

Contextual-Timed Planning Management for Ambient Systems

Radja Boukharrou, Ahmed Chawki Chaouche, Jean-Michel Ilié, Djamel Eddine Saïdouni

▶ **To cite this version:**

Radja Boukharrou, Ahmed Chawki Chaouche, Jean-Michel Ilié, Djamel Eddine Saïdouni. Contextual-Timed Planning Management for Ambient Systems. 26th IEEE International Conference on Tools with Artificial Intelligence, Nov 2014, Limassol, Cyprus. pp.107-114, 10.1109/ICTAI.2014.26 . hal-01228872

HAL Id: hal-01228872
https://hal.science/hal-01228872

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.

Contextual-Timed Planning Management for Ambient Systems

Radja Boukharrou*, Ahmed-Chawki Chaouche*§, Jean-Michel Ilié§ and Djamel Eddine Saïdouni*
*MISC Laboratory
University Constantine 2, Constantine, Algeria
§LIP6 Laboratory
University of Pierre and Marie Curie, Paris, France
Emails: boukharrou@misc-umc.org, ahmed.chaouche@lip6.fr,
jean-michel.ilie@lip6.fr and saidouni@misc-umc.org

***Abstract*— In this paper, we propose an algebraic language, called Time-AgLOTOS, which is dedicated to express BDI agent plans, according to the features and requirements of Ambient Intelligence (AmI) systems. Plan expressions are written and composed using Time-AgLOTOS, so that plans are built automatically and on the fly, as a system of concurrent processes. This language describes the time-dependent behavior of the agent and provides a theoretical foundation for performing planning under timing constraints, taking into account the duration of actions. Also, it allows to express behavioral capabilities such as communication, mobility and cooperation. In this context, we show how to achieve a powerful mechanism for a contextual guidance based on a specific and formal construction called Contextual Time Planning System (CTPS).**

***Keywords*-Real-time ambient system; time planning language and semantics; timing constraints; action duration; planning consistency.**

I. Introduction

Ambient intelligence (AmI) is an emerging technology area embedding heterogeneous electronic devices and often forming an open, dynamic and complex environment. Complexity here refers too to pro-actively, but sensibly, assisting people during their various daily activities. Actually, such systems have more and more impact over people's lifestyles [1], [2]. Usually, multi-agent systems (MAS) approaches offer interesting and powerful framework for the design of such complex systems, since their agents are considered as intelligent, autonomous, pro-active and flexible [3]. Several modeling approaches are proposed for potentially improving the software engineering of these systems and which partially focus on some of MAS aspects, e.g. [4], [5], [6]. In fact, the major problem for the system entities consists in recognizing environmental context, including location, the discovery of other agents, planning and handling information in a more *semantic* manner within *temporal* reasoning.

To verify/reason about MAS we need to capture the desired properties in the right way as well as modeling agent's abilities and functionalities. The Belief-Desire-Intention (BDI) agent model can flexibly handle the entire agent modeling and developing in an AmI system [5]. Indeed, agents are able, thanks to their mental attitudes, to use their Beliefs, Desires, and Intentions in order to select and execute a *plan* of *actions* i.e., to cope easily with the real world.

The Higher-order Agent Model (HoA), presented in [7], provides necessary theoretical foundations for designing/studding of BDI agents in AmI systems. On the other hand, it offers a useful framework, based on an *agent-centric* approach, to capture the behavior changes of the agent as consequence to its mental evolution according to the context-awareness ability. Therefore, At any instant of time the current state of an agent is expressed as an *HoA configuration* capturing the evolution of the agent in both its mental and planning states. This allows to maintain *state consistency* which yet appears to be absent from BDI theories.

In practice, in addition to the *Mental process*, the architecture of the agent is enriched within an efficient component, so-called *Planning process* ,with the aim to introduce more formal techniques for the description and the validation of ambient agent. Planning process is based on an algebraic language dedicated to the specification and the construction of a plan automatically from the intentions set of the agent and the library of elementary plans. The *AgLOTOS* [7] is a process algebra describing agent behaviors in a modular and concurrent manner. It takes into account the AmI aspects i.e. mobility and communication. This allows designing a powerful planning management. Indeed, the semantics of AgLOTOS automatically produces a *Contextual Planning System* (*CPS*), that is a state transition system which captures the evolution of the plan according to updated information, enhancing ways to guide the agent with respect to the predicted context changes.

Most research works for AmI systems assume that agents have an unlimited amount of time available to perform and execute plan, which is often not the case in many dynamic real world environments. In this context, the conditions are highly dynamic and requiring *real-time reasoning* capabilities. Moreover, within this complex environment, the agent (especially BDI agent) must make often *time-critical decisions* and should execute right actions in the right

time. So, within the absence of an appropriate mechanism considering the concepts of time, the production of good decisions would not be easy, if not impossible. In order to verify the quantitative timing properties, we argue the necessity of these two levels of time specification: *timing constraints* and *action duration* where real time performance is required.

The environment may place constraints on agents and their actions, like *time* issues, handling connectivity, localization and physical conditions (e.g. power/battery resources) which defined the key concepts of context. In [8], depending on the nature of the context, the context information is categorized in four types: *(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. Another aspect of context that is considered in some works is *activity* [9].

Although, more importantly, when the computing, user and physical contexts are recorded across *a time span*, this gives rise to a context history [8]. This type of context is important for *predicting* future values of context and developing appropriate models/planning for context evolution. It can be useful for establish trends and guidance service for the agent. In this paper, our presented work deals with the main types of context: spatial context, temporal context, social context (agent nearby) and the context history.

To address the design-time issue for more complete and correct specifications of real-time AmI systems, we propose at first an extension of AgLOTOS language, called *Time-AgLOTOS*, describing the time-dependent behavior of the agent. Time-AgLOTOS provides a theoretical foundation for performing planning under timing constraints taking into account the duration of actions.

Other aim of this paper is to investigate true concurrency semantics instead of interleaving ones for two reasons: *(1)* to support structural and temporal non-atomicity of actions, i.e., actions may be divisible and of non-null duration. To escape from these assumptions, most of works consider actions as two instantaneous events: their start and their completion, in addition to the wait between these events. Although this approach seems to be attractive, it may contribute toward graph explosion in state-transition models. *(2)* to capture and express concurrent and parallel behaviors in a more natural and intuitive way, i.e. to distinguish between sequential and parallel runs of actions which appears interesting for the design/reasoning about the agent ability to pursue multiple desires and intentions concurrently.

