



HAL
open science

A new BDI agent architecture based on the belief theory. Application to the modelling of cropping plan decision-making

Patrick Taillandier, Olivier Therond, Benoit Gaudou

► To cite this version:

Patrick Taillandier, Olivier Therond, Benoit Gaudou. A new BDI agent architecture based on the belief theory. Application to the modelling of cropping plan decision-making. International Environmental Modelling and Software Society (iEMSs), 2012, Leipzig, France. hal-00714325

HAL Id: hal-00714325

<https://hal.science/hal-00714325v1>

Submitted on 4 Jul 2012

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.

A new BDI agent architecture based on the belief theory. Application to the modelling of cropping plan decision-making

Patrick Taillandier^a, Olivier Therond^b and Benoit Gaudou^c

^aUMR IDEES, CNRS – Université de Rouen, MTG; ^bINRA, UMR 1248 AGIR F-31326 Castanet Tolosan ; ^cUMR 5505 IRIT, CNRS – Université de Toulouse 1
patrick.taillandier@univ-rouen.fr, olivier.therond@toulouse.inra.fr,
benoit.gaudou@univ-tlse1.fr

Abstract: Agent-based simulations are now widely used to study complex systems. However, the problem of the agent design is still an open issue, especially for social-ecological models, where some of the agents represent human beings. In fact, designing complex agents able to act in a believable way is a difficult task, in particular when their behaviour is led by many conflicting needs and desires. A widely used way to formalise the internal architecture of such complex agents is the BDI (Belief-Desire-Intention) paradigm. This paradigm allows to design 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 by non-computer-scientists. Moreover, they are often very time-consuming in terms of computation. In this paper, we propose a new architecture based on the BDI paradigm that copes with these two issues. In our architecture, the choice of the most relevant action by an agent is based on the belief theory. We present an application of our agent architecture to an actual model dedicated to cropping plan decision-making. This application that takes into plays thousands of farmer agents shows promising results.

Keywords: agent-based modelling; BDI architecture; belief theory; cropping plan decision-making.

1 INTRODUCTION

The study of social-ecological systems is a major issue for our society. It has many applications such as risk assessment, country planning or management of natural resources (e.g. (MAELIA, 2012)). Carrying out such studies is most of time very complex. In fact, these systems involve complex spatial and social dynamics. One of the most promising modelling techniques to study the dynamic of such systems is the agent-based modelling and simulation. This technique consists in modelling the studied system as a collection of entities called agents representing both inactive resources and autonomous decision-making actors. An autonomous agent can individually assess its situation and makes its own decisions. An agent-based model can exhibit complex behaviour patterns and provide useful information about the dynamics of the real-world system that it emulates. These last years have seen the emergence of several platforms allowing to ease the development of agent-based models. However, even with these platforms, the problem of the agent design is still an open issue. In fact, designing agents able to make complex reasoning and to interact with its biophysical environment and other agents is a difficult task, in particular for researchers that have no programming skills.

In this paper, we propose a new cognitive agent architecture that is particularly well-fitted for simulation of complex systems with numerous autonomous agents. This architecture is based on the BDI (Belief-Desire-Intention) paradigm and the use of the evidence theory to formalise the agent reasoning.

2 EXISTING AGENT ARCHITECTURES

The problem of the agent design and implementation is a classic problem in multi-agent simulations and numerous agent architectures were proposed. Some of these architectures such as the finite state machines or the motivational architecture (Robert & Guillot, 2005) 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 behaviour of more complex cognitive agents is the BDI paradigm (Rao & Georgeff, 1991). Some works showed the interest of using such paradigm in simulation context (e.g. Adam et al., 2011), yet, it is still rarely used. A reason is that most agent architectures based on the BDI paradigm (e.g. JAM (Hubert, 1999), JASON (Bordini et al., 2007)) are complex to understand and to use by non-computer-scientists. Moreover, they are often very time-consuming in terms of computation and thus not adapted to simulations with thousands of agents.

3 ARCHITECTURE PROPOSED

3.1 General Architecture

We propose a new agent architecture based on the BDI paradigm that copes with two issues mentioned in Section 2: the understandability aspect and the scalability problem. Our architecture is composed of four databases (Figure 1):

- **Plans:** the agent's strategic vision. The plans define the strategies that can be followed by the agent. A plan is composed of the following components:
 - *Action set:* defines the action that can be applied while considering the plan.
 - *Action application rules:* define the conditions to apply the different actions.
 - *Plan update rules:* define when and how the plan has to be updated (or deleted from the intention base).
- **Desires:** agent's desires. Desires are formalised as a set of criteria that will be used to evaluate the plans. Section 3.2 gives more details about the desires.
- **Beliefs:** agent's beliefs about the system functioning. They are used to compute the values of the desires (criteria). Examples are given in Section 4.1.
- **Intention:** the chosen plan.

