A new BDI architecture to formalize cognitive agent behaviors into simulations

Van-Minh LE[1], Benoit GAUDOU[2] and Patrick TAILLANDIER[3]
[1] *Institut de la Francophonie pour l'Informatique (IFI), Hanoi, Vietnam*
[2] *UMR 5505 IRIT, CNRS – University of Toulouse*
[3] *UMR IDEES, CNRS – University of Rouen*
vanminh.le246@gmail.com, benoit.gaudou@univ-tlse1.fr,
patrick.taillandier@univ-rouen.fr

Abstract. Nowadays, agent-based modeling is more and more used to study complex socio-ecological systems. These last years have also seen the development of several agent-based simulation platforms. These platforms allow modelers to easily and quickly develop models with simple agents. However, socio-ecological systems need agents able to make decisions in order to represent human beings and the design of such complex agents is still an open issue: even with these platforms, designing agents able to make complex reasoning is a difficult task, in particular for modelers that have no programming skill. In order to answer the modeler needs concerning complex agent design, we propose a new agent architecture based on the BDI paradigm and integrated into a simulation platform (GAMA). This paradigm allows designing expressive and realistic agents, yet, it is rarely used in simulation context. A reason is that most agent architectures based on the BDI paradigm are complex to understand and to use by non-computer-scientists. Our agent architecture answers this problem by allowing modelers to define complex cognitive agents in a simple way. An application of our architecture on a model concerning forest fire and firefighter helicopters is presented.

Keywords: Cognitive agent design, BDI architecture, GAMA modeling and simulation platform

1 Introduction

The use of agent-based simulation to study socio-ecological systems is booming since twenty years: for example DeAngelis' work in ecology [6], Tesfatsion's in economy [16] or Gilbert's in social sciences [7]. In fact, agent-based simulation is a powerful tool to study complex systems, in particular when these ones deal with human beings. However, in most of actual models, the modeling of the human behavior is still rather simple and this has a negative impact on the model realism. A reason is the difficulty to design cognitive agents.

These last years have seen the emergence of several platforms easing the development of agent-based models (NetLogo [9], GAMA [15]…). However, even with these platforms, the problem of the cognitive agent design is still an open issue. In fact, designing agents able to make complex reasoning is a difficult task, in particular for researchers that have no programming skill.

In this paper, we propose new cognitive agent architecture based on the BDI (Belief, Desire, Intention) [11] paradigm and directly integrated into the GAMA simulation platform. Our architecture allows modelers to define complex agents and at the same time to understand and to use them easily.

The paper is organized as follows. In Section 2, we present a brief state of the art of agent architectures. Section 3 is dedicated to the presentation of our BDI agent architecture. In Section 4, we present an application of the architecture in a context of a model concerning helicopter firefighters. At last, Section 5 concludes.

2 State of the art

The problem of the agent design is a classic problem in agent-based simulations and numerous formalisms and architectures were proposed.

Among all these formalisms or architectures, the finite state machines or the motivational architecture [12] can be very useful when designing simple agents, but are not adapted to complex cognitive agents as their representation capability is fairly limited.

A classic paradigm to formalize the behavior of cognitive agents is the BDI (Belief, Desire, Intention) paradigm [11]. This paradigm allows designing expressive and realistic agents. It has been implemented in a huge number of architectures. Most of existing architectures are based on Bratman's resource-bounded reasoning principle [4,5] and on the PRS (Procedural Reasoning System) framework [10].

The PRS framework [10] makes the assumption that an intelligent agent should have a thinking process before reacting. This framework includes three main processes: the perception (in which agent acquires information from the environment), the central interpreter (which helps the agent to deliberate its goals and then to select the available actions) and the execution of intention (which represents agent's reactions).

The resource-bounded reasoning concept proposed in [4] makes the assumption that the thinking process costs resources (at least computation-time). In order to cope with this issue, Bratman proposes to use plans. In fact, an agent does not have to take into account all the perceptions and actions, but it only has to consider a set of pre-defined plans that depend on the current situation.