Then, we give true-concurrency semantics to Time-AgLOTOS in terms of maximality semantics [10], [11], [12]. The underlying semantics is a Contextual Time Planning System (CTPS) which is very close to Durational Action Timed Automata model [13]. As a main result, CTPS allows the verification of new properties related to simultaneous progress of actions at different states of the system. Moreover, based on the possible evolution specified in a plan, we aim at offering a predictive service helping the agent on the fly to take time-critical decisions and to guide it to achieve its adopted desires concurrently.

II. Ambient Algebraic Language for Plan Specification

The *Higher-order Agent Model* (*HoA*) [7] provides a formal modeling for BDI agent, expressing reasoning mechanism and planning process explicitly. The building of the agent plan is described by using a process-algebra-based language, called *AgLOTOS* [7]. Focusing on Intention based reasoning, the agent plan is structured to explicit the parts of the plan associated with each intention selected by the agent.

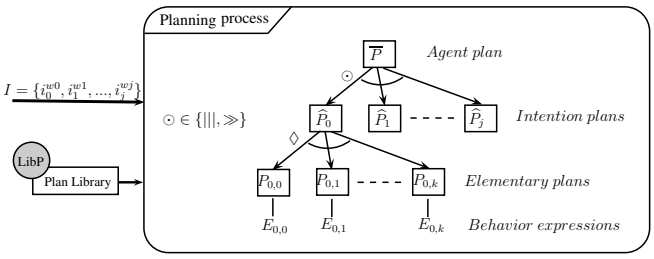


Figure 1. Planning Process

As highlighted in Figure 1, The planning structure is composed of three levels of plans: elementary plans, intention plans and finally agent plan. The *agent plan* is obtained by sequential and/or a parallel composition of *intentions plans* where each one is dedicated to achieve its corresponding intention. Moreover, an Intention plan is viewed as an alternate of several *elementary plans* to express the possibility to achieve the intention. In the HoA architecture, elementary plans are assumed to be already specified so can be extracted from the so-callled *LibP* library. This library is assumed to be indexed by the set of the possible intentions that the agent can commit.

As a process algebraic language, AgLOTOS provides a way to describe plans in terms of algebraic process expressions, composed in modular and concurrent ways. This language uses actions as its primitives and some of them have a special meaning e.g., mobility and communication. Actions can then be combined to describe elementary plans using a variety of defined operators. Over operators are used *(1)* to form the alternation of elementary plans and *(2)* to compose the intention plans leading to the agent plan. AgLOTOS has an operational semantics that takes the planning structure into account to offer a clear separation of the different executions related to the intention plans.

Let us now detail the highlights of the AgLOTOS-based specifications for plans.

Agent plan level: The set of intention plans can be globally handled by using the concurrent $|||$ or sequential $\gg$ AgLOTOS composition between intention plans, leading to the specification of an Agent plan. Let $\overline{\mathcal{P}}$ be the set of names qualifying the possible agent plans and $\widehat{\mathcal{P}} \in \overline{\mathcal{P}}$ be the set of names used to identify the possible intention plans with $\widehat{P} \in \widehat{\mathcal{P}}$, such that $\overline{P}$ is any agent plan defined by:

$$\overline{P} \ ::= \ \widehat{P} \ \mid \ \overline{P} \ ||| \ \overline{P} \ \mid \ \overline{P} \gg \overline{P}$$

Intention plan level: An intention plan can be expressed by an alternate of several elementary plans, such that each one can be launched to solve the corresponding intention. This is captured in AgLOTOS by using the alternate composition operator $\lozenge$ which allows to specify an alternate of elementary plans. In particular, an intention is satisfied iff at least one of the associated elementary plans is successfully terminated. Formally, we define an intention plan $\widehat{P}$ as:

$$\widehat{P} \ ::= \ P \ \mid \ \widehat{P} \lozenge \widehat{P}$$

Elementory plan level: Elementary plans are descried by *AgLOTOS expressions*, referring to a finite set of observable actions. Any AgLOTOS expression is associated with contextual information relating to the (current) BDI state of an agent. For that, let Θ be a finite set of space locations where an agent can move and Λ be the set of agents with which it can communicate. Let $\mathcal{O}$ be the (finite) set of observable actions which are viewed as instantiated predicates, ranged over by $a, b, ...$ and let L be any subset of $\mathcal{O}$. $H \subset \mathcal{O}$ is the set of the so-called AmI primitives which represent the mobility and communication. The agent mobility is expressed by the primitive $move(\ell)$ which is used to handle the agent move to some location ℓ ($\ell \in \Theta$). The syntax of the communication primitives is defined by the expression $\chi!(\nu)$ which specifies the emission to the agent χ ($\chi \in \Lambda$) of some message ν and the expression $\chi?(\nu)$ means that the message ν is received from some agent χ.

Let $Act = \mathcal{O} \cup \{\tau, \delta\}$, be the set of actions, where τ is the internal action and δ is a particular observable action which features the successful termination of a plan. The AgLOTOS language specifies pairs for each elementary plan composed of a name to identify it and an AgLOTOS expression to feature its behavior. Consider that elementary plan's names are ranged over $P_1, P_2, ...$ and that the set of all the possible behavior expressions is denoted $\mathcal{B}$, ranged over $E_1, E_2, ...$ The AgLOTOS expressions are written by composing actions through LOTOS operators. The syntax of an AgLOTOS elementary plan P is defined inductively as follows:

$$\begin{array}{lcll} P & ::= & E & \\ E & ::= & exit \mid stop & \\ & & \mid a; E \mid E \odot E \mid \tau; E & (a \in \mathcal{O}) \\ & & \mid hide \ L \ in \ E & \\ H & ::= & \mid move(\ell) & (H \subset \mathcal{O}, \ell \in \Theta) \\ & & \mid \chi!(\nu) \mid \chi?(\nu) & (\chi \in \Lambda) \\ \odot & = & \{ \ |||, |[L]|, ||, [\ ], \gg, [> \ \} & \end{array}$$

The elementary expression $stop$ specifies a plan behavior without possible evolution and $exit$ represents the successful termination of some plan. In the syntax, the set $\odot$ represents the standard LOTOS operators: $E[\ ]E$ specifies a non-deterministic choice, $hide \ L \ in \ E$ a hiding of the actions of L that appear in E, $E \gg E$ a sequential composition and $E [> E$ the interruption. The LOTOS parallel composition, denoted $E \,|[L]|\, E$, can model both synchronous composition, $E \,||\, E$ if $L = \mathcal{O}$, and asynchronous composition, $E \,|||\, E$ if $L = \emptyset$.

