A Higher-order Agent Model for Ambient Systems

Ahmed Chawki Chaouche, Amal El Fallah-Seghrouchni, Jean-Michel Ilié, Djamel Eddine Saïdouni

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 21 (2013) 156 – 163

Procedia Computer Science

The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2013)

A Higher-Order Agent Model for Ambient Systems

A-C. Chaouche[*,a,b], A. El Fallah Seghrouchni[a], J-M. Ilié[a], D-E. Saïdouni[b]

[a] LIP6 Laboratory, Paris VI, France
[b] MISC Laboratory, Constantine II, Algeria

Abstract

The multi-agent systems (MAS) paradigm provides an interesting alternative for the design of ambient intelligence (AmI) systems. However, the verification of the resulting MAS and their constituent agents still requires a suitable model that meets AmI systems requirements.

The HoA model presented in this paper, aims at representing the mental evolution of a BDI agent, in order to be handled in further verification approaches. This model captures the *reasoning* of a BDI agent and its plan, whose expression is defined from *AgLOTOS*, an original agent algebraic language dedicated to AmI features and capabilities.

Selection and peer-review under responsibility of Elhadi M. Shakshuki

Keywords: AmI systems, MAS, BDI agent, Higher-order agent algebra, verification

1. Introduction

Ambient Intelligence (AmI) is the vision of ubiquitous electronic environment that is non-intrusive and pro-active, when assisting people during various activities [1]. Usually, AmI systems are distributed to be very close from their final users and, whatever the architecture of distribution, pro-activity and context-awareness needs are supported.

This is where the agent-oriented development paradigm comes in, and where the research in Multi-Agent Systems (MAS) can contribute to the design of AmI systems [2]. Indeed, agents provide interesting features that are very much-needed by AmI system designers, like autonomy, pro-activity, reasoning, mobility [3]. Actually, Agents can behave pro-actively (by their own) and can communicate with other agents (e.g. their neighbors). Moreover, many AmI applications implement context-awareness as one of their core features. The agent *context-awareness* includes various information like space, time and environmental considerations, social relationships, knowledge on equipments and state of the user.

Various approaches have emerged as candidates for the studying of agent-oriented systems, e.g. [4, 5, 6, 3, 7]. A well known approach in this area is the BDI architecture [8, 9]. *BDI agents* can be intentional, deliberative and

*Corresponding author
Email address: ahmed.chaouche@lip6.fr (A-C. Chaouche*,)

1877-0509 © 2013 The Authors. Published by Elsevier B.V.
Selection and peer-review under responsibility of Elhadi M. Shakshuki
doi:10.1016/j.procs.2013.09.022

rational. Actually, 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.

Due to the very rich context offered by MAS, many modeling approaches are proposed, which partially focus on some of the MAS aspects. Ones aim at specifying agent oriented systems like AUML [10] and CML [11], other are oriented language e.g. [3, 12]. Also, specific models have been introduced to perform analysis like the Petri Nets family for behavioral aspects, e.g. [13, 14, 15], or oriented logics [16, 17] to formally capture the mental reasoning of agent.

Designing safe and sound BDI agent in MAS requires to model the major of the AmI features and functionalities, and to associate a well-defined semantics in order to validate the MAS execution. The major challenge is to capture the huge dynamic of AmI systems, inherent to several considerations: Not only the AmI systems are open to new agents and allow the agent outputs, but also agent behaviors change consequently to agent mental reasoning.

This paper aims to introduce an original agent-based modeling in order to allow the checking and the validation of an agent execution. In this model, each agent has a partial view of the system but endowed with ambient and intelligent capabilities: (a) the *mental state* integrates the three mental attitudes: Beliefs, Desires and Intentions, allowing to select plans of actions, (b) the *behavioral planning state* expressed by a plan language which includes the several AmI features like communication and mobility.

The paper is organized as follows : Section 2 details the considered AmI features and the BDI architecture. This allows us to introduce in Section 3, our Agent model, namely Higher-order agent model (*HoA*), which provides a specification and verification framework for AmI agents. A realistic scenario is given as an illustration of our approach. In Section 4, the specifications of the mental and planning state of the agent are formally defined. The key point is an agent algebra called *AgLOTOS*, proposed to specify the agent plans. The operational semantics of *AgLOTOS* specifications allow one to process verification. The last section concludes and outlines our perspectives.

