of duration 2, requires 2 times more to be achieved by the agent p. Accordingly to any global time t, the time rate $\tau_q = 2t$ and $\tau_p = t$, in such a way that at any time τ_q / τ_p belongs to $\mathbb{Z}$.

Coordinated Plan Specification

The plan of an agent p is specified by an agent expression E_p describing the actions to be executed for achieving the plan. The execution is assumed to be controlled by the agent p, however E_p can be composed of sub-expressions whose execution can be performed by other agents. An agent expression inherits from the syntax of (Basic) LOTOS as follows:

$$P ::= E_p$$
$$E_p ::= exit \quad | \quad stop$$
$$| \quad a; E_p \qquad (a \in Act)$$
$$| \quad E_q \odot E_r \qquad q, r \in Ag$$
$$\odot \in \{|||, ||, |[L]|, [], \gg, [>\} \ \ LOTOS\ operator$$

The elementary expression *stop* specifies a plan behavior without possible evolution and *exit* represents the successful termination of some plan. In the syntax, any operator ⊙ is standard in LOTOS: $E_q[]E_r$ specifes a non-deterministic choice, $E_q{>>}E_r$ a sequential composition and $E_q[>E_r$ the interruption of the left hand side part by the right one.

The LOTOS parallel composition, denoted $E_q|[L]|E_r$, can model both synchronous composition when (L=Act), denoted $E_q||E_r$, and asynchronous composition, $E_q|||E_r$ when (L=∅). In fact, the LOTOS language exhibits a rich expressivity such that the sequential executions of plans appear to be only a particular case.

Example of Transport

The frame example of this paper simply concerns two trucks A and B such that A initially placed in the location l_1 must pick up the load in the location l_2 and delivers it to the location l_4. As the load is initially placed in l_3, B which is initially placed in the location l_2, proposes to get the load from l_3 in such a way that A can meet B in l_2 and take the load. The problem for A and B is to meet them in minimum time, in case they start at the same time.

In order to coordinate, each truck is equipped with a software agent able to discuss and synchronize with the other agents in the system. Both agents A and B refer to the following coordinated plan :

$P ::= E_A\ |[meet]|\ E_B$
$E_A ::= move_A(l_2);\ meet;\ move_A(l_4);\ exit$
$E_B ::= move_B(l_3);\ get\_load_B;\ move_B(l_2);\ meet;\ exit$

Moreover, the respective learnt experiences of A and B about transportations yield a duration for each of these actions. For the simplicity of the example, all duration of actions are assumed equal to 1 time unit.

FROM BASIC LOTOS WITH ACTION DURATIONS TO DATA WITH RELATIVE TIME RATES

The DATA model of a LOTOS specification is semantical structure designed as a transition system. It is used to express the causality and temporal conditions that must hold throughout the executions of the actions in plans. A DATA with Relative time rates (DATA-R for short), inherits from the DATA structure. The common preliminaries are the followings:
Let $\mathcal{M}$, ranged over $x,y,\ldots$, be a set of clocks with non negative real values (in $\mathbb{R}_{\geq 0}$) and consider $H \subset \mathcal{M}$. The set Φ_H of temporal constraints G over H is defined by the syntax G::=true$|\neg G|G \wedge G|x \sim t$, where x is a clock in H, $\sim \in \{=,<,>,\leq,\geq\}$ and $t \in \mathbb{R}_{\geq 0}$. Further, F_x specifies a temporal constraint of the form $x \sim t$. A *valuation* v for H is a mapping which associates a value in $\mathbb{R}_{\geq 0}$ with each clock $x \in H$. The set of all valuations for H is denoted $\Xi(H)$. The satisfaction relation $\vDash$ for temporal constraints over $\Xi(H)$ is $v \vDash x \sim t \Leftrightarrow v(x) \sim t$ such that $v \in \Xi(H)$. For $I \subseteq H$, $[I \rightarrow 0]v$ indicates the valuation for H which assigns the value 0 to each $x \in I$, and agrees with the valuation v for the other clocks of H.