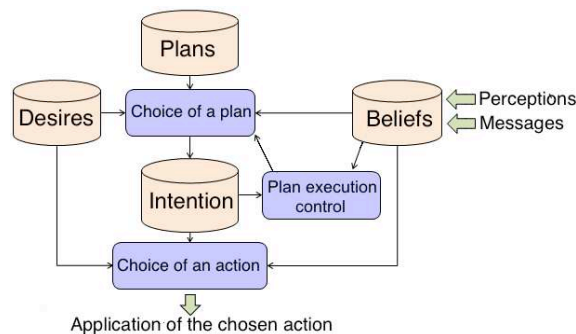


Figure 1. Agent architecture

When an agent acquires new information (by its own perception or by a message sent by another agent), it automatically updates its belief base. When an agent has no chosen intention (empty intention base), it evaluates each plan of its plan base according to its desires and beliefs. Then the agent selects a plan through a multi-criteria decision-making process. The selected plan is then added to its intention base. The agent will then continuously choose among the actions proposed by the plan the one that is the most relevant according to its context (defined by its beliefs and desires). At each simulation step, the plan can be deleted or updated through a plan execution control process.

In the next sections, we describe the 3 main decision processes: *choice of a plan*, *choice of an action* and *plan execution control*.

3.2 Choice of a plan

The choice of a plan is carried out through a multi-criteria decision making process. In order to decrease the problem complexity, we propose to filter the set of plans by keeping only the ones belonging to the Pareto front, then to apply a decision-making method. The Pareto front is the set of plans that are non-dominated. A plan is dominated by another one if no desire (criterion) has a better value (higher or lower depending on the type of desire) for this plan than for the other one.

In the literature, several methods were proposed to make a decision from a set of criteria. A first family of methods, called *partial aggregation* methods, consists in comparing the different possible decisions by pair by the mean of outranking relations (e.g. Behzadian et al. 2010). Another family of methods, called *complete aggregation* methods, consists in aggregating all criteria in a single criterion (utility function), which is then used to make the decision (e.g. Jacquet-Lagrange and Siskos 1982). *Partial aggregation* methods allow to address the problem of criterion incompatibility but lack clarity compared to *complete aggregation* methods (Ben Mena 2000). In our context, we propose to use the same decision-making method than the one we proposed in (Taillandier & Therond, 2011; Taillandier et al., 2012). This method belongs to the *complete aggregation* methods and is built on the belief theory (Shafer, 1976). An advantage of this theory is to allow to make a decision even with incompleteness, uncertainty and imprecision of knowledge.

The belief theory defines a *frame of discernment*, noted Θ , composed of a finite set of hypotheses H_i corresponding to the possible plans amongst which the choice has to be made:

$$\Theta = \{H_1, H_2, \dots, H_N\}$$

From this frame of discernment, we define the set of all possible assumptions, noted 2^Θ :

$$2^\Theta = \{\emptyset, \{H_1\}, \{H_2\}, \dots, \{H_1, H_2\}, \dots, \Theta\}$$

Each set $\{H_i, \dots, H_j\}$ represents the proposition that the most relevant hypothesis is one of the hypotheses (plans) of this set.

The belief theory is based on basic belief assignments, i.e. functions that assign to a proposition P , with $P \in 2^\Theta$, a value named the basic belief mass (*bbm*), noted $m_j(P)$. It represents how much a criterion j -called source of information- supports the proposition P . The *bbm* is ranged between 0 and 1 and has to check the following property:

$$\sum_{P \in 2^\Theta} m_j(P) = 1$$

The proposed decision-making method is composed of 3 steps:

1. Initialization of the basic belief masses.
2. Combination of the criteria.
3. Selection of the best plan.

Step 1- Initialization of the basic belief masses

The first step consists in initializing the basic belief masses for each criterion. The criteria give agent's opinion in favour of a plan, in disfavour of it or do not give its opinion. For each plan, a subset S^j of 2^Θ is defined:

$$S^j = \{\{H_i\}, \{\neg H_i\}, \emptyset\}$$

- $\{H_i\}$: this proposition means that the hypothesis H_i is true (i.e. the corresponding plan is the best one).
- $\{\neg H_i\} = \Theta - \{H_i\}$: this proposition means that the hypothesis H_i is not true (i.e. at least one plan is better than this one).
- \emptyset : this proposition means the ignorance (i.e. the plan cannot be compared to the other ones).

Thus, the initialization of the basic belief masses consists in computing, for each criterion and for each plan, the basic belief masses (*bbm*). To compute all the *bbm*, belief functions have to be defined. A belief function is a function that returns a float

value between 0 and 1 according to the value of a considered criterion for a given hypothesis. Examples of belief functions are given in Section 4.1.

Step 2- Combination of the criteria

This step consists in combining criteria with each other. We use the conjunctive operator introduced in (Smets and Kennes 1994) to provide a combined *bbm* synthesizing the knowledge from the different criteria. Let us consider two criteria C_1 and C_2 . The conjunctive operator is defined as follows:

$$\forall H_i \in \Theta, \forall P \in \{\{H_i\}, \{\neg H_i\}, \Theta\}, m_{C_1 C_2}^{H_i}(P) = \sum_{P' \cap P'' = P} m_{C_1}^{H_i}(P') \times m_{C_2}^{H_i}(P'')$$

This conjunctive operator is commutative and associative. Thus, it is possible to combine the result of a previous fusion with the belief masses of another criterion.