2. AmI requirements

The AmI systems we consider are open space systems. Their agents have complex features as schemed in Figure 1. An agent is assumed to be **autonomous**, thus operating without the direct intervention of humans or other agents. It is **anticipative** and **pro-active**, in order to process **rational** decision, based on its own knowledge and beliefs.

As a corollary of autonomy, an AmI agent is **context-aware**. We see the context of an agent as every environmental information perceived by the agent, in particular vicinity notions in a domain that considers space, time and social relationships. Thus, determining if a piece of information is relevant for an agent should be done based on its local context information.

To improve behavior and knowledge, an AmI agent can communicate and cooperate with its neighbors. Also, AmI agent can be mobile moving from one locality to another in a given space.

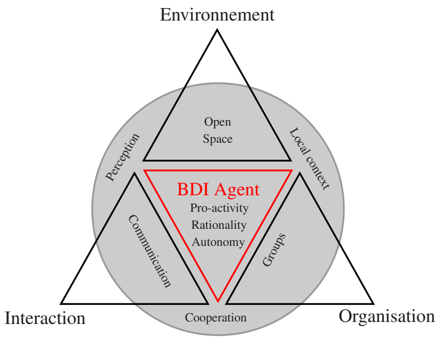

Figure 1: Agent features for AmI systems

BDI Architecture

The Belief-Desires-Intentions (BDI) approach is one of the major approaches to building agents and multi-agent systems [16, 18]. **BDI agents** are able to be intentional, deliberative or rational. The reasoning mechanism is triggered

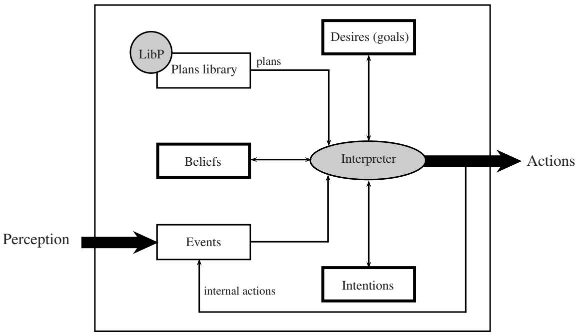

Figure 2: Architecture of a BDI agent

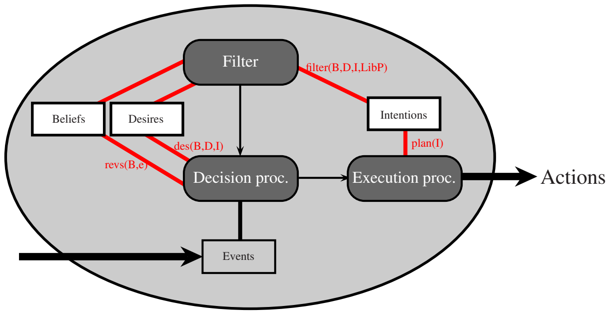

Figure 3: A functional view of a BDI interpreter

by the perceived events, based on some representations of beliefs (B), desires (D), intentions (I). Consistent action plans are produced, helped by using a library of (partial) plans (LibP) [19]. Inspired from [20], Figure 2 shows the main components of a BDI architecture. Many works have detailed the BDI interpreter; e.g. [4, 21, 22] and Figure 3 illustrates the major aspects of its functionality.

- $revs : B \times Evt \rightarrow B$ is the belief revision function applied when the agent receive a new event.
- $des : B \times D \times I \rightarrow D$ is the Desire update function that maintains consistency with the selected desires,
- $filter : B \times D \times I \times LibP \rightarrow I$ is the Intention function which yields the intentions the agent decides to pursue, among the possible options, taking into account new opportunities.
- $plan : I \rightarrow P$ is a Plan function that process an executable plan from some (filtered) Intention, knowing that any intention can be viewed as a partial plan. This function uses the plan library (represented by the *LibP module* in Figure 2).