Definition (DATA). W.r.t. a countable set of clocks $\mathcal{M}$, a *Durational Action Timed Automaton* $\mathcal{A}$ is a tuple (S, L_s, s_0, H, T) where:

- S is a finite set of states,
- $H \subset \mathcal{M}$ is a finite set of clocks,
- $L_S : S \rightarrow 2^{\Phi H}$ is a mapping to associate a set of temporal constrainsts with each state s. Each one expresses by its clock, the fact that some action is potentially in execution in s, under a duration condition.
- $s_0 \in S$ is the initial state and $L_s(s_0) = \emptyset$.
- $T \subseteq S \times 2^{\Phi H} \times Act \times H \times S$ is the set of transitions. A transition (s,G,a,x,s') represents the change from the state s to s', launching the performance of the action a, represented by the clock x once it is reset and submitted to the temporal constraint G to be satisfied by x in s'. (s,G,a,x,s') is also written $s \xrightarrow{G,\ a,\ x} s'$.

Definition (DATA with relative time rates). A *DATA with relative time rates* (DATA-R) over the set of agents Ag is a structure $(\mathcal{A}, \pi)$ where $\mathcal{A}=(S, L_s, s_0, H, T)$ is a

DATA and π is a distribution of the clocks of *H* over the agents, defined as a subjective mapping from *H* to *Ag*.

Observe that several clocks can be assigned to each agent, said to belong to that agent. The valuations of the clocks of an agent behave synchronously. In a *DATA* with relative time rates, the clocks that belong to different agents evolve synchronously but under the relative time rates assigned to these agents. Each such rate assigned to the agent *p* represents the speed of *p* and depends on some global time reference, from which the local time of *p* is derived. Formally, a mapping τ is defined from every global time value *t* of $\mathbb{R}_{\geq 0}$, assigning the tuple $(\tau_p(t))_{p\in Ag}$ to $\tau(t)$, s.t. $\tau_p(t)\in\mathbb{R}_{\geq 0}$ is the local time for *p*, taken at time *t*. Initially, $\tau_p(0)=0$.

The building of a DATA from a LOTOS expression with durations requires to define semantical operational rules, figuring out the derivation of every possible subexpression according to LOTOS operators.

The DATA-R structure has in fact the same structure as a DATA one, representing state by state, the causality and temporal constraints to respect. Prior to formally define the DATA-R rules, we now focus on our frame example to bring out its associate DATA-R structure. The reader can refer e.g. to (Courtiat and Saïdouni 1995; Saïdouni 1996) for details.

Building of a DATA-R Structure (Unformal)

Each state of a DATA refers to a LOTOS (sub-) expresion to evaluate, the analysis of which yields the next actions offered to the execute, hence the definition of the next state. Motivated by a dynamic management of clocks the set $\mathcal{M}$ is used as a collection of freed clocks, so that every offered action is assigned to a clock of $\mathcal{M}$ to specify its execution corresponding to the time the execution of the action is surely achieved. Any freed clock can be reused to the execution of another offered action, once being reset to 0.

Figure 2 highlights the DATA(-R) structure of the LOTOS frame example. From the subexpressions E_A and E_B, it appears that both actions $move_A(l_2)$ and $move_B(l_3)$ can be concurrently offered. So, each arc starting from the initial state is labeled by the offered action, e.g. $move_A(l_2)$, the clock of $\mathcal{M}$ that is allocated to the action. Initially, there is not any action in execution, therefore the set of temporal constraints attached to the initial state is empty, and the guard attached to the transitions are true (hence not represented).

From state to state, several offered actions are lauched, considering their clocks are distinct and can behave synchronously. When several actions are offered concurrently from some state, their shuffle is represented in whatever order, by successive transitions and states. This can be seen at the state 2, where the concurrent actions $move_A(l_2)$ and $move_B(l_3)$ are successively lauched, in whatever order.

The concurrency of executions is abstractly represented at the state level, since the clocks associated with the launched actions are assumed to progress synchronously, each one submitted to a duration condition corresponding to the duration of the associated action. In the state 2, the set $\{x\geq 1,\ y\geq 1\}$ expresses that the clocks *x* and *y*, respectively attached to $move_A(l_2)$ and $move_B(l_3)$, can progess synchronously but none of them can overpass a duration of 1 time unit.

