



**HAL**  
open science

## Spatio-Temporal Planning for Mobile Ambient Agents

Radja Boukharrou, Jean-Michel Ilié, Djamel Eddine Saïdouni

► **To cite this version:**

Radja Boukharrou, Jean-Michel Ilié, Djamel Eddine Saïdouni. Spatio-Temporal Planning for Mobile Ambient Agents. 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015), Aug 2015, Belfort, France. pp.96-103, 10.1016/j.procs.2015.07.174 . hal-01228874

**HAL Id: hal-01228874**

**<https://hal.science/hal-01228874>**

Submitted on 12 Feb 2017

**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.

The 12th International Conference on Mobile Systems and Pervasive Computing  
(MobiSPC 2015)

## Spatio-Temporal Planning for Mobile Ambient Agents

Radja Boukharrou<sup>a,\*</sup>, Jean-Michel Ilié<sup>b</sup>, Djamel Eddine Saïdouni<sup>a</sup>

<sup>a</sup>University Abdelhamid Mehri - Constantine 2, MISC Lab., Campus Ali Mendjeli, 25000 Constantine, Algeria

<sup>b</sup>Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris, France

---

### Abstract

The algebraic language Time-AgLOTOS was recently proposed to describe the time-dependent behavior of an ambient intelligent agent. The Spatio-Temporal Planning System (STPS) is a contextual model capturing all possible evolutions of an agent plan including context changes. It provides formal description of the possible actions of a plan supporting timing constraints, action duration and spatial requirements. Thereby, it can be derived from Time-AgLOTOS behavior expressions. In this paper, we propose a finite and symbolic representation of the STPS based on a number of spatio-temporal regions, preserving both time progress and location modeling. The resulting structure offers new possibilities and strategies for taking agent real-time decisions in context-awareness manner.

© 2015 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the Conference Program Chairs.

*Keywords:* Real-time mobile agents, spatio-temporal planning, location modeling, action duration

---

### 1. Introduction

Nowadays, the software engineering of Ambient Intelligence (AmI) systems is widely investigated. The needs comes from new applications which aim at taking profit from the ubiquitous computing, often involving the human life in its all, like in the development of smart and assisting environments<sup>1</sup>. As some major challenges, the building of intelligent systems requires holding together adaptation and sustainability properties within open, non-deterministic and uncertain environment. Several modeling approaches are already proposed to assist the designer of AmI systems. In fact, the major problem for the system entities consists in recognizing environmental context, including location identification, resource management, real-time planning, discovery of other agents, and handling information in a more *semantic* manner e.g.<sup>2,3</sup>.

Face to such a complex framework, several research works propose to make use of intelligent agents, to support the AmI systems, e.g.<sup>4,5</sup>. At a first glance, this could include some well-known Multi-Agent System (MAS) approaches, BDI agent are designed to take rational decisions as practical reasoning<sup>6</sup>. However, this is not enough when conditions

---

\* Corresponding author. Tel.: +213-777-212-676 ; fax: +213-31-895-032.

E-mail address: [boukharrou@misc-umc.org](mailto:boukharrou@misc-umc.org)

are highly dynamic like in AmI systems. This assumes an efficient context-awareness ability, dealing with unexpected situations and changes of context.

In the literature, different definitions were proposed to qualify the context of the agent. In<sup>7</sup>, four types of context information are defined: (1) *computational context*—available resources, network quality and related information; (2) *user context*—profile of the user, people nearby, social situation; (3) *physical context*—lighting, temperature, traffic conditions, noise levels, etc., and (4) *time context*—time of day, date of the year. As two other types of contexts, the work of<sup>8</sup> introduces the *activity* of some agent where in<sup>7</sup>, the *context history* is proposed.

In practice, most context modeling approaches only focus on a subset of the above types, depending on the requirements of the applications. In<sup>9</sup>, the authors focus on sensor-based agents, since the aim is to learn physical measures and control the hardware devices of an automated house (lighting, warming...). To deal with mobility or real-time object tracking, spatial location are referred e.g.<sup>10,11</sup>. Dealing with both spatial and temporal information is useful each time. However, space and time cannot be reduced easily from one to each other. The real challenge in AmI systems is the reasoning about spatial change, the modeling of spatio-temporal interactions and planning.

In this paper, we aim at providing a behavioral analyzing technique of agents handling spatio-temporal contexts. To this end, we follow the work of<sup>12</sup> which introduces a timed action model, called STPS, to represent the possible plans of the agent, w.r.t. its set of intentions viewed as concurrent processes. In this model, the spatio-temporal properties relate to the simultaneous progression of time and location change implied by the performances of actions. Nevertheless, the STPS could be an infinite structure, since time ranges over a dense domain.

To allow model-checking of spatio-temporal properties, we now investigate the building of a finite representation of the STPS. Based on symbolic spatio-temporal regions, it preserves both the STPS time progression and location changes.