To be pragmatic considering any BDI state of the agent, we propose that the agent can label the different elements of the set I of intentions[1] by using a weight function $W : I \longrightarrow \mathbb{N}$. The ones having the same weight are composed by using the concurrent parallel operator $|||$. In contrast, the intention plans corresponding to distinct weights are ordered by using the sequential operator $\gg$. For instance, let $I = \{i_0^1, i_1^2, i_2^1, i_3^0\}$ be the considered set of intentions, such that the superscript information denotes a weight value, and let $\widehat{P_0}, \widehat{P_1}, \widehat{P_2}, \widehat{P_3}$ be their corresponding intention plans, the constructed agent plan could be viewed as: $\overline{P} = \widehat{P_1} \gg (\widehat{P_0} ||| \widehat{P_2}) \gg \widehat{P_3}$.

The *AgLOTOS* operational semantics is basically derived from the one of LOTOS and can be found in [7].

A Simple AmI Example: Let Alice and Bob be two agents of an AmI University system. Such a system is clearly dynamic and open since agents can enter and leave at any moment. Furthermore, the agents are context-aware; they can perceive the enter and the location of each agent within the system and they are able to communicate through this system. Let $\ell_1, \ell_2 \in \Theta$ be two locations of the studied system where the two agents can behave. The proposed problem of Alice is that she wants to realize the two tasks at the same time: *(a)* meet Bob in ℓ_1, and *(b)* get her exam copies from ℓ_2. Clearly, the Alice's desires are inconsistent since Alice cannot be in two distinct locations simultaneously. At the considered moment, Alice is in ℓ_1 and she expresses the two inconsistent desires, whereas Bob is already arrived in ℓ_2 and has expressed the desire to work with Alice. Because of the conflictual desires, the current intention of Alice is only to meet Bob first. Consequently, the initial HoA configurations

[1]We assume that the BDI agent itself can solve conflicting situations that could arise between intentions for some context, by means of a scheduling process applied on the set of intentions.

of Alice and Bob are respectively q_0^A and q_0^B in Table I. The actions in plans are simply expressed in AgLOTOS language by using instantiated predicates, like $get_copies(\ell_2)$. Note

Table I
INITIAL HOA CONFIGURATIONS FOR ALICE AND BOB

	Alice's scenario
q_0^A	$B_0 = \{in(me, \ell_1), in(copies, \ell_2)\}$ $D_0 = \{meeting(Bob, \ell_1), getting_copies(\ell_2)\}$ $I_0 = \{meeting(Bob, \ell_1)\}$ $\overline{P_0} = meet(Bob, \ell_1); exit$
	Bob's scenario
q_0^B	$B_0 = \{in(me, \ell_2)\}$ $D_0 = \{meeting(Alice, \ell_1)\}$ $I_0 = \{meeting(Alice, \ell_1)\}$ $\overline{P_0} = move(\ell_1); meet(Bob, \ell_1); exit$

that Alice and Bob are specified separately and that they are able to coordinate for improving the satisfaction of more desires (see Section IV).

III. REAL-TIME AMBIENT ALGEBRAIC LANGUAGE

Another key requirement of AmI environment is real time reasoning capabilities, when timing constraints become increasingly important to the proper functioning of AmI agent.

By way of example, let us consider the example of Table I but in the context of *timing* dependencies of actions. For instance, Alice is already in ℓ_1, she wants to meet Bob in this location as soon as possible because she has a limited time for going to get her exam copies e.g just a delay of 5 time units. Also, she has only 2 time units for that meeting. Then, she will take her exam copies from ℓ_2 in the required time. On the other hand, Bob wants to terminate his work in the current location before leaving it to ℓ_1. This takes a certain amount of time (at least 1 time unit). In this case both agents should respond in a *timely* manner to changes that occur in environment and to its timing constraints.

According to this example, we can define different types of real-time constraints. Hence, These constraints can limit the enabling domains regardless of the duration of other actions by *delaying* for example an action of a certain quantity of time before it will be feasible; or by limiting the time during which an action is offered to its environment (*temporal restriction*). In this paper,we define an algebraic language called Time-AgLOTOS to specify the syntax and the semantics of an agent plan considering the classical timing operators similar to those introduced in temporal extensions of LOTOS, such as D-LOTOS [13] and RT-LOTOS [14] for the description of such constraints.

The delaying aspect can be specified by means of a new operator $\Delta^d E$ that describes time delays. This operator means that no evolution of E is allowed before the end of a delay equal to d. For example, the specification of Bob's plan for achieving its intention $meeting(Alice, \ell_1)$ is described by the use of the delaying operator as: $\Delta^1(move(\ell_1); meet(Alice, \ell_1); exit)$. We already know how to describe syntactically the sequencing of action occurrences by means of the action-prefixing operator of AgLOTOS. We can extend this operator to deal with the notion of temporal constraint for an action through two new prefixing operators $a\{d\}; E$ and $a@t[SP]; E$. The temporal restriction for the occurrence of an action can be expressed using the operator $a\{d\}$ which means that action a is offered during at most d time units. Whereas, in the operator $a@t[SP]; E$ (resp. $\tau@t\{d\}; E$), SP is a selection predicate that constrains the occurrence of action a (resp. τ) and t is a temporal variable recording the time taken after the enabling of action a (resp. τ). For instance in Alice's scenario, its ongoing desires are depending strongly on time limit which can be expressed by: $meet(Bob, \ell_1)@t_1[t_1 \leqslant 2]; exit\{0\} \gg move(\ell_2)@t_2[t_2 \leqslant 5 - t_1]; get_copies(\ell_2); exit\{1\}$.

Moreover, we should reason on the internal action τ as a specific action that cannot be delayed and must be performed as soon as it becomes available because it refers to the local behavior of the agent. This introduces the notion of *urgency* when the action must occur immediately and while time progression is stopped. This notion is useful for describing features like *time-outs* and *time deadlocks*. Another question arises addressing the reasoning about time in quantitative sense. In this context, we argue that the action duration must be explicitly taken into account by the Time-AgLOTOS. In what follows, we show how to explicitly introduce action duration in AgLOTOS. This is became possible by the use of a function that gives to each action a non-null duration. The idea is based on the principle of the maximality semantics allowing the expression of the structural and temporal non-atomicity of actions in a similar way introduced for D-LOTOS.