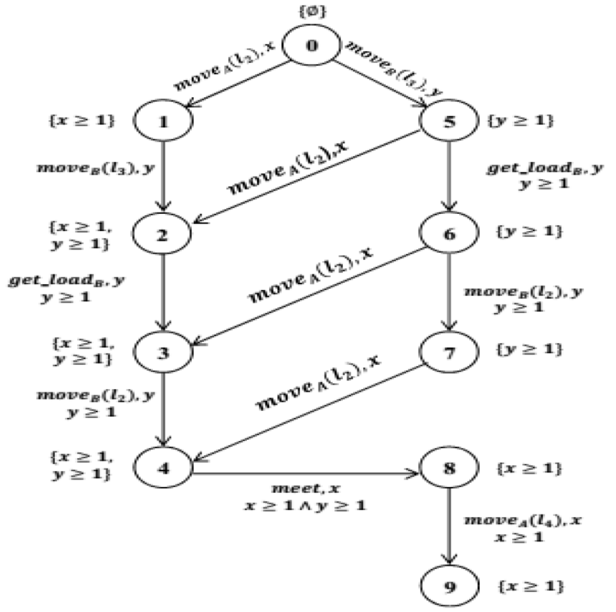


Figure 2: Example of DATA-R Structure

In the DATA model, the termination of the execution of an action is not explicitly acted in the states, and even can occur several (successive) states after the action is launched. Such a termination is only specified through the causality of actions, when launching a new action. For instance from the state 2, the action get_load_B is launched, but only under the guard $y\geq 1$, which surely corresponds to the termination of the action $move_B(l_3)$, just previously started by the agent *B*.

As another consequence, a clock mentioned in a guard of a transition becomes freed as the transition is considered, hence can be immidiatly reused. That is why the launching of get_load_B is associated with the clock *y* although it is previously associated with the previous action $move_B(l_3)$, of the agent *B*.

Lastly, when a parallel composition occurs, like from the state 4 where the action *meet* can be offered, only one transition is designed even if both agents *A* and *B* are concerned.

Semantical Rules to Build a DATA-R

The semantical rules producing the DATA-R model are detailed in Table 1, wherein each cell concerns one of

LOTOS operators. As for a DATA model, there may be several subcases. Unlike DATA, the LOTOS expression to consider are labeled symbolically by an agent name, moreover, the function π is used to specify the agent time rate which is used when making the clock progresses.

All the derivation rules are built in the same way. The state to be derived is denoted ${}_{F}[E]$ and is a pair composed of a (behavior) expression E written in LOTOS, and a set F of duration conditions ($F \in 2^{\Phi H}$) which constraints the derivation of E. When performing a derivation, the offered action is generically named a, moreover, the function *get* defined from $\mathcal{M}$, is used to obtain a free clock for a.

Consider the frame example again to have an insight of the possible derivations, in respect to the DATA-R represented in Figure 2. From the initial state 0, the action $move_A(l_2)$ in the left expression s.t. $E_A = move_A(l_2); E_A'$ is directly offered to execution, since there is no prior duration conditions to satisfy ($\varnothing$). A clock x, given by the function *get* and set to 0, is attached to the action. In the state 1, the left expression E_A' is now constrained by the termination of $move_A(l_2)$, hence the condition duration ($x \geq 1$), so:

$$\underbrace{{}_{\emptyset}[E_A]|[meet]|\,{}_{\emptyset}[E_B]}_{state_0} \xrightarrow{\emptyset,\ move_A(l_2),\ x} \underbrace{{}_{x\geq 1}[meet, move_A(l_4), exit]|[meet]|\,{}_{\emptyset}[E_B]}_{state_1}$$

From the state 1, the offering is similar but concerns $move_B(l_3)$ in the right expression and the clock y:

$$state_1 \xrightarrow{\emptyset,\ move_B(l_3),\ y} \underbrace{{}_{x\geq 1}[meet, move_A(l_4), exit]|[meet]|\ {}_{y\geq 1}[get\_load_B; move_B(l_2); meet; exit]}_{state_2}$$

From the state 4, the action meet can perform only if both actions $move_A(l_2)$ and $move_B(l_2)$ finish their executions, in other words, the duration conditions to satisfy are now $\{x \geq 1, y \geq 1\}$:

$$state_4 \xrightarrow{\{x\geq 1, y\geq 1\},\ meet,\ x} \underbrace{{}_{x\geq 1}[move_A(l_4), exit]|[meet]|\ {}_{x\geq 1}[exit]}_{state_8}$$

It is worth noticing that for any synchronization action, the assigned clock must evolve according to the slowest agent speed. In the example, the truck A is the slowest one (w.r.t. the rate τ_A and τ_B), therefore the clock x previously attached to the executions of the actions of A is again attached to the action *meet*. This mechanism ensures the synchronization at the end of each synchronized action.