The remaining of the paper is organized as follows: Section 2 introduces the Time-AgLOTOS language providing a planning taking into account timing constraints and duration of actions. This language is used to associate plans with intentions. In Section 3, the underlying spatio-temporal model, called STPS, is automatically produced by applying the true-concurrency semantics of Time-AgLOTOS. In Section 4, we show how to build a finite and symbolic graph of spatio-temporal regions from the STPS. Throughout the paper, the scenario is taken up as an illustration of our approach. The last section concludes and outlines our perspectives.

## 2. Time-AgLOTOS: An algebraic Language for Plan Specification

### 2.1. Agent Plan Structure

In the approach of<sup>5</sup>, an *agent plan* is structured as a tree structure within three level planning representations. The *agent plan* is obtained by composition of sub-plans, called *intention plans*, where each one is dedicated to achieve its corresponding intention ; Each intention plan is a composition of alternative sub-plans, called *elementary plans*. These elementary plans are assumed to be extracted from a library of plans, here called *LibP library*. They allow us to consider different ways to achieve the associated intention. The *Time-AgLOTOS* language, introduced in<sup>12</sup>, deals with modular and concurrent aspects to compose and schedule these different sub-plans, viewed as processes. Let us briefly recall the *Time-AgLOTOS-based specifications* for plans.

**Agent plan level.** The set of intention plans can be handled globally, by using the concurrent  $\parallel$  and/or sequential  $\gg$  operators between intention plans. This leads to the specification of an agent plan. Let  $\bar{\mathcal{P}}$  be the set of names qualifying the possible agent plans with  $\bar{P} \in \bar{\mathcal{P}}$  and let  $\hat{\mathcal{P}}$  be the set of names used to identify the possible intention plans with  $\hat{P} \in \hat{\mathcal{P}}$ , such that  $\bar{P}$  is any agent plan defined by:

$$\bar{P} ::= \hat{P} \mid \bar{P} \parallel \bar{P} \mid \bar{P} \gg \bar{P}$$

**Intention plan level.** An intention plan corresponding to an alternative of elementary plans is specified by using the composition operator  $\diamond$ . The associated intention is considered to be achieved iff at least one of the associated elementary plans is successfully terminated. Formally, we define an intention plan  $\hat{P}$  as:

$$\hat{P} ::= P \mid \hat{P} \diamond \hat{P}$$

**Elementary plan level.** Elementary plans are described by behavior expressions, referring to a finite set of actions in which both timing constraints and action duration are considered. However, the language allows the specification of dynamism and context-awareness of AmI agents. For that, let  $\Theta$  be a finite set of space locations where an agent can move and  $\Lambda$  be the set of agents with which he can communicate. In addition, let  $Act = O \cup \{\tau\}$ , be the set of actions, where  $O$  is a (finite) set of observable actions which are viewed as instantiated predicates, ranging over  $a, b, \dots$  and  $\tau$  is the internal action.

Each elementary plan is identified by a name ( $P$ ) and is featured by a behavior expression ( $E$ ) taking into account time aspects. The syntax of an elementary plan is defined inductively as follows:

$$\begin{aligned}
P &::= E \\
E &::= \text{exit}\{d\} \mid \text{stop} \mid a\{d\};E \mid a@t[SP];E \quad (a \in O, d \in \mathbb{T}) \\
&\quad \mid \Delta^d E \mid E \odot E \mid \text{hide } L \text{ in } E \\
H &::= \text{move}(\ell) \mid x!(v) \mid x?(v) \quad (H \subset O, \ell \in \Theta, x \in \Lambda) \\
\odot &= \{ [], \gg, [>, \llbracket L \rrbracket, \parallel, \llbracket \rrbracket \}
\end{aligned}$$

The syntax of a Time-AgLOTOS accords with the basic construction of LOTOS, where expressions are composed either sequentially, by prefixing some expression with an action (like  $a; E$ ) or concurrently between sub-expressions by using with a composition operator of the set  $\odot$ . The expression  $E [] E$  specifies a non-deterministic choice,  $E \gg E$  a sequential composition and  $E [> E$  the interruption. The LOTOS parallel composition, denoted  $E \llbracket L \rrbracket E$ , can model *both synchronous* composition for actions defined in  $L$ , denoted  $E \parallel E$  with  $L = O$ , and *asynchronous composition*, denoted  $E \lll E$  with  $L = \emptyset$ . The expression *hide*  $L$  in  $E$  represents an explicit hiding of actions mentioned in  $L$ , making them unobservable w.r.t.  $E$ . Lastly, the basic expression *stop* explicitly specifies an expression without possible evolution and *exit* explicitly specifies the successful termination of some expression.