The idea of using plans was implemented in the AgentSpeak architecture (JASON) proposed by Bordini [2]. In this architecture, an agent begins its planning phase only when a trigger comes from a change in the agent's mental states. Besides, no desire is considered in the planning phase.

Another classic "plan-based" BDI architecture is JAM (proposed in [8]). In contrary to the AgentSpeak architecture, plans in JAM contain the desires as goals to pursuit. In this architecture, the agent's behaviors are implemented through the plans: agents are more reactive than cognitive.

Some articles have shown the interest of using the BDI paradigm in simulation context (e.g. [1] or [13]), yet, it is still rarely used. A reason is that most agent architectures based on the BDI paradigm (e.g. JAM and JASON) are complex to understand and to use by non-computer-scientists.

3 BDI architecture

3.1 Architecture overview

A global overview of our architecture is presented in **Fig. 1**. Each part is described in details in the sequel. Our architecture is based on both the PRS framework [10] and the resource-bounded reasoning principle. We propose to use the same type of event-based approach as JASON one [2].

3.2 Description of the BDI components

Beliefs represent the informational states of the agent. In BDI logic, $Bel_i(\varphi)$ expresses the fact that the agent i believes φ. In our proposition, the belief is directly associated with the agent, so the agent i is omitted from the description of a belief. The content of the belief expresses a state or an activity concerning the agent or its world. We propose to use a triplet *(subject, predicate, object)* to describe this content. For example: *Bel(helicopter1, takeCare, tree12)* expresses that the agent has a belief that agent *helicopter1* will take care of the agent *tree12*. While the formula *Bel(X, burning, null)* means that the agent believes that agent (tree) X is burning (i.e. in the state burning), the *Bel(X, bel(tree12, burning, null), null)* declares that the agent believes that the gent X believes that the agent tree12 is burning.

Desires express the motivational states of the agent. As desires are also mental states like beliefs, a desire shares with a belief the same content format. From Bratman's idea that an agent cannot simultaneously pursuit competing desires, we propose two additional attributes to describe a desire:

- *Competing category*: each desire belongs to a competing category. Two desires of the same category cannot be considered at the same time (except for the desires belonging to no category).
- *Priority*: the priority is the degree of importance of a desire. The higher the priority, the more important the desire is. Among the desires of a competing category, the agent chooses the desire with the highest priority.

Events trigger the reactive activity of the agent. In the AgentSpeak architecture, an agent begins reacting when its mental states change. In this architecture, events are described as a creation or a deletion of a mental state. It means that when an agent acquires a new belief or a new desire, this agent creates an event. For example, when an agent perceives a new burning tree (tree12), it first gets the belief and then an event corresponding to this new belief creation is triggered. In our architecture, we reuse the format of event proposed by Bordini: an belief creation event is for example represented by *Bel(+,tree12, burning, null)*