Lastly, observe that in the derivation rules of the DATA-R, the following helping functions are used. These are standard in LOTOS (Belala and Saidouni 2005; Maarouk et al. 2011):

- $\psi({}_{F}[E])$ extracts from the expression E the set of clocks that are used in E.
- $\psi_{DC}({}_{F}[E])$ yields the set F of duration conditions.
- $\psi_H(F)$ extracts the set of clocks from a set of duration conditions F.
- $({}_{F}[E] \backslash K)$ removes from F, the set of durations that use a clock of the set K.
- $({}_{F}[E])\sigma$ substitutes a subset of clocks of F by as many clocks, according to the specification in σ.

Definition. The transition relation of the DATA-R $\rightarrow\ \in \mathcal{C} \times 2^{\Phi H} \times Act \times H \times \mathcal{C}$ is defined as the smallest relation satisfying the rules in Table 1.

REGION AUTOMATON

In reference to the DATA-R, a reachability graph of (timed) configurations can be computed by attaching to each state of the DATA-R, a valuation of the defined clocks. Inspired from (Alur and Dill, 1994), such configurations car be aggregated according to an equivalence relation preserving the runs. This relation is deduced from an equivalence relation defined on clocks. An equivalence class of clock valuations is named a *clock region*.

We further assume that the clock constraints of the DATA-R only involve integer constants. As precised in (Alur and Dill 1994), it is not a theoretical limitation but offers easier presentation.

Within some region, the progression of the clock valuations w.r.t. different agents refers to the notion of slopes.

Equivalence Classes of Clock Valuations

Let $A=((S, L_s, s_0, H, T), \pi)$ be a DATA-R over a set Ag.

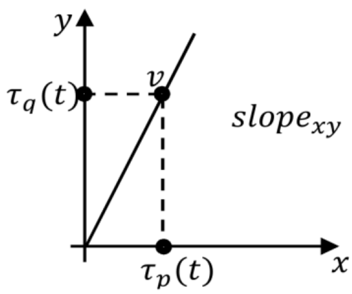
Definition ($slope_{xy}$). Let x (resp. y) be a clock that belongs to the agent p (resp. q) and evolves according to the rate function τ_p (resp. τ_q). We define $slope_{xy}$ as the ratio of local-time rate functions τ_p and τ_q, denoted $slope_{xy} = \tau_q / \tau_p$ (see Figure 3).

As the number of clock constraints used in the DATA-R is necessarily limited, we can determine for each used clock x, an integer value $c_x \in \mathbb{N}$ which corresponds to the greatest value x is compared in some clock constraint (guard) in the DATA-R. Moreover, whatever the pair of clocks x and y, the parameter $slope_{xy}$ is assumed an integer constant.

We now define the equivalence relation that is used to build the regions. The notation $\lfloor v(x) \rfloor$ represents the integer value of the valuation $v(x)$, and *fract* its decimal part.