In Time-AgLOTOS, every action is assumed to have a non-null duration, however only some temporal constraints are defined associated with the actions or the expressions of the elementary plans. Let  $\mathbb{T}$  be a domain of time like  $\mathbb{Q}^+$  or  $\mathbb{R}^+$ .  $\Upsilon : Act \rightarrow \mathbb{T}$  is the duration function which associates to each action its duration. Let  $d \in \mathbb{T}$  a value in the temporal domain. For any action  $a$ ,  $a\{d\}$  expresses a temporal restriction specifying that the launching of the action  $a$  must be in-between  $[0, d]$ ; The notation  $a@t[SP]; E$  is simply more general ( $a \in O$ ). It means that the starting interval of  $a$  is specified by the *selection predicate*  $SP$ . For sake of clarity in this paper, we restrict  $SP$  to  $SP = \text{min}\sim t \sim \text{max}$ , such that  $\sim \in \{<, \leq\}$  with  $\text{min}, \text{max} \in \mathbb{T}$ . Here,  $t$  represents a temporal variable attached to  $a$ , which is used to record the time past since the enabling to start  $a$ , and which will be substituted by zero when this action ends its execution. Lastly, the notation  $\Delta^d E$  represents a time delay to be respected before performing the actions of  $E$ .

Thereby, the elementary expression *stop* specifies a behavior without possible evolution and *exit* $\{d\}$  represents the successful termination of some plan which must be taken within the interval  $[0, d]$ . We assume also that *stop* and *exit* are of null duration.

## 2.2. Building Agent Plan from Intention set

In order to account for agent intentions, we propose that the agent can label the different elements of the set  $I$  of intentions (We assume that the agent, through its BDI attitudes, can solve conflicting situations that could arise between intentions for some context, by means of a scheduling process applied to the set of intentions) by using a weight function  $\text{weight} : I \rightarrow \mathbb{N}$ . The ones having the same weight are composed by using the parallel operator  $\lll$ . In contrast, the intention plans corresponding to distinct weights are ordered by using the sequential operator  $\gg$ . For instance, let  $I = \{i_g^2, i_e^1, i_m^2\}$  be the considered set of intentions, such that the superscript information denotes a weight value, and let  $\widehat{P}_g, \widehat{P}_e, \widehat{P}_m$  be their corresponding intention plans, the constructed agent plan could be viewed as:  $\widehat{P} = (\widehat{P}_g \lll \widehat{P}_m) \gg \widehat{P}_e$ , where  $\widehat{P}_g = P_g$ ,  $\widehat{P}_m = P_{m1} \diamond P_{m2}$  and  $\widehat{P}_e = P_{e1} \diamond P_{e2} \diamond P_{e3}$ . Further, each elementary plan  $P_k$  is associated with an expression  $E_k$  describing its behavior. For instance, let us simply consider an elementary plan  $P_g$ , achieving a travel scenario consisting of two sequential tasks, first, getting ticket from the "travel agency" in location  $\ell_1$  ( $\text{getT}(\ell_1)$ ), then move to the "airport" in location  $\ell_2$ , in order to board the plane ( $\text{move}(\ell_2)$ ). The corresponding expression is  $E_g = \text{getT}(\ell_1)\{1\}; \text{move}(\ell_2)@t[1 \leq t \leq 2]; \text{exit}\{0\}$ . The action  $\text{move}(\ell_2)@t[1 \leq t \leq 2]$  means that, before moving, the agent must wait between 1 and 2 time units. It can be assumed for instance that the move is supported by

a taxi called from the agency, with an arrival estimation comprised in the time unit interval  $[1, 2]$ . Moreover, the move from  $\ell_1$  to  $\ell_2$  is assumed to take  $\Upsilon(\text{move}(\ell_2)) = 1$  time unit.

### 2.3. Planning State of the agent

The state of an agent plan, also called an *agent plan configuration*, is denoted like  $\overline{P}$ . According to Definition 2.1, it consists of a behavior expression, built from the expressions of the intention plan level, themselves built from the expressions attached to the elementary plans. Considering the former example,  $\overline{P} = (E_g, P_g) \parallel (E_{m1} \diamond E_{m2}, \widehat{P}_m) \gg (E_{e1} \diamond E_{e2} \diamond E_{e3}, \widehat{P}_e)$ .

**Definition 2.1.** Any Agent plan configuration  $\overline{P}$  has a canonical representation defined by the following two rules:

$$(1) \frac{\overline{P} ::= \widehat{P} \quad \widehat{P} ::= \diamond^{k=1..n} P_k \quad P_k ::= E_k}{[\overline{P}] ::= (\diamond^{k=1..n} E_k, \widehat{P})} \quad (2) \frac{\overline{P} ::= \overline{P}_1 \odot \overline{P}_2 \quad \odot \in \{\parallel, \gg\}}{[\overline{P}] ::= [\overline{P}_1] \odot [\overline{P}_2]}$$

The formal semantics defined in<sup>12</sup> allows to investigate all the different ways to make an agent plan evolve. For instance, due to the parallel operator ( $\parallel$ ), the configurations reached from  $\overline{P}$  can be derived, either from the operational semantics of  $E_g$  or from the one of  $(E_{m1} \diamond E_{m2})$ . This principle is extended and exemplified in the next section.