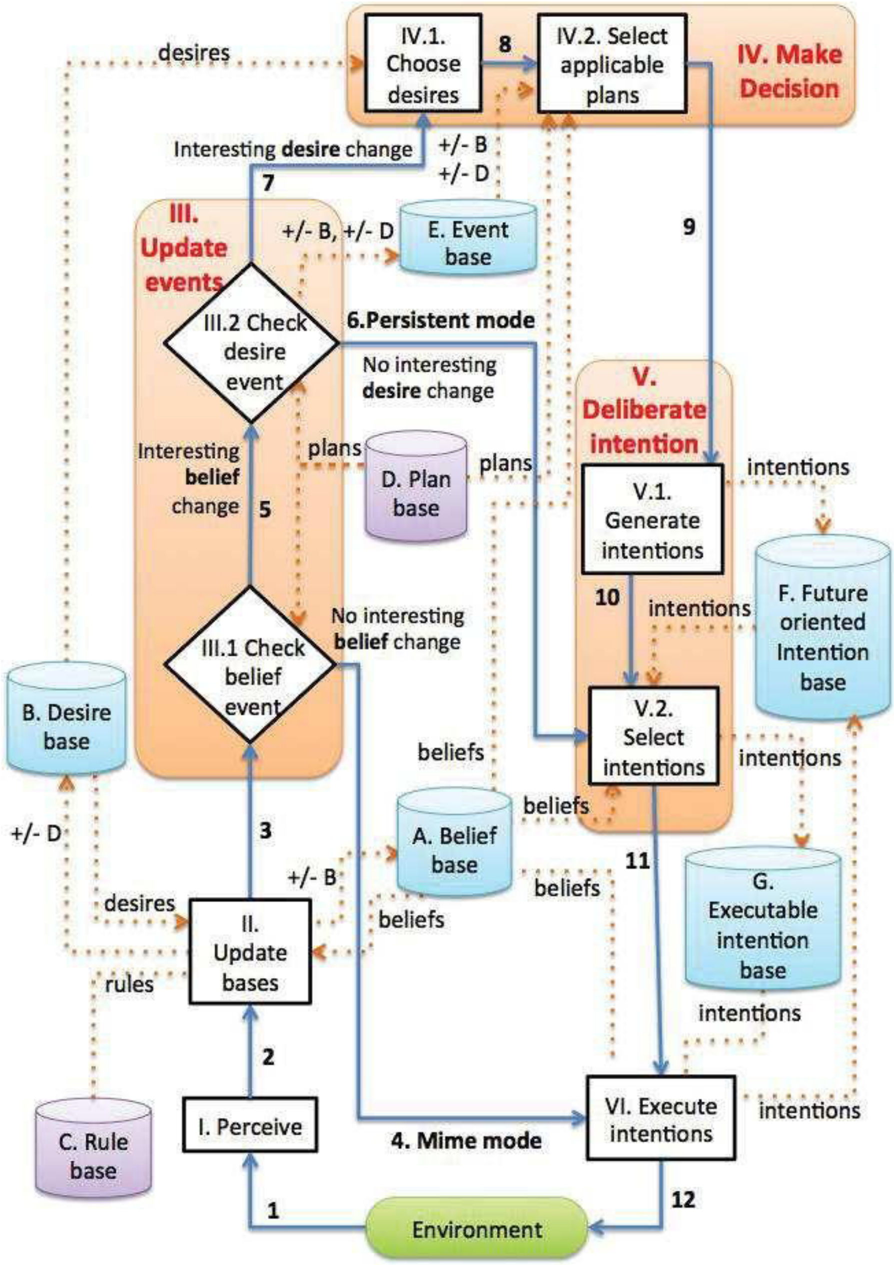

Fig. 1. BDI Architecture

Rules are what an agent uses to make the logic deduction creating new beliefs and new desires from current beliefs and desires. The modifications of mental states come not only from what the agent perceives but also from its internal reasoning process. That means that an intelligent agent is also capable to reason in order to update its beliefs and desires according to its current mental states. As the first order logic is used to describe the beliefs and desires, we propose to use the implicative normal form to describe the rules. As example, the following expression means that an agent would create the desire to go toward another agent (tree agent) if it perceives that this agent is burning and if it is carrying water:

Bel (_self, carrying, water) ^ Bel(X, burning, _null) => Des (_self, goto, X) ;

A **Plan** is a sequence of declared actions that the agent has to apply to reach one (or many) goals. This means that a plan describes the fact that the agent has to execute some particular actions once it gets a specific condition on mental states (beliefs, desires).

A plan is composed of:

- *Goals*: desires of the agent,
- *Context*: conditions on mental states,
- *Trigger*: events that trigger the plan,
- *Actions*: actions to execute.

The following plan declares that the agent will execute the action *.informFire(X)* (to tell other agents that the agent X is on fire) and then the action *.goto(X)* (to move to the agent X) to satisfy the desire to extinguish the fire on the agent X when it is carrying water[1] and it gets the information that the agent X is burning. The agent begins to reason on this plan only when it gets the new belief that an agent X is burning.

Plan:

goal(Des(_self, extinguish, X));
context(bel(X, burning, null),Bel(_self, carryingWater, null));
trigger(Bel(+, X, burning, null));
action(.informFire(X), .goto(X));

When all the conditions of a plan (the *goal*, the *context* and the *trigger*) are satisfied – which means that the agent has the beliefs, the desires and the events matching with the all the variables of the plan conditions – the plan is considered as applicable.