Table 1: Derivation Rules Producing the DATA-R Model

$\dfrac{}{{}_F[exit] \xrightarrow{F,\delta,x}{}_{\{x\geq 0\}}[stop]}$ $\quad x = get(\mathcal{M})$	$\dfrac{}{{}_F[a;E_p] \xrightarrow{F,a,x}{}_{\{x\geq \gamma(a)\}}[E_p]}$ $\quad x = get(\mathcal{M})\ and\ \pi(x) = p$
$\dfrac{\mathcal{E}_p \xrightarrow{F,a,x} \mathcal{E}'_p}{\mathcal{F}_q\ [\,]\ \mathcal{E}_p \xrightarrow{F,a,x} \mathcal{E}'_p \qquad \mathcal{E}_p\ [\,]\ \mathcal{F}_q \xrightarrow{F,a,x} \mathcal{E}'_p}$	$(a)\quad \dfrac{\mathcal{E}_p \xrightarrow{F,a,x} \mathcal{E}'_p \qquad a \neq \delta}{\mathcal{E}_p \gg F_q \xrightarrow{F,a,x} \mathcal{E}'_p \gg F_q}$ $(b)\quad \dfrac{\mathcal{E}_p \xrightarrow{F,\delta,x} \mathcal{E}'_p}{\mathcal{E}_p \gg F_q \xrightarrow{F,\delta,x}{}_{\{x\geq 0\}}[F_q]}$
$(a)\quad \dfrac{\mathcal{E}_p \xrightarrow{F,a,x} \mathcal{E}'_p \qquad a \notin L \cup \{\delta\}}{\mathcal{E}_p \mid[L]\mid \mathcal{F}_q \xrightarrow{F,a,y} \mathcal{E}'_p[y/x] \mid[L]\mid \mathcal{F}_q \backslash F \qquad \mathcal{F}_q \mid[L]\mid \mathcal{E}_p \xrightarrow{F,a,y} \mathcal{F}_q \backslash F \mid[L]\mid \mathcal{E}'_p[y/x]}$ $y = get\left(\mathcal{M} - \left(\left(\psi(\mathcal{E}_p) \cup \psi(\mathcal{F}_q)\right) - \psi_H(F)\right)\right)\ and\ \pi(y) = p$ $(b)\quad \dfrac{\mathcal{E}_p \xrightarrow{F,a,x} \mathcal{E}'_p \qquad \mathcal{F}_q \xrightarrow{G,a,y} \mathcal{F}'_q \qquad a \in L \cup \{\delta\}}{\mathcal{E}_p \mid[L]\mid \mathcal{F}_q \xrightarrow{F\cup G,a,z} \mathcal{E}'_p[z/x]\backslash G \ \mid[L]\mid\ \mathcal{F}'_q[z/y]\backslash F}$ $z = get\left(\mathcal{M} - \left(\left(\psi(\mathcal{E}) \cup \psi(\mathcal{F})\right) - \left(\psi_H(F) \cup \psi_H(G)\right)\right)\right)\ and\ \pi(z) = \begin{cases} p\ if\ \tau_p \leq \tau_q \\ q\ otherwise \end{cases}$	
$(a)\quad \dfrac{\mathcal{E}_p \xrightarrow{F,a,x} \mathcal{E}'_p \qquad a \neq \delta}{\mathcal{E}_p\ [>\ \mathcal{F}_q \xrightarrow{F,a,y} \mathcal{E}'_p[y/x]\ [>\ \mathcal{F}_q \backslash F}$ $\quad (b)\quad \dfrac{\mathcal{E}_p \xrightarrow{F,\delta,x} \mathcal{E}'_p \qquad \forall z \in \psi(\mathcal{F}_q)}{\mathcal{E}_p\ [>\ \mathcal{F}_q \xrightarrow{F,\delta,y} \mathcal{E}'_p[y/x]\ \ [>\ {}_{\psi_{DC}(\mathcal{F}_q)-F_z}[stop]}$ $\quad \pi(y) = p$ $(c)\quad \dfrac{\mathcal{F}_q \xrightarrow{F,a,x} \mathcal{F}'_q \qquad \forall z \in \psi(\mathcal{E}_p)}{\mathcal{E}_p\ [>\ \mathcal{F}_q \xrightarrow{F,a,y} {}_{\psi_{DC}(\mathcal{E}_p)-F_z}[stop]\ \ [>\ \mathcal{F}'_q[y/x]}$ $\quad \pi(y) = q$ $y = get\left(\mathcal{M} - \left(\left(\psi(\mathcal{E}) \cup \psi(\mathcal{F})\right) - \psi_H(F)\right)\right)$	



$$slope_{xy} = \frac{\tau_q(t)}{\tau_p(t)}, \forall t \in \mathbb{R}_{>0}$$

Figure 3: Two Clocks Evolution with Different Speeds

Definition (*equivalence relation* ~). We define the equivalence relation ~ over the set of all clock valuations, $v \sim v'$ iff all the following conditions hold:

1. For all $x \in H$, either $\lfloor v(x) \rfloor$ and $\lfloor v'(x) \rfloor$ are the same, or both $v(x)$ and $v'(x)$ are greater than c_x.
2. For all $x,y \in H$ with $v(x) \leq c_x$, $v(y) \leq c_y$ and x (resp. y) evolves according to τ_p (resp. τ_q):
    - $c.\frac{1}{slope_{xy}} \leq v(x) \leq (c+1).\frac{1}{slope_{xy}}$ $iff\ c.\frac{1}{slope_{xy}} \leq v'(x) \leq (c+1).\frac{1}{slope_{xy}}$ $for\ c \in \mathbb{N}$
    - $fract\left(slope_{xy} v(x)\right) \leq fract\left(v(y)\right)$ $iff\ fract\left(slope_{xy} v'(x)\right) \leq fract\left(v'(y)\right)$

An equivalence class of clock valuations induced by ~ is a *clock region* of A.