## 3. Spatio-Temporal Model

We define a spatio-temporal model able to make different executions evolve in true-concurrency, either in a temporal point of view or a spatial one or both. In this paper, we accord with the maximality semantics<sup>13</sup> which concentrates on the starting of actions, letting the termination being handled dynamically. Most of actions in execution make the time progress, however, spatio-temporal changes can occur when dealing with *move* actions. The starting of an action *move* does not directly capture the change of location, but the fact to progress to a target location, which is only reached when the duration of the move is achieved. The next definitions describe our spatio-temporal model as a transition system, assuming that they cannot be two occurrences of *move* action concurrently. The evolutions are driven contextually in the sense that any action can only be enabled in some context, and that the performance of an action makes the context evolves. This model is only predictive therefore the execution of an action is assumed to always be a success.

### 3.1. Spatio-Temporal Planning State

With respect to some evolution, the so-called *spatio-temporal planning state* of the agent, takes into account the agent location, the duration conditions of actions possibly in execution, and an information about the intention plans that are (already) achieved. The location information can also handle the fact that a move is in progress, by using the following notion of *expected location*.

**Definition 3.1.** An expected location is represented by a pair  $\langle \ell, x \rangle$  with  $\ell \in \Theta$  and  $x \in \mathcal{H}$ . It captures two cases:  $\langle \ell, nil \rangle$  defines the fact that the agent is in some location  $\ell$ , whereas  $\langle \ell, x \rangle$  specifies that the agent is being moved to  $\ell$  by performing the action  $\text{move}(\ell)$  and  $x$  is the clock associated with this performance. For sake of concision, the expected location  $\langle \ell, x \rangle$  is denoted  $\ell^x$ , whereas  $\langle \ell, nil \rangle$  is simply denoted  $\ell$ .

**Definition 3.2.** A spatio-temporal planning state is a tuple  $(ps, \ell^x, DC, T)$ , where  $ps$  is any plan configuration  $\overline{P}$ ,  $\ell^x$  is an expected location,  $DC$  is the set of duration conditions associated to the actions in execution, and  $T$  is the subset of intention plans which are terminated.

### 3.2. Spatio-Temporal Planning System

The possible changes from the spatio-temporal planning states are obtained by reusing the operational semantic rules of<sup>12</sup>, but with the considerations of the location information and the intention plans that are terminated. This

yields a *Spatio-Temporal Planning System* built from the initial spatio-temporal planning state  $(ps_0, \ell, \emptyset, \emptyset)$  where  $ps_0 = [\bar{P}]$  is the initial planning state configuration.

Let  $\mathcal{H}$  be the set of clocks with non-negative values within the time domain  $\mathbb{T}$ . The set  $\Phi_t(\mathcal{H})$  of temporal constraints ( $\gamma$ ) over  $\mathcal{H}$  is defined by  $\gamma ::= x \sim t$  where  $x$  is a clock in  $\mathcal{H}$ ,  $\sim \in \{=, <, >, \leq, \geq\}$  and  $t \in \mathbb{T}$ . A *valuation*  $v$  for  $\mathcal{H}$  is a function which associates to each  $x \in \mathcal{H}$  a value in  $\mathbb{T}$ . The set of all valuations for  $\mathcal{H}$  is noted  $\Xi(\mathcal{H})$ . A valuation  $v$  satisfies a clock constraint  $\gamma$  over  $\mathcal{H}$  if and only if  $\gamma$  is *true* by using clock values given by  $v$ . The satisfaction relation  $\models$  for temporal constraints is defined over the set of valuations for  $\mathcal{H}$  by:  $v \models x \sim t \Leftrightarrow v(x) \sim t$  such that  $v \in \Xi(\mathcal{H})$ . For any subset  $\mathcal{U} \subseteq \mathcal{H}$ ,  $[\mathcal{U} \rightarrow 0]v$  indicates the valuation for  $\mathcal{H}$  which assigns 0 to each  $x \in \mathcal{U}$ , and agrees with  $v$  over the other clocks of  $\mathcal{H}$ . For all  $x \in \mathcal{H}$ ,  $C_x$  is the largest constant with which  $x$  is compared within some temporal constraint, and  $\lfloor v(x) \rfloor$  denotes the integral part of the clock valuation where  $frac(v(x))$  denotes the fractional part, i.e.,  $v(x) = \lfloor v(x) \rfloor + frac(v(x))$ .