3. The Higher-Order Agent Model (HoA)

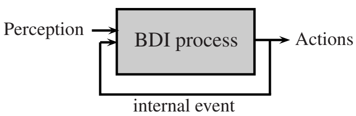

Figure 4: BDI process

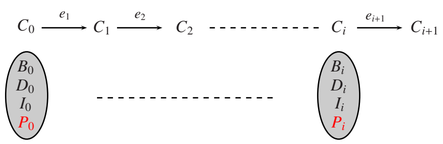

Figure 5: The agent behavioral changes

In this paper, we are interested in modeling the evolution of the agent globally. The BDI process is globally viewed as state-transition-based system, which reacts to events and iteratively selects a plan from an update of the believe, desire, and intention structures, w.r.t. the library of plans. Our agent-based model captures two main aspects of the agent : (a) the mental reasoning of the agent and its *BDI state*, (b) the evolution of the selected plan and its *Planning state*.

Further, a state of an agent is called a configuration. Each configuration is composed of a *BDI state* and a selected plan, knowing that the operational semantics of plans can yield all the possible evolutions implied by the selected plan (see Section 4.2). As illustrated by Figure 5, the occurrences of events may cause some changes of configurations. In our work, two kinds of events are considered from some configuration : (*a*) a *relevant event* causes an effective change of configurations. It is due to an update of the BDI state implied by the information embedded with the event. Of course, the change of the BDI state can also cause the change of the selected plan. (*b*) Secondly, a *negligible events* only causes a change of the planning state within the selected plan, keeping safe the BDI state of the configuration.

We now define our *Higher-order agent* model, named *HoA* for short. This model focuses on the evolution of the mental reasoning of the agent. Observe that the BDI and planning states are formally expressed in Section 4.

3.1. Formalization of an AmI agent

Definition 3.1. *Considering any agent of the AmI system, let BDI be the set of all possible states of the BDI structure and P be the set of possible plans. The HoA model of the agent is a transition system Ω, represented by a tuple $\langle C, C_0, T, L_M, L_P \rangle$, where :*

- *C is the set of agent configurations,*
- *C_0 is the initial configuration such that $C_0 \subseteq C$,*
- *$T \subseteq C \times Evt \times C$ is the set of transitions between configurations, assuming an interleaving semantics of events,*
- *$L_M : C \rightarrow BDI$ associates with each configuration a BDI state,*
- *$L_P : BDI \rightarrow P$ associates with each BDI state a plan.*

The HoA model is defined over an alphabet of events triggered by the actions being executed and by perception events, namely $Evt = E_{Act} \cup Perc$. Moreover, the HoA model differently considers two kinds of events, from some configuration : (a) a *relevant event* causes an effective change of configurations, due to an update of the BDI state implied by the information embedded with the event. Of course, the change of the BDI state can also cause the change of the agent plan. (b) Secondly, a *negligible events* only causes a change of the planning state, keeping safe the BDI state of the configuration.

3.2. Scenario.

	Alice	Bob
C_0	$B_0 = \{in(me, \ell_1), in(copies, \ell_2)\}$ $D_0 = \{meeting(Bob, \ell_1), get\_copies(\ell_2)\}$ $I_0 = meeting(Bob, \ell_1)$ $P_0 = meeting; exit$	$B_0 = \{in(me, \ell_2)\}$ $D_0 = \{meeting(Alice, \ell_1)\}$ $I_0 = meeting(Alice, \ell_1)$ $P_0 = Alice?(x); exit \vert\vert\vert move(\ell_2, \ell_1); meeting; exit$
C_1	$B_1 = \{in(me, \ell_1), in(copies, \ell_2), in(Bob, \ell_2)\}$ $D_1 = \{meeting(Bob, \ell_1), ask(Bob, get\_copies(\ell_2))\}$ $I_1 = ask(Bob, get\_copies(\ell_2)) >> meeting(Bob, \ell_1)$ $P_1 = Bob!(get\_copies(\ell_2)); exit >> meeting; exit$	$B_1 = \{in(me, \ell_2), in(copies, \ell_2)\}$ $D_1 = \{meeting(Alice, \ell_1), get\_copies(\ell_2)\}$ $I_1 = get\_copies(\ell_2) >> meeting(Alice, \ell_1)$ $P_1 = get\_copies(\ell_2); exit >> move(\ell_2, \ell_1); meeting; exit$

Table 1: A state evolution for Alice and Bob

Let Alice and Bob be two agents of an AmI Universitary system. Such a system is clearly open since agents can enter and leave. The fact that Bob is entering the system can be perceived by Alice in case she is already in. Since Alice is context aware, she can take advantage of this information, together with other information like the fact she is able to communicate with Bob through the system.