Syntax: We now define *Time-AgLOTOS* in which both timing constraints and actions duration are considered. Absolutely, as far AgLOTOS, the language allows the specification of dynamism and context-aware of AmI system. The syntax of a Time-AgLOTOS agent plan directly relates to the building of a behavior expression representing the whole agent behavior according to the construction approach discussed above in Section II. Indeed, we propose a temporal extension to the syntax of elementary plan taking into account time aspects and preserving the overall structure of the agent plan. The syntax of an Time-AgLOTOS elementary plan is defined inductively as follows:

$$
\begin{array}{lcl}
P & ::= & E \\
E & ::= & exit\{d\} \mid stop \\
 & & \mid \Delta^d E \mid E \odot E \\
 & & \mid a@t[SP]; E \mid \tau@t\{d\}; E \quad (a \in \mathcal{O}) \\
 & & \mid hide\ L\ in\ E \\
\odot & = & \{\ |||, |[L]|, ||, [\ ], \gg, [>\ \}
\end{array}
$$

Let $\mathcal{D}$ be a domain of time like $\mathbb{Q}^+$ or $\mathbb{R}^+$. $\Upsilon : Act \to \mathcal{D}$ is the duration function which associates to each action its duration. Let a be an action, E a behavior expression and $d \in \mathcal{D}$ a value in the temporal domain. We recall that $a\{d\}$ expresses the temporal restriction, $\Delta^d E$ represents a time delay and $a@t[SP]; E$ and $\tau@t\{d\}; E$ describe timing dependencies. Thereby, the elementary expression $stop$ specifies a plan 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 the $stop$ and $exit$ are of null duration. We denote that at both agent plan and intention plan levels, only the order of plan executions is expressed within time specification abstraction (as in untimed case). The specification of time-dependency is allowed explicitly at elementary plans.

IV. Time-AgLOTOS Semantics

As far as *Time-AgLOTOS* is concerned, the operational semantics of elementary plans is given by rules of Definition 4.5. This semantics is basically derived from the one of D-LOTOS [13] and which captures the evolution of a concurrent processes. A pair (E, P) represents an *elementary plan configuration*, viewed as a process and identified by P such that its behavior expression is E. A transition relation between configurations, embodying the behavior expressions of elementary plans, is expressed in the context of maximality semantics and which gives as a result a well-structured state transition system. In *Time-AgLOTOS*, observable actions are not urgent as their occurrences depend on the willingness of the environment. Whereas, for the internal action τ, the situation is more complex as one may want to express *urgent* occurrence of this action.

A. Formalization

In the following, $\mathcal{H}$ is a set of clocks with non-negative values (within a time domain $\mathbb{T}$, like $\mathbb{Q}^+$ or $\mathbb{R}^+$). The set $\Phi_t(\mathcal{H})$ of temporal constraints γ over $\mathcal{H}$ is defined by $\gamma ::= x \sim t$ where x is a clock in $\mathcal{H}$, $\sim \in \{=, <, >, \leq, \geq\}$ and $t \in \mathbb{T}$. F_x will be used to indicate a constraint of the form $x \sim t$. A *valuation* (or *interpretation*) v for $\mathcal{H}$ is a function which associates to each $x \in \mathcal{H}$ a value in $\mathbb{T}$. A valuation v for $\mathcal{H}$ satisfies a temporal constraint γ over $\mathcal{H}$ iff γ is true by using clock values given by v. For $\mathcal{U} \subseteq \mathcal{H}$, $[\mathcal{U} \to 0]v$ indicates the valuation for $\mathcal{H}$ which assigns value 0 to each $x \in \mathcal{U}$, and agrees with v over the other clocks of $\mathcal{H}$. The set of all valuations for $\mathcal{H}$ is noted $\Xi(\mathcal{H})$. 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 $v \in \Xi(\mathcal{H})$. The set of all the possible elementary configurations is denoted $\mathcal{C}$ and given by Definition 4.1.

Definition 4.1: The set $\mathcal{C}$ of elementary configurations is defined by:

- $\forall E \in \mathcal{B}, \forall F \in 2^{\Phi_t(\mathcal{H})} : {}_F[E] \in \mathcal{C}$
- $\forall P \in \mathcal{P}, \forall F \in 2^{\Phi_t(\mathcal{H})} : {}_F[P] \in \mathcal{C}$
- if $\mathcal{E} \in \mathcal{C}$ then hide L in $\mathcal{E} \in \mathcal{C}$
- if $\mathcal{E} \in \mathcal{C}$ and $F \in \mathcal{B}$ then $\mathcal{E} \gg F \in \mathcal{C}$
- if $\mathcal{E}, \mathcal{F} \in \mathcal{C}$ then $\mathcal{E}\, op\, \mathcal{F} \in \mathcal{C}$ with $op \in \{\ [\,], |||, ||, |[L]|, [>\ \}$
- $\forall \mathcal{E} \in \mathcal{C}, \forall d \in \mathcal{D} : \triangle^d \mathcal{E} \in \mathcal{C}$
- if $\mathcal{E} \in \mathcal{C}$ and $\{a_1, ..., a_n\},\ \{b_1, ..., b_n\} \in 2^{\mathcal{O}}$ then $\mathcal{E}\,[b_1/a_1, ..., b_n/a_n] \in \mathcal{C}$

In order to determine a set of duration conditions w.r.t. some configuration $\mathcal{C}$, we use the function $\psi : \mathcal{C} \to 2^{\Phi_t(\mathcal{H})}$ such that for each configuration of the form ${}_F[E] \in \mathcal{C}$ we have $\psi({}_F[E]) = F$.

Definition 4.2: The function $\psi_{\mathcal{H}} : 2^{\Phi_t(\mathcal{H})} \to 2^{\mathcal{H}}$, determines the set of used clocks in a duration conditions set and is defined recursively by:

$$\begin{aligned}
&\psi_{\mathcal{H}}(\emptyset) = \emptyset \\
&\psi_{\mathcal{H}}(\{x \sim t\}) = \{x\} \\
&\psi_{\mathcal{H}}(F_1 \cup F_2) = \psi_{\mathcal{H}}(F_1) \cup \psi_{\mathcal{H}}(F_2)
\end{aligned}$$

such as $F_1, F_2 \in 2^{\Phi_t(\mathcal{H})}$, $x \in \mathcal{H}$, $\sim \in \{=, <, >, \leq, \geq\}$ and $t \in \mathbb{R}^+$.

If ${}_F[E]$ is an elementary configuration then ${}_F[E]\backslash K$ indicates the new configuration obtained by removing the set K of timing constraints from the set F s.t. the concerning clocks are used in F. ${}_F[E]\backslash \mathrm{K}={}_{F\backslash K}[E]$) with $F\backslash K$ given by the Definition 4.3.

Definition 4.3: Let F be a set of ending conditions; $F\backslash K$ indicates the set obtained by removing (starting from F) all the duration conditions which are related to the clocks of K from F. $F\backslash K$ is recursively defined on F as follows:

$$\begin{aligned}
&\emptyset\backslash K = \emptyset \\
&(F_1 \cup F_2)\backslash K = F_1\backslash K \cup F_2\backslash K \\
&\{x \sim t\}\backslash K = \begin{cases} \emptyset & \text{if } x \in \psi_{\mathcal{H}}(K) \\ \{x \sim t\} & \text{otherwise} \end{cases}
\end{aligned}$$

such that $F_1, F_2 \in 2^{\Phi_t(\mathcal{H})}$, $K \in 2^{\Phi_t(\mathcal{H})}$, $x \in \mathcal{H}$, $\sim \in \{=, <, >, \leq, \geq\}$ and $t \in \mathbb{R}^+$.