**Definition 3.3.** A *Spatio-Temporal Planning System (STPS for short)* is a tuple  $\Omega = \langle S, s_0, \mathcal{H}, Tr, \mathcal{L}, \mathcal{DC}, \mathcal{T} \rangle$  where:

- $S$  is a finite set of contextual planning states such that  $s = (ps, \ell^x, T) \in S$ ,
- $s_0 = (ps, \ell, \emptyset) \in S$  is the initial contextual planning state, such that  $ps = [\bar{P}]$  and  $\ell$  represents the current location of the agent,
- $\mathcal{H}$  is a finite set of clocks,
- $Tr \subseteq S \times 2^{\Phi_t(\mathcal{H})} \times 2^{\Phi_t(\mathcal{H})} \times Act \times \mathcal{H} \times S$  is the set of transitions of the form  $(s, G, D, a, x, s')$ ,
- $\mathcal{DC} : S \rightarrow 2^{\Phi_t(\mathcal{H})}$  is a function which corresponds to each state  $s$  the set of duration conditions of actions possibly in execution in  $s$ . Duration conditions are temporal constraints of the form  $x \geq t$  where  $x \in \mathcal{H}$  and  $t \in \mathbb{T}$ ,
- $\mathcal{L} : S \rightarrow \Theta \times \mathcal{H}$  is a function which yields the expected location  $\langle \ell, x \rangle$  in each state  $s$ ,
- $\mathcal{T} : S \rightarrow 2^{\bar{P}}$  is a function which captures the terminated intention plans in each state  $s$ .

A transition  $(s, G, D, a, x, s')$ , also denoted  $s \xrightarrow{G, D, a, x} s'$ , represents a change from the state  $s$  to the state  $s'$  by starting an occurrence of execution of  $a$  and by resetting the clock  $x$  assigned to this occurrence.  $G$  is the guard which must be satisfied to consider this transition.  $D$  is the deadline which forces the action  $a$  to occur, at the moment of its satisfaction. We assume that  $(\models D) \Rightarrow (\models G)$  and  $D$  is of the form  $x \geq t$ . Moreover, for each state  $s \in S$  where a move occurs, we define  $\mathcal{L}(s) = \ell^x$  as the expected location to be reached after the move characterized by the clock  $x$ .

Considering the example of Section 2, Figure 1 brings out a partial view of the STPS obtained from  $([\bar{P}])$ , considering that  $(E_g, \bar{P}_g)$  is totally processed from the initial state  $([\bar{P}], \ell, \emptyset)$ , where  $E_g = getT(\ell_1)\{1; move(\ell_2)@t[1 \leq t \leq 2]; exit(0)$ .

For instance, the state  $s_1$  captures the current action under execution ( $getT$ ) through its corresponding clock  $x$  such that the duration condition of  $getT$  is  $x \geq 1$ . In  $s_3$ , the reached location is  $\ell_2$  and the intention plan  $\bar{P}$  is successfully terminated. It is worth observing that the clock  $x$  is reused to identify the successive action occurrences. That is made possible here by the fact that the execution of  $move(\ell_2)$  and the one of  $getT$  are exclusive. Actually, they are performed sequentially due to the used operator  $;$  in the expression  $E_g$ . The guard  $G$  of the first transition specifies the delay required before performing the action  $getT$ , whereas the one of the second takes the termination of  $getT$  into account, to launch the action  $move(\ell_2)$ . The third transition takes the termination of  $move(\ell_2)$  into account, to consider the termination of the intention plan  $\bar{P}_g$ . In this case, the action to be performed is simply  $\tau$ .

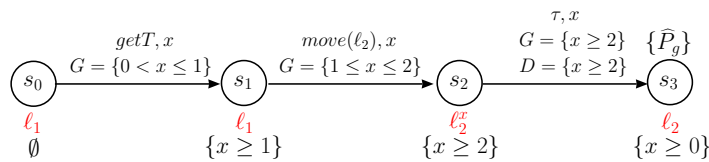


Fig. 1. STPS corresponding to the agent plan  $\bar{P}_1$  of Alice

Since time refers to a dense domain, the different possible evolutions that can be extracted from the STPS could be infinite. The next section copes with this problem.

**Definition 3.4.** The semantics of a STPS  $\Omega = \langle S, s_0, \mathcal{H}, Tr, \mathcal{L}, \mathcal{DC}, \mathcal{T} \rangle$  is an infinite transition system  $\Sigma_\Omega$  defined over  $Act \cup \mathbb{T}$ . A state of  $\Sigma_\Omega$  is a pair  $\langle s, v \rangle$  such that  $s$  is a state of  $\Omega$  and  $v$  is a clock valuation of  $\mathcal{H}$ . A state  $\langle s_0, v_0 \rangle$  is initial if  $s_0$  is the initial state of  $\Omega$  and  $\forall x \in \mathcal{H}, v_0(x) = 0$ . As defined in<sup>14</sup>, two types of transitions are specified between the states of  $\Sigma_\Omega$ . The time passing transitions are defined according to the rules (RA1) and (RA2) whereas the action transitions, which refer to the ones of  $\Omega$ , are related to the rule (RD), where  $\eta \subset \mathbb{T}$  is the set of the smallest quantities of time in which no action occurs.