Let $L = \{\ell_1, \ell_2\}$ be 2 localities of the system where the agents behave. The proposed problem of Alice is that she does not make the two following tasks in the same period of time : *(a)* to meet with Bob in ℓ_1, and *(b)* to get her exam copies from ℓ_2. Clearly, the Alice's desires are inconsistent since Alice cannot be in two distinct places simultaneously.

The table 1 represents certain evolution of the configurations of these agents, once both agents are within the system. The initial configurations of Alice and Bob are respectively C_0^A and C_0^B, s.t. Alice is in ℓ_1 and has the mentioned two inconsistent desires, whereas Bob is in ℓ_2 and he is desiring to work with Alice. The current intention of Alice is only to meet with Bob. Here, BDI information is simply expressed by using intuitive predicate assertions. Moreover, plans are expressions formally defined in Section 4.2.

The AmI scenario evolves as it is shown in Figure 6 by the HoA transition systems of Alice and Bob. The changes of configurations leads to the configurations C_1^A and C_1^B, separately, according to the respective perceptions of Alice and Bob and the fact that they are pro-active and anticipative. Actually, after having perceived that Bob is in ℓ_2 ($evt_1 = perc(in(Bob, \ell_2))$), thus in the same localities as the exam copies, Alice enriches her beliefs, desires and intentions, aiming to communicate to Bob in order to be helped by him in order to get the copies. Consequently, she evolves to the new configuration C_1^A, where the generated plan suggests to Bob to bring copies. She asks Bob for helping her, by sending the message $Bob!(get\_copies(\ell_2))$.

When Bob receives the message $Alice?(x)$, a relevant event is triggered to him, $evt_2 = Alice?(x)$). He accepts to bring copies to Alice, then expands his beliefs ($in(copies, \ell_2)$) and also considers a new desire $get\_copies(\ell_2)$, in fact consistent with his previous ones. Consequently, the state of Bob is changing to the new configuration C_1^B, and a new plan is generated to satisfy all of his desires, getting first the copies then going to meet Alice.

In addition to the already mentioned events, the negligible event evt_3 specifies that *Bob* can start executing his last plan, e.g. $get\_copies(\ell_2)$.

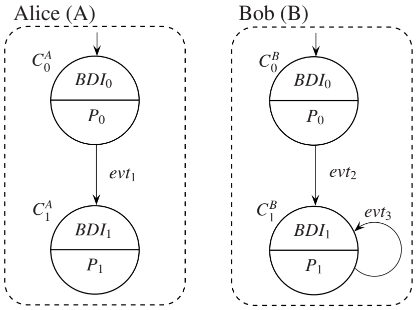

Figure 6: HoA modular representation

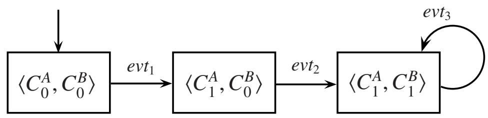

Figure 7: One possible evolution of AmI system scenario

Figure 7 illustrates the resulting evolution of our AmI system scenario. A state of the system is composed of one configuration of Alice, one for Bob. In fact, we abstract the communication transfer assuming that the receiving of messages is guarantied based-on a reliable communication network.

4. Mental and planning specification

In this section, we describe how to formally specify the mental and planning states of our model. Actually, many representations are possible, among these the one used in our scenario. We also adopt the Rao and Georgeff's *BDI logic*, which is known to be very appropriate for modeling rational and pro-active BDI agents [23]. As far as *planning states* are concerned, we propose an algebraic language, called *AgLOTOS*, as a compact representation of plans.