The Representation of Clock Regions

Each region can be specified by the finite set of clock constraints it satisfies. The notation $[v]$ represents the clock region to which v belongs. In Figure 4, the region $[0<y<2x<1]$ contains all the clock valuations that satisfy $(0<y<2x<1)$.

Example 1. Consider two clocks x and y which evolve at different rates such that $slope_{xy}=2$, $c_x=2$ and $c_y=2$. The clock regions obtained by the relation ~ are depicted in Figure 4, containing 15 corner points, 38 open line segments and 23 open regions.

Definition (*slope*$_{max}$). For each clock $x \in H$, we define $slope_{max}(x)$ as the largest value of $slope_{xy}$ for all $y \in H$.

In the example 1:

- $slope_{max(x)}=max(slope_{xy}, slope_{xx})=max(2/1,1)=2$.
- $slope_{max(y)}=max(slope_{yx}, slope_{yy})=max(1/2,1)=1$.

Note that if x is the fastest clock in H then $slope_{max(x)}=slope_{xx}=1$.

The smallest amount of time in which x cannot stay in the same region is $1/slope_{max(x)}$. In the example 1, the clock x changes the region every half unit of time corresponding to $1/slope_{max(x)}=1/2$, whereas y does this change every one unit of time (except for regions represented by points).

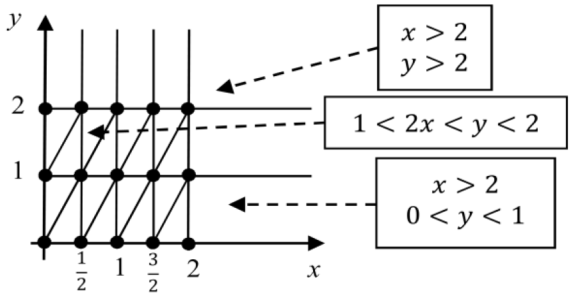


Figure 4: Clock Regions Deducted by the Relation ~

The representation of a clock region accords with the two following points:

1. For each clock x which evolves according to τ_p, there is one clock constraint taken from the set:

$$\{x = c \mid c = 0, \frac{1}{slope_{xy}}, 2\frac{1}{slope_{xy}}, \dots, 1, 1+\frac{1}{slope_{xy}}, 1+2\frac{1}{slope_{xy}}, \dots, c_x \ \ for\ all\ y \in H\}$$
$$\cup \left\{ \bigwedge_{y\in H}\left(c-\frac{1}{slope_{xy}} < x < c\right) \middle| \begin{array}{l} c = \frac{1}{slope_{xy}}, 2\frac{1}{slope_{xy}}, \dots, 1, 1+\frac{1}{slope_{xy}}, \\ 1+2\frac{1}{slope_{xy}}, \dots, c_x \ \ for\ each\ y \end{array} \right\}$$
$$\cup \{x > c_x\}. \qquad (1)$$

2. For each pair of clocks x and y which evolve respectively according to τ_p and τ_q such that $c<x<c+(1/slope_{max(x)})$ and $d<y<d+(1/slope_{max(y)})$ appear in (1) for some c and d, whether $slope_{xy}(x-c)$ is less than, equal to or greater than $(y-d)$.

The time-successors of clock regions

In the following, we introduce the time-successor relation over clock regions. When time advances from any clock valuation v in the region α, we will reach all its time-successors α'. Formally, we say that α' is a time-successor of the region α if there are v in α, v' in α', $t\in\mathbb{R}_{>0}$ such that $v'=v\oplus\tau(t)$, with $v\oplus\tau(t)=(v(x)+\tau_p(t))_{\pi^{-1}(x)=p}$.

For example, in Figure 4, the five time-successors of the region α=[(1,5<x<2),(1<y<2x–2)] are: itself, [(x=2),(1<y<2)], [(x>2),(1<y<2)], [(x>2),(y=2)] and [(x>2),(y>2)]. These regions are those covered by a line drawn from any point in α parallel to the line y=slope$_{xy}x$=2x (in the upwards direction).