$$(RA1) \quad \frac{d \in \mathbb{T}, d > \eta \quad \forall d' \leq d, v + d' \not\models \mathcal{D}}{\langle s, v \rangle \xrightarrow{d} \langle s, v + d \rangle} \quad (RA2) \quad \frac{\varepsilon \in \mathbb{T} \quad v + \varepsilon \models \mathcal{D} \wedge \varepsilon \in \eta}{\langle s, v \rangle \xrightarrow{\varepsilon} \langle s, v + \varepsilon \rangle}$$

$$(RD) \quad \frac{(s, G, D, a, x, s') \in Tr \quad v \models G}{\langle s, v \rangle \xrightarrow{a} \langle s', [\{x\} \mapsto 0]v \rangle}$$

According to the maximality semantics, the label  $a$  in (RD) rule implies the start of action  $a$  and not the whole execution of this action. By construction, if  $D$  or  $G$  are satisfied, we deduce that the actions, which the  $a$  depends, have finished their executions. Let  $K$  be the set of all transitions stemming from state  $s$ . The formula  $\mathcal{D} = \bigvee_{k \in K} D_k$  is a disjunction of deadlines such that  $\{(s, G_k, D_k, a_k, x_k, s_k)\}_{k \in K}$ . Indeed, whenever a deadline  $D_k$  holds, time can not progress regardless of the other deadlines in  $\mathcal{D}$ .

#### 4. Spatio-Temporal Region Graph

Because the semantic graph of the STPS can be infinitely large, the analysis and validation techniques of ambient real-time systems could not be applicable. We now show how to convert the STPS into an equivalent *finitely symbolic* transition system called *Spatio-Temporal Region Graph (STRG)* where the state reachability is decidable. The STRG not only preserves the temporal and concurrent properties of the STPS but also the spatial ones.

##### 4.1. Time Region Modeling

In practice, many equivalence relations are already proposed to aggregate configurations in equivalence classes, as in<sup>15,16</sup>. The equivalence classes of the clock valuations are named *clock regions*, and are defined as a subset of  $2^{\Xi(\mathcal{H})}$ .

The following *equivalence relation*, namely  $\approx$ , yields a finite partition over the space of the clock valuations. Two clock valuations  $v$  and  $v'$  are *clock region equivalent* ( $v \approx v'$ ) if and only if the three following conditions hold:

- $\forall x \in \mathcal{H}, (\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor) \vee (v(x) > C_x \wedge v'(x) > C_x)$ ,
- $\forall x \in \mathcal{H}, (v(x) \leq C_x) \Rightarrow ((\text{frac}(v(x)) = 0)) \Leftrightarrow (\text{frac}(v'(x)) = 0))$ , and
- $\forall x, y \in \mathcal{H}, (v(x) \leq C_x \wedge v(y) \leq C_y) \Rightarrow ((\text{frac}(v(x)) \leq \text{frac}(v(y))) \Leftrightarrow (\text{frac}(v'(x)) \leq \text{frac}(v'(y))))$

By considering that the number of clocks  $|\mathcal{H}|$  is fixed w.r.t. the maximum constants with which clocks are compared in the STPS, the finite number of clock regions can be at most  $|\mathcal{H}|! * 2^{|\mathcal{H}|-1} * \prod_{x \in \mathcal{H}} (2C_x + 2)$ , as proved in<sup>16</sup>. For instance, consider a set of two clocks  $\mathcal{H} = \{x, y\}$ , where  $C_x = 2$  and  $C_y = 1$ . The corresponding number of clock regions is 28 (see Figure 2).

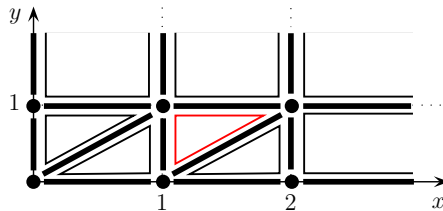


Fig. 2. clock regions: 6 intersections, 14 lines, 8 spaces

Any clock  $x$  in a region can be represented by:  $\{x = t \mid t = 0, 1, \dots, C_x\} \cup \{t-1 < x < c \mid t = 1, \dots, C_x\} \cup \{x > t \mid t = C_x\}$ . Two valuations of the same clock region  $cr$  must satisfy the same constraints, thus allowing the same transitions.