4.1. Mental state (BDI State)

In the BDI logic of Rao and Georgeff's, namely BDI_{CTL}, each agent is viewed as having three mental attitudes: belief, desire, and intention. BDI_{CTL} is a multi-modal logic whose basic syntax is expressed as follows :

$$\varphi \quad ::= \quad true \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid X\varphi \mid \varphi U \varphi \mid F\varphi \mid E\varphi \mid Bel(\varphi) \mid Des(\varphi) \mid Int(\varphi)$$

Actually, BDI_{CTL} includes standard *computational tree logic* operators, either CTL or CTL^*, combined with three epistemic modal operators for the agent's beliefs, desires, and intentions. The semantics of the logic is given through a rich Kripke model within which each world has an internal branching structure to represent the evolution of a given mental state of mind [17].

4.2. Planning state

The planning is an complex operation often restricted to a sequence of tasks in practical MAS platforms and languages, e.g. [7, 24, 25, 3]. Rather, we propose to consider tasks of plans as processes in a concurrent system and aim at taking profit from the rich expressivity offered by the specification languages used to describe such a system. In this paper, we extend the *LOTOS* language [26], to specify plans in terms of an algebraic language called *AgLOTOS*. Our specification is agent-based, expressing the syntax and semantics of an agent plan. The augmented primitives concern 3 key points involved by the AmI system dynamicity. An agent can perceive the enter and leave of other agents in the AmI system, it can move between the AmI system localities and can communicate with some agent in the system.

The syntax of a plan P is recursively defined as follows :

$$\begin{array}{lcll} P & ::= & exit \mid stop & \\ & & \mid a; P \mid P \odot P & \\ & & \mid hide\ L\ in\ P & \\ & & \mid x!(v) \mid x?(v) & \textit{Communcation} \\ & & \mid in(\ell) \mid out(\ell) \mid move(\ell, \ell) & \textit{Mobility} \\ \odot & = & \{\texttt{|||}, \texttt{|[L]|}, \texttt{||}, \texttt{[]}, \texttt{>>}, \texttt{[>}\} & \end{array}$$

Any term of an *AgLOTOS* expression features a plan expression, then plan can be composed from (partial) plans. *AgLOTOS* relates to the basic *LOTOS* terms : *exit* for the successful termination and *stop* for the abnormal termination of some (partial) plan. We retrieve the basic $\odot$ operators of *LOTOS*, in particular, the non-deterministic choice $P[]P$, the interiorization *hide L in P*, the sequential composition $P >> P$; the interruption $P[> P$. Also observe that the *LOTOS* parallel composition $P|[L]|P$ can model both synchronous composition, $P||P$ if $L = \mathcal{G}$, and asynchronous composition, $P|||P$ if $L = \emptyset$. Thus, *AgLOTOS* language exhibits a rich expressivity s.t. the sequential executions of plans appears to be only a particular case.

Agent mobility is expressed by the basic primitives *in*, *out*, e.g. the operation $in(\ell)$, is used to place the agent in some locality ℓ. Moreover, *move* is an extension composing the *out* and *in* primitives, to easily handle the agent move from a locality to another.

The communication syntax primitives are inspired from those of the π-calculus primitives, however considered within a totally dynamic communication support, hence without specification of predefined channels. The expression $x!(m)$ specifies the emission to the agent x of some message m, whereas, the expression $x?(m)$ means that m is received from some agent x.

In this paper, the plan of any agent configuration is expressed as an *AgLOTOS* expression. In particular, we are able to define the possible planning states of the agent according to its configuration.

The semantics of *AgLOTOS* is brought out by the derivation rules of Table 2. From any *AgLOTOS* plan expression, these rules are applied to formally produce a labeled transition system representing the possible evolution of the plan. As plans are embedded in configurations, these allows to define the Planning Transition System of any configuration C, denoted $PTS(C)$, like a tuple $(PS, ps_0, \rightarrow)$ where :

- PS is the set of planning states of the plan,
- ps_0 is the initial planning state of the agent such that $s_0 \in S$,
- $\rightarrow \subseteq PS \times Act \times E_{Act} \times PS$ is the set of transitions.

The derivations rules of the *AgLOTOS* are formally defined in Table 2. The first table part is inspired by the standard LOTOS rules, whereas the second shows the semantics of our new primitives. Let P, P', Q, Q' be plans. Let $\mathcal{G}$, ranged over g, be the set of the so-called LOTOS gates, that means observable actions, let $i \notin \mathcal{G}$ be the internal action and $\delta \notin \mathcal{G}$ the successful termination action. The set of possible actions is standardly defined as $Act = \mathcal{G} \cup \{i, \delta\}$ and L denotes any finite subset of $\mathcal{G}$.