The **Intention** represents the deliberative state of the agent, *i.e.* what the agent has chosen to do. According to Bratman, "intending to do something (or having an intention) and doing something intentionally are not the same phenomenon" [3]. Thus, intentions are classified into two types:

- A *Future-oriented* intention is a specific instance of an applicable plan.

[1] Note that *_self* in the following plan refers to the agent itself.

- A *Present-oriented* intention is the future-oriented intention that the agent has chosen to pursuit.

Actions are one of the components of an intention. An intention may contain several actions that will sequentially be performed. Each action describes the agent's behavior and the action conditions. An action is composed of three components corresponding to three situations:

- *Normal* situation: Normal situation is a situation where the mental condition of the agent meets the intention condition that is inherited from the plan context. At this moment, the action is normally performed.
- *Success* situation is a situation where agent's mental condition allows the agent to decide that certain goals are achieved (for example, the agent acquires new desires that allow the agent to satisfy the desires of its intention goals).
- *Failure* situation is a situation where the agent considers that the action failed and where it decides to stop following the action goal. In this case, the agent removes the corresponding desires or creates new events that trigger the backup plans.

3.3 Practical reasoning processes

Phase 1: Perception. The agent either perceives a change in its environment or receives a message from other agents (processing flow 1). The agent then subjectively considers this new information to select what it finds useful by using a user-programmed filter. The filtered information is transferred into its beliefs. For example, in the context of a forest fire simulation, a helicopter agent may simultaneously perceives burning and intact trees; in the this simulation context, intact trees are not important, so they are ignored; in contrary, the burning trees are important, so the beliefs on these trees (for example: tree12, tree13) are created as follows: *Bel(tree12, burning, _null); Bel(tree13, burning, _null).*

Phase 2: Update beliefs and desires. In this phase, the agent executes the logic deduction on the current beliefs and desires (including the new incoming beliefs) by applying the rules to create new beliefs and desires (processing flow 2). Once new beliefs or desires are created, agent adds the corresponding events into the event base.

At the end of this phase, the agent obtains new beliefs (added to the belief base), desires (added to the desire base) and events that will be used in next phase.

Phase 3: Update events. In this phase, the agent analyses the events created in the previous phase (processing flow 3) in order to choose a suitable behavior according to the current context. The idea here is that the agent only needs to elaborate a new plan if the changes from its world impacts what is already planned. So instead of losing time to choose again the same plan, the agent continues what it is doing.

The agent begins planning only when there are new interesting events. An event is considered interesting when it matches one of the abstract events described in the trigger part of the plan. The events can be of two types: belief events and desire events. The filtering of events is thus separated into two steps.

Step 1: Verify interesting events on beliefs. The agent checks whether it perceives any interesting events of belief. If there is no belief event considered as interesting, the agent considers that there is no change from its environment or that the changes are not important enough to force it to reconsider its situation. In this context, the agent executes the current executable intentions (processing flow 4). The fact that the agent ignores the thinking process to continue pursuing its intention is called *mime mode*, which means that the agent repeats what it was already doing. In contrary, if the agent perceives some interesting belief events, it moves to the next step to verify the desire events (processing flow 5).

Step 2: Verify interesting events on desires. In this step, the agent checks whether there are any interesting events of desire or not. If all the desire events are considered as not interesting, the agent keeps following the same goals and continues to select the future-oriented intentions, which are deliberated according to the goals that were previously chosen (processing flow 6). We call the way the agent continues to deliberate intentions the persistent mode, because the agent persistently pursuits its pre-decided goals.

In case the agent finds that some of the desire events are interesting, it moves to next phase to decide what to do (processing flow 7).

Phase 4: Decision-making. In this phase, the agent makes its decisions so that its next behaviors are compatible with the current conditions of its mental state (which are corresponding to its environment modification).

Concerning the decision-making process, previous works have proposed plenty of approaches on how an agent decides what to do. For example the approach proposed by Bordini (in AgentSpeak) [2] states that agent's planning phase is based on the intention.