The equivalence of clock regions is compatible with the progression of time in monotonic way. It is therefore possible to define a time-successor function for all the regions. We denote  $succ(cr)$ , all the successors of the clock region  $cr$ , such that the considered elapsing time progression accords with:  $cr' \in succ(cr) \Leftrightarrow \exists v \in cr, \exists t \in \mathbb{T}$  such as  $v + t \in cr'$  and  $v + t' \in cr \cup cr'$  for all  $t' < t$ . For instance, the time-successors of  $cr = [(1 < x < 2), (0 < y < x < 1)]$  (illustrated in Figure 3 by the red triangle) are:  $cr_1 = [(1 < x < 2), (y = 1)]$ ,  $cr_2 = [(1 < x < 2), (y > 1)]$ ,  $cr_3 = [(x = 2), (y > 1)]$  and  $cr_4 = [(x > 2), (y = 1)]$ . Here,  $[v]$  is used to denote the equivalence class of  $\Xi(\mathcal{H})$  such that  $[v] = \{v' \in \Xi(\mathcal{H}) \mid v \approx v'\}$ . Thus, any clock region  $cr$  is defined by  $[v]$ .

#### 4.2. Location Labeling Function

In addition to the clock labeling that comes from the STPS semantics, the states of the region graph are labeled by a location information. If  $\ell$  is the current location of the agent, it can be either a discrete location (e.g.  $\ell = \ell_1$ ) or a spatial path between two discrete locations (e.g.  $\ell_1 < \ell < \ell_2$ ) in order to describe the move of the agent.

#### 4.3. Construction of the Spatio-Temporal Region Graph

The following definition yields the formal structure of a Spatio-Temporal Region Graph (STRG). Figure 3 is an example developed from the STPS of Figure 1.

**Definition 4.1.** Let  $\Omega = (S, s_0, \mathcal{H}, Tr, \mathcal{L}, \mathcal{DC}, \mathcal{T})$  be a STPS, the corresponding Spatio-Temporal Region Graph  $STRG(\Omega) = (S_R, s_{0_R}, Tr_R, \mathcal{LR}, \mathcal{CR}, \mathcal{T}_R)$ , is defined as follows:

- The states in  $S_R$  are of the form  $\langle s, cr, lr \rangle$  where  $s \in S$ ,  $cr$  is a clock region and  $lr$  is a location region,
- The initial state  $s_{0_R} = \langle s_0, [v_0] \rangle$  where  $s_0 = (ps, \ell, \emptyset, \emptyset)$  is the initial state in  $\Omega$  and  $\forall x \in \mathcal{H}, v_0(x) = 0$ ,
- The set of transitions  $Tr_R$  is composed of two types of transitions:
  - the transitions which represent the time passing:  $\langle s, cr, lr \rangle \longrightarrow \langle s, cr', lr' \rangle$  if and only if  $cr'$  is a time-successor of  $cr$  and  $cr' \neq \mathcal{D}$ ,
  - the transitions which are in correspondence with the transitions of  $\Omega$  such that:  $\langle s, cr, lr \rangle \xrightarrow{a,x} \langle s', cr', lr' \rangle$  if and only if  $\exists (s, G, D, a, x, s') \in Tr, cr \models G$  and  $cr' = [\{x\} \rightarrow 0]cr$ ,
- The labeling functions of region states  $\mathcal{LR}$  and  $\mathcal{CR}$  yield the clock and location regions. Hence, for each  $\langle s, cr, lr \rangle \in S_R$ , we have  $\mathcal{LR}(\langle s, cr, lr \rangle) = lr$  and  $\mathcal{CR}(\langle s, cr, lr \rangle) = cr$ ,
- The function  $\mathcal{T}_R$  extends the one already defined in the STPS, such that  $\mathcal{T}_R(\langle s, cr, lr \rangle) = \mathcal{T}(s)$ .

## 5. Conclusion

The formal semantics of the Time-AgLOTOS algebraic language for specifying agent plans, was used to associate a Spatio-Temporal Planning System (STPS). Hence, from some current set of weighted intentions, this allows one to capture all the possible evolutions of the agent plan on-the-fly. The STPS looks like a true-concurrency model of actions, enhancing the terminations of the executions of the intentions, with regards to the various timing constraints and the spatial environmental context, required when performing the actions.

Because the semantic graph of a STPS could be infinite, we showed how to make use of clock value regions, leading to a finite structure called Spatio-Temporal Region Graph (STRG), over which time and spatial properties can be checked. By considering a finite number of clocks, it is easy to assert that the STPS is necessarily finite, and so does the STRG. Nevertheless, there may be numerous states developed in the STRG without any action to perform, therefore we currently investigate how to compact this structure.