Unlike standard LOTOS, we augment the derivation rules to take an event system mechanism into account, able to ensure the consistency between the mental and planning states of the agent. In fact, any action in a plan causing an update of the mental state could be captured. In the derivation rules, the transformation of plans, denoted $P \xrightarrow{a,e} Q$, specifies that the plan Q is derived from P, when the action a is launched and the event e is triggered. Usually, the occurrences of e relates to the end of a. The possible events are described by the set $E_{Act} = \{e_T, e_L, e_R, e_N\}$, the elements of which are respectively, the termination event, the locality changing event, the receiving message event and the negligible event. For sake of clarity, a derivation $\xrightarrow{a,e_N}$ is simply represented as $\xrightarrow{a}$.

Two new rules are brought out in the second part of Table 2. Let Θ be a finite set of space localities and assume that each agent can communicate with a finite set Λ of AmI agents. In our approach, Λ is a dynamic set which evolves according the agent contextual perceptions concerning the entry and exit of (other) agents in the AmI system. Moreover, we consider that communication towards an agent is reliable, provided this agent belongs to Λ.

Definition 4.1. *W.r.t. any AmI agent A, let C be any configuration of the HoA transition system of A and P the corresponding plan, then the possible planning states of A in C are the planning states of $PTS(C)$.*

(Termination)	$\frac{}{exit \xrightarrow{\delta,evt_T} stop}$		
(Prefixing)	$\frac{}{a;P \xrightarrow{a} P}$		
(Choice)	$\frac{P \xrightarrow{a} P'}{Q\ [\,]\ P \xrightarrow{a} P' \qquad P\ [\,]\ Q \xrightarrow{a} P'}$		
(Parallel)	$\frac{P \xrightarrow{a} P' \quad a \notin L \cup \{\delta\}}{P \vert[L]\vert Q \xrightarrow{a} P' \vert[L]\vert Q}$	$\frac{P \xrightarrow{a} P' \quad a \notin L \cup \{\delta\}}{Q \vert[L]\vert P \xrightarrow{a} Q \vert[L]\vert P'}$	$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{a} Q' \quad a \in L \cup \{\delta\}}{P \vert[L]\vert Q \xrightarrow{a} P' \vert[L]\vert Q'}$
(Interiorization)	$\frac{P \xrightarrow{a} P' \quad a \notin L}{hide\ L\ in\ P \xrightarrow{a} hide\ L\ in\ P'}$	$\frac{P \xrightarrow{a} P' \quad a \in L}{hide\ L\ in\ P \xrightarrow{i} hide\ L\ in\ P'}$	
(Sequence)	$\frac{P \xrightarrow{a} P' \quad a \neq \delta}{P \gg Q \xrightarrow{a} P' \gg Q}$	$\frac{P \xrightarrow{\delta} P'}{P \gg Q \xrightarrow{i} Q}$	
(Interruption)	$\frac{P \xrightarrow{a} P' \quad a \neq \delta}{P\ [>\ Q \xrightarrow{a} P'\ [>\ Q}$	$\frac{P \xrightarrow{\delta} P'}{P\ [>\ Q \xrightarrow{\delta} P'\ [>\ stop}$	$\frac{Q \xrightarrow{a} Q'}{P\ [>\ Q \xrightarrow{a} stop\ [>\ Q}$
(Substitution)	$\frac{P \xrightarrow{a} P' \quad a \notin \{a_1,\ldots,a_n\}}{P[b_1/a_1,\ldots,b_n/a_n] \xrightarrow{a} P'[b_1/a_1,\ldots,b_n/a_n]}$	$\frac{P \xrightarrow{a} P' \quad a = a_i\ (1 \leq i \leq n)}{P[b_1/a_1,\ldots,b_n/a_n] \xrightarrow{bi} P'[b_1/a_1,\ldots,b_n/a_n]}$	
(Communication)	$\frac{P \xrightarrow{x!(v)} Q \quad x \in \Lambda}{P \xrightarrow{i} Q}$	$\frac{P \xrightarrow{x?(v)} Q \quad perc_C(x,msg) \quad x \in \Lambda}{P \xrightarrow{i,e_R} Q\{msg \backslash v\}}$	
(Mobility)	$\frac{P \xrightarrow{in(\ell)} Q \quad \ell \in \Theta}{P \xrightarrow{i,e_L} Q}$	$\frac{P \xrightarrow{out(\ell)} Q \quad \ell \in \Theta}{P \xrightarrow{i,e_L} Q}$	$\frac{P \xrightarrow{move(\ell_1,\ell_2)} Q \quad \ell_1,\ell_2 \in \Theta}{P \xrightarrow{out(\ell_1)} P' \xrightarrow{in(\ell_2)} Q}$