In our architecture, we propose another approach: the agent establishes its plan according to its desires. The planning phase only consists in creating the future-oriented intentions. The actions are only created when the context condition of the intention is met. Thus, the decision in our model is separated into two steps.

Step 1: Selection of desires.

From its desire base, the agent selects a sub-set of desires according to their competing category and priority. First, the agent takes all non-competing desires. Then, among the desires of a category, it selects only the ones that have the highest priority.

Step 2: Select applicable plans.

The agent selects the plan of which the conditions are met according to the agent's mental states (processing flow 8). The plans are selected in the following way:

- The agent only selects the plans corresponding to the desires chosen from the previous step.
- The agent uses the objects and the subjects from all the beliefs and desires in its bases to apply to the variable of the abstract conditions of each plan.
- For each instance of the plan, the agent verifies all the conditions (including the context condition and the triggering condition)
- An instance of the plan of which all the conditions are satisfied is considered as applicable. The agent then uses it for the next phase (processing flow 9)

Phase 5: Deliberate intentions. In this phase, the agent organizes its intentions to select the current executable intention to perform. First, the agent creates the future-oriented intentions from the applicable instances of plans. The intentions, inheriting the goals, the context and the actions from the plan, are put into a structured base. In this base, these intentions are grouped according to their desire competing categories. Then, the agent chooses the intentions which conditions are currently satisfied according to the belief base (processing flow 10). The satisfied intentions are also called present-oriented intentions and are put into the executable intention base for the next phase (processing flow 11).

Phase 6: Execute intentions. In this phase, the agent performs one by one the actions of the selected intentions in order to react to its environment, and then it verifies the predefined situation condition of each action. If the success condition is satisfied, the agent performs the behaviors for success situation. If the failure condition is met, the agent executes the behaviors for failure situation. In both cases, the intention is removed from the base. In the normal situation, the agent keeps the intention into its base for the next step of the simulation.

4 Application and discussion

4.1 Implementation and evaluation

In this section, we present an application of our architecture on a simulation of forest fire. In the simulation, we model the spreading of fire in a forest of 250x150 m^2 size in which there are 1000 agents representing the tree (which have the 3 states of the real tree in case of fire: intact, caught fire, dead), 2 agents representing the firefighter helicopters (which have the 3 simple actions: patrolling, extinguish fire, recharge water) and 1 agent representing the base where the helicopters recharge the water.

In order to evaluate our architecture, we built two strategies: one based on our new BDI architecture and another one based on simple reactive behaviors. For the BDI-using strategy, we built the plan base and the rule base so that firefighter helicopter could pass the maximum trees instead of moving to the nearest burning tree as described in the simple strategy.

The chart in Fig.2 shows that the BDI-using strategy is more effective than the reactive one to extinguish fires, which shows the usefulness of our proposed BDI architecture to make betters strategies. In fact, this result chart comes from the average result of 100 testing times of simulations for both strategies. Most of the simulations return a stable result which shows that the performance of BDI-based strategy is approximately two times better than the reactive strategy.

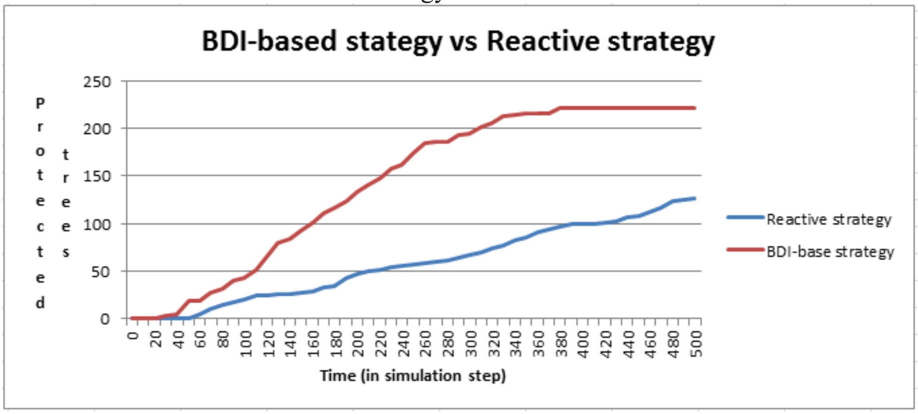

Fig. 2. Result of testing BDI-based strategy and Reactive strategy

4.2 Discussion

In addition to allow modelers to build rich and effective behaviors, our architecture also allows them to easily modify the agent behavior:

- The structure of the plan permits to easily build plenty of backup plans. Modelers only need to define various plans with the same goal and each plan can have distinct context conditions. The reasoning phase would help the agent to select suitable plans according to its specific situation.
- The agent behavior strategy is based on a pre-defined plan base and on a rule base. Thus, modeler can easily fix agents actions, and then change these two bases to test different strategies.

5 Conclusion

In this paper, we propose a new cognitive agent architecture based on the BDI paradigm. Our architecture allows defining easily cognitive agents able to make complex reasoning. We illustrated the use of our architecture through a simple model dealing with forest fire and firefighter helicopters.

Although our framework have been tested on a simple simulation, the way we represent the knowledge and the decision-making process proposes promising extensions so that our architecture would adapt to other problems. An example of theoretical extension is that the decision phase will be made by considering not only the importance of the desire but also the urgency level, which will make the architecture adapt to the other simulations of urgent situation.

6 References

1. Adam, C., Gaudou, B., Hickmott, S., Scerri, D. : Agents BDI et simulations sociales, Unis pour le meilleur et pour le pire. Revue d'Intelligence Artificielle, Special issue: Simulation sociale orientée agent, Hermès Science Publications, Vol. 25, p. 11-42, 2011
2. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak Using Jason. Wiley Series in Agent Technology. John Wiley & Sons, 2007
3. Cohen P.R., Levesque H.J: Intention is choice with commitment. In Journal Artificial Intelligence archive, Volume 42 Issue 2-3, March 1990, Elsevier Science Publishers Ltd. Essex, UK
4. Bratman, M.E., Israel, D.J., Pollack, M.E.: Plans and resource-bounded practical reasoning. Computational Intelligence, 4(4):349-355, 1988
5. Bratman, M.E.: Two Faces of Intention. The Philosophical Review, Vol. 93, No. 3 (Jul., 1984), pp.375-705
6. DeAngelis D., Gross L., Individual-Based Models and Approaches in Ecology, Chapman and Hall, New York, 1992
7. Gilbert N., Troitzsch K. G., Simulation For The Social Scientist, Open University Press, 2005
8. Huber, M.J.: JAM: A BDI-Theoretic Mobile Agent Architecture. Agents (1999) 236-243
9. Wilensky U.: Netlogo. Technical report, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999
10. Myers, K.L.: User Guide for the Procedural Reasoning System Technical Report. Artificial Intelligence Center, Technical Report, SRI International, Menlo Park, CA, 1997
11. Rao, A.S., Georgeff, M.P.: Modeling Rational Agents within a BDI-Architecture. (1991) 473-484
12. Robert, G., Guillot A.: A motivational architecture of action selection for non-player characters in dynamic environments, In International Journal of Intelligent Games & Simulation, vol. 4, pp. 1-12 (2005)
13. Shendarkar, A., Vasudevan, K., Lee, S., Son, Y.J.: Crowd simulation for emergency response using BDI agents based on immersive virtual reality. Simulation Modelling Practice and Theory 16 (2008) 1415–1429
14. Taillandier, P. and O. Therond.: Use of the Belief Theory to formalize Agent Decision Making Processes: Application to cropping Plan Decision Making, in *ESM*, pp. 138-142, Guimaraes, Portugal, 2011
15. Taillandier, P., Drogoul, A., Vo, D.A.: GAMA: a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In The 13th International Conference on Principles and Practices in Multi-Agent Systems (2012)
16. Tesfatsion L.: Agent-based computational economics: growing economies from the bottom up, Artificial Life, vol. 8, p. 55-82, 2002