The specification of delays and timing constraints in the generation process requires the following Definition.

Definition 4.4: Let $G \in 2^{\Phi_t(\mathcal{H})}$ be a timing constraint and t be time units:

- The function $shift(G, t)$ dealing with the move of a temporal domain G by a time units t is defined by
  $shift(G, t) \stackrel{def}{=} \{(\min + t \sim x \sim \max + t) \mid (\min \sim x \sim \max) \in G\}$ *and* $\sim \in \{<, \leq\}$
- The function $maxval(G)$, which restricts a temporal domain G on its upper bound, is defined as
  $maxval(G) = \vee_{\{min \sim x \sim max\} \in G}\ (x = max)$ with $\sim \in \{<, \leqslant\}$

Now, we introduce the operational semantics for Time-AgLOTOS elementary plan as rules of Definition 4.5.

Definition 4.5: The maximality transition relation $\longrightarrow \subseteq \mathcal{C} \times 2^{\Phi_t(\mathcal{H})} \times 2^{\Phi_t(\mathcal{H})} \times Act \times \mathcal{H} \times \mathcal{C}$ is defined as being the smallest relation satisfying the following rules:

- $$\underset{x=get(\mathcal{M})}{{}_{\emptyset}[exit\{u\}}} \xrightarrow{G=\{c_{\emptyset} \leq u\}, D=\{\mathtt{false}\}, \delta, x} {}_{\{x \geq 0\}}[stop]$$
- $$\underset{x=get(\mathcal{M})}{{}_{F}[exit\{u\}}} \xrightarrow[D=\{\mathtt{false}\}, \delta, x]{G=\{\cup_{h \in \mathcal{H}}\{shift(0 \leq h \leq u, t)\} \backslash F_h = \{h \geq t\}\}} {}_{\{x \geq 0\}}[stop]$$

- $$\frac{}{{}_{\emptyset}[a\{u\};E] \xrightarrow[x=get(\mathcal{M})]{G=\{c_\emptyset \le u\},D=\{\texttt{false}\},a,x} {}_{\{x\ge\tau(a)\}}[E]}$$
- $$\frac{}{{}_{F}[a\{u\};E] \xrightarrow[x=get(\mathcal{M})]{G=\{\cup_{h\in\mathcal{H}}\{shift(0\le h\le u,t)\}\setminus F_h=\{h\ge t\}\}}{D=\{\texttt{false}\},a,x} {}_{\{x\ge\tau(a)\}}[E]}$$
- $$\frac{}{{}_{\emptyset}[h\{u\};E] \xrightarrow[x=get(\mathcal{M})]{G=\{c_\emptyset \le u\},D:=G,h,x} {}_{\{x\ge 0\}}[E]}$$
- $$\frac{}{{}_{F}[h\{u\};E] \xrightarrow[x=get(\mathcal{M})]{G=\{\cup_{h\in\mathcal{H}}\{shift(0\le h\le u,t)\}\setminus F_h=\{h\ge t\}\}}{D:=G,h,x} {}_{\{x\ge 0\}}[E]}$$
- $$\frac{SP=\{\min\sim t\sim\max\} \qquad \sim\in\{<,\le\}}{{}_{\emptyset}[a@t[SP];E] \xrightarrow[x=get(\mathcal{M})]{G=\{[c_\emptyset/t]SP\},D=\{\texttt{false}\},a,x} {}_{\{x\ge\tau(a)\}}[E]}$$
- $$\frac{SP=\{\min\sim t\sim\max\} \qquad \sim\in\{<,\le\}}{{}_{F}[a@t[SP];E] \xrightarrow[x=get(\mathcal{M})]{G=\{\cup_{h\in\mathcal{H}}\{[h/t]shift(SP,d)\}\setminus F_h=\{h\ge d\}\}}{D=\{\texttt{false}\},a,x} {}_{\{x\ge\tau(a)\}}[E]}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}'}{\mathcal{F}\,[\,]\,\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad \mathcal{E}\,[\,]\,\mathcal{F} \xrightarrow{G,D,a,x} \mathcal{E}'}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad a\notin L\cup\{\delta\}}{\mathcal{E}\,|[L]|\,\mathcal{F} \xrightarrow[y=get(\mathcal{M}-(\psi_{\mathcal{H}}(\psi(\mathcal{E})\cup\psi(\mathcal{F}))-\psi_{\mathcal{H}}(G)))}{G,D,a,y} \mathcal{E}'[y/x]\,|[L]|\,\mathcal{F}\setminus G}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad a\notin L\cup\{\delta\}}{\mathcal{F}\,|[L]|\,\mathcal{E} \xrightarrow[y=get(\mathcal{M}-(\psi_{\mathcal{H}}(\psi(\mathcal{E})\cup\psi(\mathcal{F}))-\psi_{\mathcal{H}}(G)))}{G,D,a,y} \mathcal{F}\setminus G\,|[L]|\,\mathcal{E}'[y/x]}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad \mathcal{F} \xrightarrow{G',D',a,y} \mathcal{F}' \qquad a\in L\cup\{\delta\}}{\mathcal{E}\,|[L]|\,\mathcal{F} \xrightarrow[z=get(\mathcal{M}-(\psi_{\mathcal{H}}(\psi(\mathcal{E})\cup\psi(\mathcal{F}))-(\psi_{\mathcal{H}}(G)\cup\psi_{\mathcal{H}}(G'))))}{G\cup G',D\cup D',a,z} \mathcal{E}'[z/x]\setminus G'\,|[L]|\,\mathcal{F}'[z/y]\setminus G}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad a\notin L}{hide\ L\ in\ \mathcal{E} \xrightarrow{G,D,a,x} hide\ L\ in\ \mathcal{E}'}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad a\in L}{hide\ L\ in\ \mathcal{E} \xrightarrow{G,D:=G,h,x} hide\ L\ in\ \mathcal{E}'}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad d>0}{\Delta^d\mathcal{E} \xrightarrow{shift(G,d),shift(D,d),a,x} \mathcal{E}'}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad a\neq\delta}{\mathcal{E}\gg F \xrightarrow{G,D,a,x} \mathcal{E}'\gg F}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,\delta,x} \mathcal{E}'}{\mathcal{E}\gg E \xrightarrow{G,D,h,x} {}_{\{x\ge 0\}}[E]}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad a\neq\delta}{\mathcal{E}\,[>\,\mathcal{F} \xrightarrow[y=get(\mathcal{M}-(\psi_{\mathcal{H}}(\psi(\mathcal{E})\cup\psi(\mathcal{F}))-\psi_{\mathcal{H}}(G)))}{G,D,a,y} \mathcal{E}'[y/x]\,[>\,\mathcal{F}\setminus G}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,\delta,x} \mathcal{E}' \qquad \forall z\in\psi(\mathcal{F})}{\mathcal{E}\,[>\,\mathcal{F} \xrightarrow[y=get(\mathcal{M}-(\psi_{\mathcal{H}}(\psi(\mathcal{E})\cup\psi(\mathcal{F}))-\psi_{\mathcal{H}}(G)))}{G,D,\delta,y} \mathcal{E}'[y/x]\,[>\,{}_{\psi(\mathcal{F})-G_z}[stop]}$$
- $$\frac{\mathcal{F} \xrightarrow{G,D,a,x} \mathcal{F}' \qquad \forall z\in\psi(\mathcal{E})}{\mathcal{E}\,[>\,\mathcal{F} \xrightarrow[y=get(\mathcal{M}-(\psi_{\mathcal{H}}(\psi(\mathcal{E})\cup\psi(\mathcal{F}))-\psi_{\mathcal{H}}(G)))}{G,D,a,y} {}_{\psi(\mathcal{E})-G_z}[stop]\,[>\,\mathcal{F}'[y/x]}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad a\notin\{a_1,\ldots,a_n\}}{\mathcal{E}[b_1/a_1,\ldots,b_n/a_n] \xrightarrow{G,D,a,x} \mathcal{E}'[b_1/a_1,\ldots,b_n/a_n]}$$
- $$\frac{\mathcal{E} \xrightarrow{G,D,a,x} \mathcal{E}' \qquad a=a_i\ (1\le h\le n)}{\mathcal{E}[b_1/a_1,\ldots,b_n/a_n] \xrightarrow{G,D,b_i,x} \mathcal{E}'[b_1/a_1,\ldots,b_n/a_n]}$$
- $$\frac{P::=E \qquad {}_{F}[E] \xrightarrow{G,D,a,x} \mathcal{F}}{{}_{F}[P] \xrightarrow{G,D,a,x} \mathcal{F}}$$