To compute each time-successor of a region α, we must give (i). for every clock x, a constraint of the form $(x=c)$ or $(c<x<c+(1/slope_{max(x)}))$ or $(x>c_x)$ and (ii). for every pair x and y such that $(c<x<c+(1/slope_{max(x)}))$ and $(d<y<d+(1/slope_{max(y)}))$ appear in (i), the ordering relationship between $slope_{xy}(x-c)$ and $y-d$.

The possible time-successors are now presented according to 3 cases :

First, each clock x in the region α satisfies the constraint $(x>c_x)$: in this case there is only one time-successor to α which is α itself. This is the case of region [(x>2),(y>2)] in Figure 4.

Second, this case is considered when there is at least, in the region α, one clock x which satisfies the constraint $x=c$ for some $c\le c_x$. The set H_0 contains all clocks appearing in similar constraint form as x. The clock region α will be changed immediately when the time progresses, because the fractional part of each clock in H_0 becomes different from zero. The clock regions α and β have the same time-successors, s.t. β is specified by :

1. A set of clock constraints specified as follows:
   a. For each clock $x\in H_0$:
      i. If α satisfies $(x=c_x)$ then β satisfies $(x>c_x)$;
      ii. If α satisfies $(x=c)$ then β satisfies $(c<x<c+(1/slope_{max(x)}))$.
   b. For each clock $x\notin H_0$, the clock constraint in α remains the same in β.
2. The ordering relation between $slope_{xy}(x\text{-}c)$ and $(y-d)$ of each pair of clocks x, y in α is the same as that in β, such that $x<c_x$ and $y<c_y$ hold in the region α.

For example, in Figure 4, the time-successors of the region [(x=0),(0<y<1)] are the same as the time-successors of the region [0<2x<y<1].

Finaly, If the first and the second case do not apply, then let H_0 be the set of clocks x for which the region α satisfies two constraints $c<x<c+(1/slope_{max(x)})$ and $slope_{xy}(x\text{-}c) \ge (y-d)$ for all clocks y for which the region α satisfies $d<y<d+(1/slope_{max(y)})$. Thus, when the time progresses, the clocks in H_0 take the values $c+(1/slope_{max(x)})$. Therefore, the time-successors of the region α are the 2 regions α and β, together with all the time-successors of β , such that β is specified by :

1. A set of clock constraints, described as follows:
   a. For each clock $x\in H_0$, if α satisfies $c<x<c+(1/slope_{max(x)})$ then β satisfies $(x=c+(1/slope_{max(x)}))$;
   b. For each clock $x\notin H_0$, the clock constraint in α remains the same in β.
2. For each pair of clocks x and y such that both intervals $(c<x<c+(1/slope_{max(x)}))$ and $(d<y<d+(1/slope_{max(y)}))$ appear in (1.b), the ordering relationship between $slope_{xy}(x\text{-}c)$ and $(y-d)$ in α remains the same in β.

For example, the time-successors of the region [0<2x<y<1] in Figure 4, include the region itself, the region [(0<x<0,5),(y=1)] and all the time-successors of [(0<x<0,5),(y=1)].

Algorithm (region automaton).

Let A=((S,L_s,s_0,H,T) ,π) be a DATA-R over the set of agents Ag. The region automaton $R(A)$ is an automaton over the alphabet Act such that

- The configurations of $R(A)$ are of the form <s|α> where s is a state of A and α is a clock region. The initial configuration is of the form <s_0|[v_0]> where $v_0(x)$=0 for every $x \in H$.
- A transition of $R(A)$, from the configuration <s|α> to <s'|α'>, is labeled by $a \in Act$ iff there is a transition (s,G,a,x,s') in T and a clock region α" which satisfies
  - α" is a time-successor of the region α.
  - $\alpha" \models G$.
  - $\alpha' = [\{x\} \mapsto 0]\ \alpha"$.

Figure 5 highlights a part of the region graph of the frame example.