Table 2: Semantic rules of *AgLOTOS*

As a simple example taken from our scenario (Section 3.2), the plan expression P_1 of the configuration C_1^B of Bob, $PTS(C_1^B) : ps_0 \xrightarrow{get\_copies} ps_1 \xrightarrow{\delta,e_T} ps_2 \xrightarrow{move(\ell_2,\ell_1)} ps_3 \xrightarrow{meeting} ps_4 \xrightarrow{\delta,e_T} ps_5$. The evt_3 negligible event represents the fact that the planning state of Bob can be in one of the PTS states.

5. Conclusion

The Higher-order agent model formally represents BDI-AmI open systems where agents can reason, communicate and move. Agent dynamicity and context awareness are handled due to the fact that agents can change their mental state adequately to the perceptions of new events. The proposed *AgLOTOS* agent-based algebra appears to be a powerful and intuitive way to express an agent plan. The presented scenario shows how the HoA model is rich and in fact can be adapted to many MAS system scenarii.

The key point of our approach is the defined HoA semantics, expressed in two levels. The (external) HoA level captures the mental evolution of the agent, whereas the (internal) level allows to reason on both the BDI state and its associated planning transition system (PTS) which represents the planning evolution of the agent. The evolution consistency of these levels is ensured by the proposed event system mechanism and the BDI process of the agent.

In the HoA model, the fact that both external and internal levels can be represented by transition systems, allows one to perform various agent-based verification. First, one can perform BDI statical analyses and tests of mental consistency as proposed for instance in [27]. As an original contribution, the PTS representation allows to check the agent consistency between the mental state of the agent and its executing plan. Moreover, verification modules can be added to the specification of agent, exploiting the represented agent mental evolution to handle higher concerns like learning, guidance and prevention aspects. Therefore, our next perspective is to show how the HoA transition systems can be reduced according to the dynamic properties to check.

References

[1] A. Olaru, A. M. Florea, A. El Fallah-Seghrouchni, A context-aware multi-agent system as a middleware for ambient intelligence, MONET 18 (3) (2013) 429–443.

[2] A. El Fallah-Seghrouchni, A. M. Florea, A. Olaru, Multi-agent systems: A paradigm to design ambient intelligent applications., in: M. Essaaidi, M. Malgeri, C. Badica (Eds.), IDC, Vol. 315, 2010, pp. 3–9.

[3] A. El Fallah-Seghrouchni, A. Suna, Claim: A computational language for autonomous, intelligent and mobile agents., in: M. Dastani, J. Dix, A. El Fallah-Seghrouchni (Eds.), PROMAS, Vol. 3067 of Lecture Notes in Computer Science, Springer, 2003, pp. 90–110.

[4] M. E. Bratman, D. J. Israel, M. E. Pollack, Plans and resource-bounded practical reasoning., Computational Intelligence 4 (1988) 349–355.

[5] J. Doyle, Rationality and its roles in reasoning., Computational Intelligence 8 (1992) 376–40.

[6] A. S. Rao, M. P. Georgeff, Modeling rational agents within a bdi-architecture., in: J. F. Allen, R. Fikes, E. Sandewall (Eds.), KR, Morgan Kaufmann, 1991, pp. 473–484.

[7] Y. Shoham, Agent-oriented programming., Artif. Intell. 60 (1) (1993) 51–92.

[8] A. S. Rao, M. P. Georgeff, An abstract architecture for rational agents., in: B. Nebel, C. Rich, W. R. Swartout (Eds.), KR, Morgan Kaufmann, 1992, pp. 439–449.