Such as, $c_\emptyset$ is a particular clock which is created and initialized at the enabling time of the agent system.

Let us consider now another of level of plan and its semantics. The behavior expression of the agent plan $\overline{P}$ under evolution is called *agent plan configuration* and denoted $[\overline{P}]$. Definition 4.6 specifies how an agent plan configuration is formed composition manner from the *intention plan configurations* like $(E, \widehat{P})$, themselves built from an alternate of *elementary plan configurations* as $(\Diamond^{k=1..n} E_k, P)$.

Definition 4.6: Any Agent plan configuration $[\overline{P}]$ has a canonical representation defined by the following two rules:

1) $$\frac{\overline{P}::=\widehat{P} \qquad \widehat{P}::=\Diamond^{k=1..n} P_k \qquad P_k::=E_k}{[\overline{P}]::=(\Diamond^{k=1..n} E_k,\ \widehat{P})}$$

2) $$\frac{\overline{P}::=\overline{P_1}\ \odot\ \overline{P_2} \qquad \odot\in\{|||,\gg\}}{[\overline{P}]::=[\overline{P_1}]\ \odot\ [\overline{P_2}]}$$

Notice that the *contextual planning state* takes into account the agent location and the termination information concerning the different intention plans already achieved; its definition is given as follows:

Definition 4.7: A *contextual planning state* is a tuple (ps, ℓ, T), where ps is any plan configuration $[\overline{P}]$, ℓ corresponds to an expected location for the agent, and T is the subset of intention plans which are terminated.

Table II shows the operational semantic rules defining the so-called *contextual planning state* changes for the agent. These rules are applied to produce a *Contextual Time Planning System* (*CTPS*). Thereby, the initial contextual planning state $(ps_0, \ell, \emptyset)$ means that the agent is initially at location ℓ, and its plan configuration is $ps_0 = [\overline{P}]$ with no specific plan termination.

B. Contextual Planning Construction

From any HoA configuration q, we are now able to build a transition system representing the agent plan and its state changes. Further, from the set of intentions represented in q, denoted $I(q)$, the CTPS is built using the former rules defined above. In addition to the timing constraints and the maximality information (ending condition and concurrent execution) of actions, $CTPS$ takes into account three kinds of information represent the contextual ones as : *(a)* the reached location in a given planning state, *(b)* the set of intention plans that are terminated when reaching this planning state, and *(c)*, more globally, the set $\Lambda(q)$ of neighbors currently known by the agent. Considering $q = (bdi, ps)$ any HoA configuration of the agent, its CTPS in q, starting from ps, is given by the following Definition.

Definition 4.8: A Contextual Time Planning System is a tuple $\langle S, s_0, L_s, Tr, \mathcal{L}, \mathcal{T}\rangle$ of the support Act where :

- S is a finite set of contextual planning states s.t $s = (ps, \ell, T) \in S$,
- $s_0 = (ps, \ell, \emptyset) \in S$ is the initial contextual planning state of the agent, such that $ps = [\overline{P}]$ and ℓ represents the current location of the agent,
- L_s: $S \to 2^{\Phi_t(\mathcal{H})}$ is a function which corresponds to each state s the set F of ending conditions (duration conditions) of actions possibly in execution in s,
- $Tr \subseteq S \times 2^{\Phi_t(\mathcal{H})} \times 2^{\Phi_t(\mathcal{H})} \times Act \times \mathcal{H} \times S$ is the set of transitions of the form (s, G, D, a, x, s')[2],
- $\mathcal{L} : S \to \Theta$ is the location labeling function,
- $\mathcal{T} : S \to 2^{\widehat{P}}$ is the termination labeling function which captures the terminated intention plans.

A transition (s, G, D, a, x, s') represents a switch from the state s to the state s' by starting execution of action a and resetting clock x. G is the corresponding guard which must be satisfied to fire this transition. D is the corresponding deadline which requires, at the moment of its satisfaction, that action a must occur. (s, G, D, a, x, s') can be written $s \xrightarrow{G,D,a,x} s'$.

[2]We assume that $D \Rightarrow G$ for each transition.