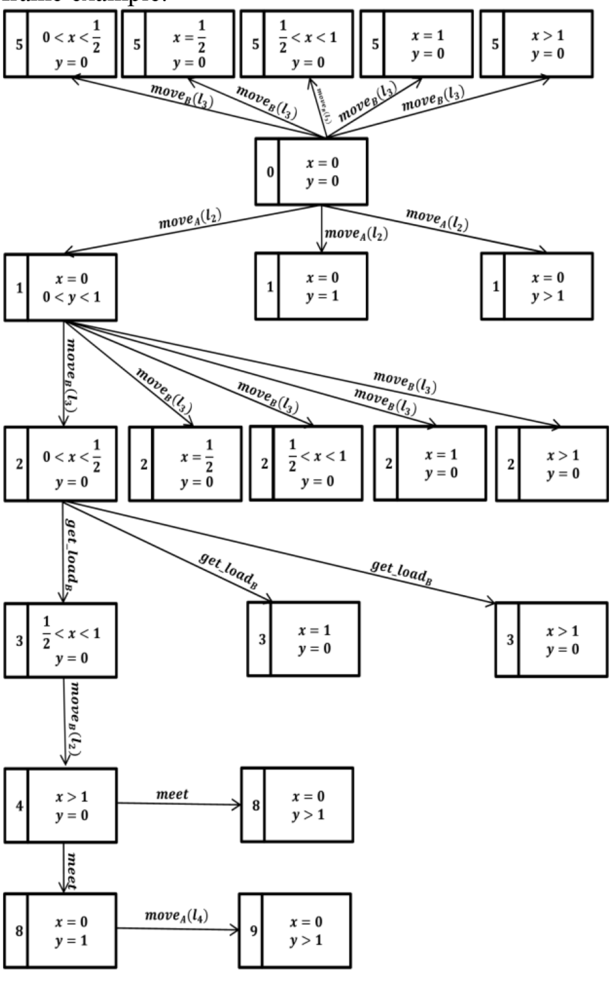


Figure 5. A Region Graph

CONCLUSIONS AND PERSPECTIVES

The proposed algebraic language extends the standard LOTOS specification, with a capacity of describing timed plans with action durations, that can be shared in between some coordinated agents.

Although agents can be heterogeneous, they can reasonably be (re)synchronized to start the execution of any coordinated plan and that they can behave under relative time rates.

Taking benefit from the semantics of such a specification, we demonstrated how to build a finite (time) region graph, which is bisimilar in respect to the infinite set of behaviors implied by the specification. Compared to existing approaches, our technique benefits from true concurrency properties, without increasing the size complexity of the region graph.

To examplify our technique, we applied it to a simple but realistic ambient use case, wherein the coordination of trucks is required. Agents are currently reduced to having one clock, however, the extension to several is immediate.

Nevertheless, interesting concepts have been already developed through other models, that can be compatible with the computation of regions, e.g. [hybrid automata][priced automata]. This motivates us to investigate the possibility of integrating similar concepts within our approach.

REFERENCES

Alur, R. and D. L. Dill. 1994. "A theory of timed automata." *Theoretical computer science*, 126(2), 183-235.

Belala, N. and D.-E. Saïdouni. 2005. "Non-atomicity in timed models." In *Proceedings of International Arab Conference on Information Technology (ACIT 2005)*, Jordan.

Chaouche, A.-C.; A.-E. Seghrouchni; J.-M. Ilié and D.-E. Saïdouni. 2014. "A Higher-Order Agent Model with Contextual Planning Management for Ambient Systems." *LNCS, vol. 8780, pp 146-169, journal TTCI.*

Courtiat, J.-P. and D.-E. Saïdouni. 1995. "Relating Maximality-based Semantics to Action Refinement in Process Algebras." In *Proceedings of FORTE'94*, pp. 293-308, Chapman and Hall.

Guivarch, V.; V. Camps and A. Pninou. 2012. "Context awareness in ambient systems by an adaptive multi-agent approach." In *Ambient Intelligence*. Volume 7683 of LNCS. Springer Berlin Heidelberg 129-144.

Maarouk, T.-M.; D.-E. Saidouni and M. Khergag. 2011. "DD-LOTOS: A distributed real time language." In *Proceedings 2nd Annual International Conference on Advances in Distributed and Parallel Computing (ADPC 2011) Special Track: Real Time Embedded Systems* (RTES 2011) pp. 45-50.

Pi, N.S.; D. Griol; J. Carbo and J.-M.-M. Lopez. 2013. "Evaluation of agents interactions in context-aware system." *T. Computational Collective Intelligence* 9 79-97.

Saïdouni D.-E. 1996. "Sémantique de Maximalité: Application au Raffinement d'Actions en LOTOS." *PhD Thesis*, LAAS-CNRS, Toulouse, France.