[9] A.-C. Chaouche, A. El Fallah-Seghrouchni, J.-M. Ilié, D.-E. Saidouni, Towards a MAS approach to model Ambient Systems, 8th NII-LIP6 WorkShops on Multi-Agent and Distributed Systems, 2013.

[10] J. Odell, H. Parunak, B. Bauer, Extending UML for agents, in: Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National Conference on Artificial Intelligence, 2000.

[11] G. Schreiber, B. J. Wielinga, H. Akkermans, W. V. de Velde, A. Anjewierden, Cml: The commonkads conceptual modelling language., in: L. Steels, G. Schreiber, W. V. de Velde (Eds.), EKAW, Vol. 867 of Lecture Notes in Computer Science, Springer, 1994, pp. 1–25.

[12] T. M. Maarouk, D.-E. Sadouni, M. Khergag, Towards a calculus for distributed, real-time and mobile systems., JSW 7 (3) (2012) 564–574.

[13] A. El Fallah-Seghrouchni, S. Haddad, H. Mazouzi, Protocol engineering for multi-agent interaction, in: Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: MultiAgent System Engineering, MAAMAW '99, Springer-Verlag, London, UK, UK, 1999, pp. 89–101.

[14] S. Kouah, D. E. Sadouni, J. M. Ilie, Synchronized petri net: A formal specification model for multi agent systems, Journal of Software 8 (3) (2013) 587–602.

[15] D. Dahmani, J.-M. Ilié, M. Ioualalen, Reachabilty Analysis For Recursive Petri Nets With Shared Places, in: International Workshop on Abstractions for Petri nets Analysis and Other Models of Concurrency (APNOC'2009), in conjunction with the 30th Int. Conf. on Application and Theory of Petri Nets and Other Models of Concurrency (Petri Nets 2009), P13, 2009, pp. 65–79, moVe INT LIP6.

[16] P. R. Cohen, H. J. Levesque, Intention is choice with commitment., Artif. Intell. 42 (2-3) (1990) 213–261.

[17] A. S. Rao, M. P. Georgeff, Decision procedures for bdi logics, Journal of Logic and Computation 8 (3) (1998) 293–342.

[18] A. S. Rao, M. Georgeff, BDI agents: from theory to practice, in: First International Conference on Multi-Agent Systems (ICMAS-95), S. Francisco, CA, 1995, pp. 312–319.

[19] M. P. Georgeff, B. Pell, M. E. Pollack, M. Tambe, M. Wooldridge, The belief-desire-intention model of agency., in: J. P. Mller, M. P. Singh, A. S. Rao (Eds.), ATAL, Vol. 1555 of Lecture Notes in Computer Science, Springer, 1998, pp. 1–10.

[20] M. d'Inverno, D. Kinny, M. Luck, M. Wooldridge, A formal specification of dmars., in: M. P. Singh, A. S. Rao, M. Wooldridge (Eds.), ATAL, Vol. 1365 of Lecture Notes in Computer Science, Springer, 1997, pp. 155–176.

[21] M. E. Pollack, The uses of plans., Artif. Intell. 57 (1) (1992) 43–68.

[22] M. E. Pollack, J. F. Horty, Ther's more to life than making plans: Plan management in dynamic, multi-agent environments, AI Magazine 20 (1999) 71–84.

[23] M. Dastani, L. W. N. van der Torre, An extension of bdictl with functional dependencies and components., in: M. Baaz, A. Voronkov (Eds.), LPAR, Vol. 2514 of Lecture Notes in Computer Science, Springer, 2002, pp. 115–129.

[24] A. S. Rao, AgentSpeak(L): BDI agents speak out in a logical computable language, in: R. van Hoe (Ed.), Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, 1996.

[25] K. V. Hindriks, F. S. de Boer, W. van der Hoek, J.-J. C. Meyer, Agent programming in 3apl., Autonomous Agents and Multi-Agent Systems 2 (4) (1999) 357–401.

[26] E. Brinksma (Ed.), ISO 8807, LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour, 1988.

[27] J. Sudeikat, L. Braubach, A. Pokahr, W. Lamersdorf, W. Renz, Validation of bdi agents, in: Proceedings of the 4th international conference on Programming multi-agent systems, ProMAS'06, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 185–200.