Table II
SEMANTIC RULES OF BOTH INTENTION AND AGENT CONFIGURATIONS

Intention plan level		
(Action)	$\dfrac{E \xrightarrow{G,D,a,x} E' \quad a \in \mathcal{O} \cup \{\tau\}}{(E,\widehat{P}) \xrightarrow{G,D,a,x} (E',\widehat{P})}$	$\dfrac{E \xrightarrow{G,D,\delta,x} E'}{(E,\widehat{P}) \xrightarrow[\widehat{P}]{G,D,\tau,x} (E',\widehat{P})}$

Agent plan level		
(Action)	$\dfrac{ps \xrightarrow{G,D,a,x} ps' \quad a \in \mathcal{G} \cup \{\tau\}}{(ps,\ell,T) \xrightarrow{G,D,a,x} (ps',\ell,T)}$	$\dfrac{ps \xrightarrow[\widehat{P}]{G,D,\tau,x} ps'}{(ps,\ell,T) \xrightarrow{G,D,\tau,x} (ps',\ell,T \cup \{\widehat{P}\})}$
(Communication)	$\dfrac{ps \xrightarrow{G,D,\chi!(\nu),x} ps' \quad \chi \in \Lambda}{(ps,\ell,T) \xrightarrow{G,D,\chi!(\nu),x} (ps',\ell,T)}$	$\dfrac{ps \xrightarrow{G,D,\chi?(\nu),x} ps \quad \chi \in \Lambda}{(ps,\ell,T) \xrightarrow{G,D,\chi?(\nu),x} (ps',\ell,T)}$
(Mobility)	$\dfrac{ps \xrightarrow{G,D,move(\ell'),x} ps'}{(ps,\ell,T) \xrightarrow{G,D,move(\ell'),x} (ps',\ell',T)}$	
(Sequence)	$\dfrac{ps_1 \xrightarrow{G,D,a,x} ps_1' \quad a \in \mathcal{O} \cup \{\tau\}}{(ps_1 \gg ps_2,\ell,T) \xrightarrow{G,D,a,x} (ps_1' \gg ps_2,\ell,T)}$	$\dfrac{ps_1 \xrightarrow[\widehat{P}]{G,D,\tau,x} ps_1'}{(ps_1 \gg ps_2,\ell,T) \xrightarrow{G,D,\tau,x} (ps_2,\ell,T \cup \{\widehat{P}\})}$
(Parallel)	$\dfrac{ps_1 \xrightarrow{G,D,a,x} ps_1' \quad a \in \mathcal{O} \cup \{\tau\}}{(ps_1 \vert\vert\vert ps_2,\ell,T) \xrightarrow{G,D,a,x} (ps_1' \vert\vert\vert ps_2,\ell,T)}$	$\dfrac{ps_1 \xrightarrow{G,D,a,x} ps_1' \quad a \in \mathcal{O} \cup \{\tau\}}{(ps_2 \vert\vert\vert ps_1,\ell,T) \xrightarrow{G,D,a,x} (ps_2 \vert\vert\vert ps_1',\ell,T)}$
	$\dfrac{ps_1 \xrightarrow[\widehat{P_1}]{G,D,\tau,x} ps_1' \quad ps_2 \xrightarrow[\widehat{P_2}]{G',D',\tau,y} ps_2'}{(ps_1 \vert\vert\vert ps_2,\ell,T) \xrightarrow{G \cup G',D \cup D',\tau,z} (ps_1' \vert\vert\vert ps_2',\ell,T \cup \{\widehat{P_1},\widehat{P_2}\})}$	

As usual in real-time formal model, like [13], the semantics of a CTPS is defined as an infinite timed transition system Σ_A of the same alphabet Act.

Definition 4.9: The semantics of a Contextual Time Planning System $A = \langle S, s_0, L_s, Tr, \mathcal{L}, \mathcal{T} \rangle$ associates to the CTPS an infinite transition system Σ_A defined over $Act \cup \mathbb{R}^+$. A state of Σ_A is a pair $\langle s, v \rangle$ such that s is a state of A and v a valuation for the clocks of $\mathcal{H}$. A state $\langle s_0, v_0 \rangle$ is initial if s_0 is the initial state of A and $\forall x \in \mathcal{H}, v_0(x) = 0$. Two types of transitions are specified in-between the configurations of Σ_A. The time passing transitions are specified according to the rules $RA1$ and $RA2$ whereas the action transition, which refers to the launching of transition from A, is related to the rule RD.

$$(RA1)\ \frac{d \in \mathbb{R}^+, d > \eta \quad \forall d' \leq d, v + d' \not\models \mathfrak{D}}{\langle s, v \rangle \xrightarrow{\mathrm{d}} \langle s, v + d \rangle} \tag{1}$$

$$(RA2)\ \frac{\varepsilon \in \mathbb{R}^+ \quad v + \varepsilon \models \mathfrak{D} \wedge \varepsilon \in \eta}{\langle s, v \rangle \xrightarrow{\varepsilon} \langle s, v + \varepsilon \rangle} \tag{2}$$

$$(RD)\ \frac{(s, G, D, a, x, s') \in T_D \quad v \models G}{\langle s, v \rangle \xrightarrow{\mathrm{a}} \langle s', [\{x\} \mapsto 0]v \rangle} \tag{3}$$

Where η is the smallest real quantity of time in which no action occurs.

Remark: According to the proposed operational semantics for the Time-AgLOTOS language, the obtained CTPS structure is an extension of the Durational Action Timed Automata (or DATA for short) of [13], enhancing contextual and intention termination information.

As a simple example, consider the following expression plan $\overline{P_1} = meet(Bob, \ell_1)@t_1[t_1 \leqslant 2]; exit\{0\} \gg move(\ell_2)@t_2[t_2 \leqslant 5 - t_1]; get_copies(\ell_2); exit\{1\}$, the generated CTPS is illustrated in Figure 2. For instance, state S_5 captures the current action under execution which is mentioned by the corresponding clock x, the reached location ℓ_2 and the set of intention plans that are terminated successfully (here it is expressed by $\{\widehat{P_1}, \widehat{P_2}\}$).

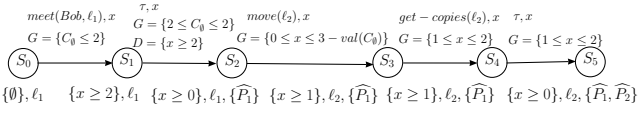


Figure 2. CTPS corresponding to the agent plan $\overline{P_1}$ of Alice

V. DISCUSSION AND RELATED WORK

In this paper, we present an expressive model that provides theoretical foundations and a practical reasoning framework which are dedicated to real-time AmI systems based on agent approach. Up to our knowledge; many works have proposed multi-agent modeling based on logics or behavioral languages, but few of them have dealt timing constraints.

The works of Hoogendoorn et al., e.g. [15], present a multi-agent-based model for a medicine usage, associated with formal analysis methods. In fact, both multi-agent system and corresponding ambient agents are specified in the logical language TTL. This language supports the formal specification and analysis of some dynamic properties. Further, the beliefs and intentions of an agent are presented as predicates in temporal rules which can be used by the agent to perform some simulations. Nevertheless, there is no explicit expression referring to the agent desires or capacities like mobility, rationality, adaptability or context-awareness .