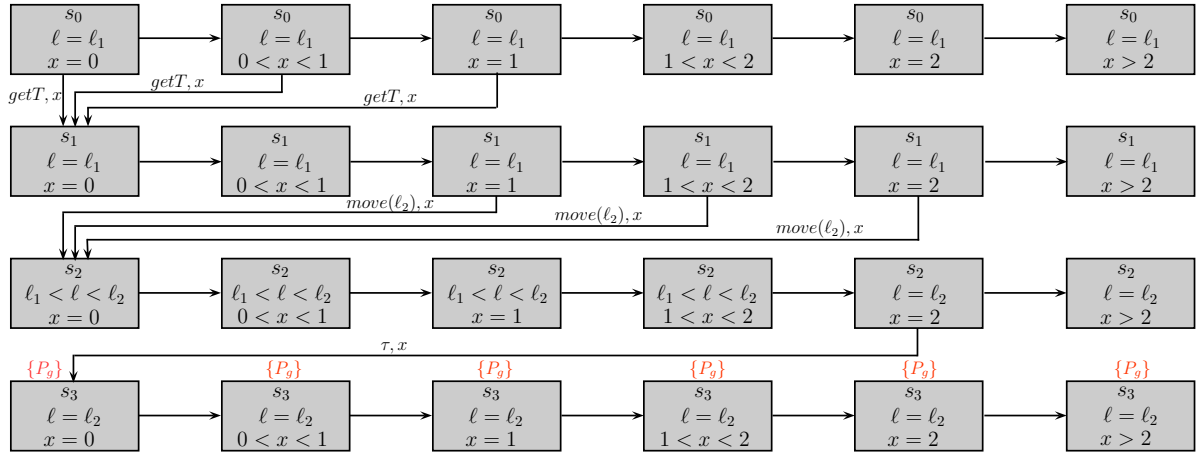


Fig. 3. The Spatio-Temporal Region Graph corresponding to the scenario

## References

1. Nakashima, H., Aghajan, H.K., Augusto, J.C., editors. *Handbook of Ambient Intelligence and Smart Environments*. Springer; 2010. ISBN 978-0-387-93807-3.
2. Cook, D., Youngblood, M., Das, S.. A multi-agent approach to controlling a smart environment. In: *Designing Smart Homes*; vol. 4008 of *LNCIS*. Springer. ISBN 978-3-540-35994-4; 2006, p. 165–182.
3. de Silva, L., Dekker, A., Harland, J.. Planning with time limits in bdi agent programming languages. In: *Proceedings of CATS'07*; vol. 65. Darlinghurst, Australia: Australian Computer Society, Inc. ISBN 1-920-68246-5; 2007, p. 131–139.
4. Nygard, K.E., Xu, D., Pikalek, J., Lundell, M.. Multi-agent designs for ambient systems. In: *Proceedings of Ambi-Sys'08*. ICST. ISBN 978-963-9799-16-5; 2008, p. 10:1–10:6.
5. Chaouche, A.C., El Fallah Seghrouchni, A., Ilić, J.M., Saïdouni, D.E.. A Formal Approach for Contextual Planning Management: Application to Smart Campus Environment. In: Bazzan, A.L., Pichara, K., editors. *Advances in Artificial Intelligence - IBERAMIA 2014*; vol. 8864 of *Lecture Notes in Artificial Intelligence*. Springer International Publishing. ISBN 978-3-319-12026-3; 2014, p. 791–803.
6. Rao, A.S., Georgeff, M.. BDI agents: from theory to practice. In: *First International Conference on Multi-Agent Systems (ICMAS-95)*. S. Francisco, CA; 1995, p. 312–319.
7. Chen, G., Kotz, D.. A Survey of Context-Aware Mobile Computing Research. Tech. Rep. TR2000-381; Dartmouth College, Computer Science; Hanover, NH; 2000.
8. Henriksen, K., Indulska, J.. Developing context-aware pervasive computing applications: Models and approach. *Pervasive Mob Comput* 2006;**2**(1):37–64.
9. Guivarch, V., Camps, V., Péninou, A.. AMADEUS: an adaptive multi-agent system to learn a user's recurring actions in ambient systems. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* 2013;**1**(3).
10. Domnitcheva, S.. Location modeling: State of the art and challenges; 2000.
11. Hu, H., Lee, D.L.. Semantic location modeling for location navigation in mobile environment. In: *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on*. 2004, p. 52–61.
12. Boukharrou, R., Chaouche, A.C., Ilić, J.M., Saïdouni, D.E.. Contextual-Timed Planning Management for Ambient Systems. In: *26th IEEE International Conference on Tools with Artificial Intelligence*. IEEE Computer Society; 2014, .
13. Courtiat, J.P., Saïdouni, D.E.. Relating maximality-based semantics to action refinement in process algebras. In D Hogrefe and S Leue, editors, *IFIP TC6/WG61, Int Conf on FORTE'94* 1995::293–308.
14. Belala, N., Saïdouni, D.E., Boukharrou, R., Chaouche, A.C., Seraoui, A., Chachoua, A.. Time Petri Nets with Action Duration: A True Concurrency Real-Time Model. *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)* 2013;**4**(2):62–83.
15. Alur, R., Dill, D.L.. A theory of timed automata. *Theoretical Computer Science* 1994;**126**(2):183 – 235.
16. Baier, C., Katoen, J.P.. *Principles of Model Checking*. MIT Press; 2008. ISBN 978-0-262-02649-9.