In contrast, our proposed AmI agent model provides a powerful framework for real-time AmI system design, which covers the major features and functionalities of such system. It also combines the mental reasoning of a BDI agent as well as its behavior evolution represented in a powerful algebraic construction within time specification.

In [16], the authors propounded a specification process and a set of formal tools for ambient design activities.They collected some existing languages and techniques which are combined in order to handle different aspects of AmI applications: formal ontology to define the entities of the application domain, ambient calculus (AC) with its related logic to specify the moves of the mobile agents

in MAS and finally, real-time temporal logic (RTTL) to express temporal constraints for reasoning about time.

In contrast to such tool box based on multiple semantics, our approach provides an unified and well-founded algebra methodology for describing plans and checking within them real-time behavioral properties with respect to rich and unique semantics.

Outside of the agent-centric approaches, the methodology proposed in [17] aims at verifying the logical and temporal properties of an intelligent domotic environment. It checks the correctness of such system by using UCTL model checking techniques over the UML State Charts model describing the system.

Unlike to our formal approach which is intended to be concrete, this methodology presents an abstract specification of timing constraints while taking into account concurrency. However, the major features of AmI systems are not handled explicitly in this model.

Another category of works used some algebraic formalisms a-la-LOTOS to specify distributed real-time systems. In [18], the authors present the MDD-LOTOS language as a calculus modeling the mobility in distributed systems under time requirements. This model allows to specify the migration of processes over a set of distributed locations and also the dynamic creation of new processes. Two types of communication are presented, the local communication and the remote one. Like in our approach, they also propose a semantics model for the building of automatic verification tools, based on the maximality semantics. Moreover, it takes into account timing constraints and duration expressions. Due to the AmI requirements, in our agent-based model, the remote communications are between agents, whereas, the synchronized ones are defined locally inside the agent and are related to the parallel composition of its behaviors. So, we provide two distinct designing levels for MAS, the inside agent and outside one including moving ability.

VI. Conclusion

The proposed Time-AgLOTOS agent algebraic language appears to be a powerful way to express agent plan as an hierarchical structure made of concurrent processes. This language provides a theoretical basis to deal with timing constraints, taking into account the duration of actions. The formal semantics of Time-AgLOTOS associates a Contextual Time Planning System (CTPS) with each BDI state of the agent. Hence, from some current set of intentions, this allows one to capture all the possible evolutions of the agent plan, in respect to the timing constraints of the executed actions, under various environmental contexts. Observing that the presented work provides an original time mechanism that works contextually to guide the agent for its future executions, we now aim at combining the CTPS approach with learning techniques, viewed as a guidance mechanism based on past-experiences.

References

[1] J. Augusto, "Ambient intelligence: The confluence of pervasive computing and artificial intelligence," *Schuster, A. (ed.) Intelligent Computing Everywhere. Springer, Heidelberg*, pp. 213–234, 2007.

[2] D. Preuveneers and P. Novais, "A survey of software engineering best practices for the development of smart applications in ambient intelligence," *JAISE*, vol. 4, no. 3, pp. 149–162, 2012.

[3] R. H. Bordini, L. Braubach, M. Dastani, A. El Fallah Seghrouchni, J. J. Gmez-Sanz, J. Leite, G. M. P. O'Hare, A. Pokahr, and A. Ricci, "A survey of programming languages and platforms for multi-agent systems." *Informatica (Slovenia)*, vol. 30, no. 1, pp. 33–44, 2006.

[4] D. Cook, M. Youngblood, and S. Das, "A multi-agent approach to controlling a smart environment," in *Designing Smart Homes*, ser. LNCS. Springer, 2006, vol. 4008, pp. 165–182.

[5] K. E. Nygard, D. Xu, J. Pikalek, and M. Lundell, "Multi-agent designs for ambient systems," in *Proceedings of Ambi-Sys'08*. ICST, 2008, pp. 10:1–10:6.

[6] L. de Silva, A. Dekker, and J. Harland, "Planning with time limits in bdi agent programming languages," in *Proceedings of CATS'07*, vol. 65. Darlinghurst, Australia: Australian Computer Society, Inc., 2007, pp. 131–139.

[7] A.-C. Chaouche, A. El Fallah Seghrouchni, J.-M. Ilié, and D. E. Saïdouni, "A Higher-order Agent Model with Contextual Management for Ambient Systems," in *Transactions on Computational Collective Intelligence XVI*, ser. LNCS. Springer Berlin Heidelberg, 2014, vol. 8780, pp. 1–24.

[8] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research," Dartmouth College, Computer Science, Hanover, NH, Tech. Rep. TR2000-381, November 2000.

[9] K. Henricksen and J. Indulska, "Developing context-aware pervasive computing applications: Models and approach," *Pervasive Mob. Comput.*, vol. 2, no. 1, pp. 37–64, Feb. 2006.

[10] D. E. Saïdouni, "Sémantique de maximalité : Application au raffinement d'actions en LOTOS," Ph.D. dissertation, LAAS-CNRS, Toulouse, France, 1996.

[11] J. P. Courtiat and D. E. Saïdouni, "Relating maximality-based semantics to action refinement in process algebras," *In D. Hogrefe and S. Leue, editors, IFIP TC6/WG6.1, Int. Conf. on FORTE'94*, pp. 293–308, 1995.

[12] D. E. Saïdouni, N. Belala, and M. Bouneb, "Using maximality-based labelled transition system as a model for petri nets," *(IAJIT'2009)*, vol. 5, no. 6, pp. 440–446, 2009.

[13] D. E. Saïdouni and N. Belala, "Actions duration in timed models," *Proceedings of ACIT'2006*, 2006.

[14] J.-P. Courtiat, C. A. S. Santos, C. Lohr, and B. Outtaj, "Experience with rt-lotos, a temporal extension of the lotos formal description technique," *Computer Communications*, vol. 23, no. 12, pp. 1104–1123, 2000.

[15] M. Hoogendoorn, M. C. A. Klein, Z. A. Memon, and J. Treur, "Formal specification and analysis of intelligent agents for model-based medicine usage management," *Comp. in Bio. and Med.*, vol. 43, no. 5, pp. 444–457, 2013.

[16] A. Coronato and G. D. Pietro, "Formal design of ambient intelligence applications," December 2010, pp. 60–68.

[17] F. Corno and M. Sanaullah, "Design time methodology for the formal verification of intelligent domotic environments," in *ISAmI 2011*.

[18] T. M. Maarouk, D. E. Saïdouni, and M. Khergag, "Towards a calculus for distribu-ted, real-time and mobile systems," *Journal of Software*, vol. 7, no. 3, pp. 564–574, 2012.