Step 3- Selection of the best plan

The last step consists in making the decision. We use the *pignistic* probability (Smets 1990) to evaluate each proposition. The *pignistic* probability of a proposition A is computed by the following formula:

$$P(A) = \sum_{A \subseteq B} m(B) \frac{|A|}{|B|}$$

This probability represents the utility of the plan: the selected plan will be the one that maximizes this probability.

Definition of a desire in our BDI architecture

In our BDI architecture, a desire is then defined by:

- A function that computes the criterion (desire) value according to agent beliefs.
- A belief function that evaluates the hypothesis that a plan is the best one according to the desire value.
- A belief function that evaluates the hypothesis that a plan is not the best one according to the desire value.
- A belief function that evaluates the ignorance according to the desire value.

3.3 Choice of an action

As mentioned in Section 3.1, a plan contains a set of actions and a set of rules that defines how to select the best action according to the agent beliefs and desires. This process consists in applying, according to the set of action application rules, the most pertinent action proposed by the plan.

3.4 Plan execution control

This process consists in determining if the plan has to be deleted from the intention base or updated. The plan contains rules defining, according to the agent beliefs, if the plan has to be deleted from the intention base. In the same way, it contains rules defining, according to the agent beliefs, if the plan has to be updated and how.

4 CASE-STUDY: CROPPING PLAN DECISION-MAKING

In this section, we present an application of our architecture to a real model concerning cropping plan decision-making. In Section 4.1, we present our model following the first part of the ODD protocol (Grimm et al., 2006). Section 4.2 describes a first experiment carried out with our model.

4.1 Description of the model

Overview

Purpose

The model is part of the MAELIA project (MAELIA, 2012) aiming at developing a platform for the simulation of the socio-environmental impacts of water management norms on the water resources. In particular, this project proposes to model the impacts of norms on the farmer behaviours that are in many regions the most important water users in low water period. Our model aims at simulating the behaviours of farmers in their choice of crops and their day-life activities.

Entities, state variables, and scales

Two main types of entities are in the model: the *farmer* (autonomous) agents and the *field* agents.

A *farmer* agent has the following state variables:

- *fields*: a list of fields
- *plans*: a list of the possible crop rotation–field combinations
- *intention*: the current crop rotation for each field
- *desires*: a list of desire (defined as described in Section 3.2).
- *beliefs*: information used to make decisions. It is composed of:
 - *last_production_memory*: production for each type of crop
 - *last_weather*: weather of the five last days (rain, temperature evapotranspiration) on each field
 - *crop_prices*: current cropping market price
 - *crop_costs*: cost of the different crop managements
- *water*: quantity of water available
- *money*: financial resources

A *field* agent has the following state variables:

- *shape*: spatial geometry (georeferenced)
- *is_irrigable*: boolean
- *current_crop*: current crop and crop management
- *weather*: weather of the day (rain, temperature, evapotranspiration)
- *water_irrigation*: quantity of irrigation water

In the model, one time step represents one day. This time-step is necessary to take into account the daily activity of the farmers (that has a direct impact on the water resource), which is directly impacted by the daily weather and the government water crisis decrees (e.g. interdiction to use water for several days). The *farmer* agents make their cropping plan decision only once a year.

Process overview and scheduling

The *field* agents have only one behaviour: compute the quantity of crop produced. For the non-irrigated crops, the computation is done by drawing a Gaussian random number according to the observed crop yield distribution. For the irrigated crops, the computation is done thanks to a simple plant growing model that depends on the weather and on the crop irrigation.

Concerning the behaviour of the *farmer* agent, we used the architecture described in Section 3. For the *farmer* agent, a **plan** represents a complete assignment of crop rotations to its fields. *Farmer* agents have to make a choice amongst a set of 10 different crop rotations (e.g. Durum wheat – Sunflower – Durum wheat – Rape). A plan is then defined by:

- *Set of actions*:
 - *Sowing*: updates the *current_crop* attribute of a *field* agent and the *money* attribute of the *farmer* agent.
 - *Irrigating*: updates the *water_irrigation* attribute of a *field* agent and the *water* attribute of the *farmer* agent. In this model, irrigation is costless.

- Harvesting: updates the *current_crop* attribute of a *field* agent and the *money* attribute of the *farmer* agent.
- *Action application rules*: if it is possible to sow (good moment of the year for the considered crop -for instance, between the 10th of April and the 15th of May of maize- not too much rain during the last two days), the *farmer* agent sows, otherwise, it tries to harvest a field. If it is not possible to harvest (not the good moment of the year -for instance, between the 1st of October and the 15th of December for maize- or too much rain during the last two days), it tries to irrigate a field. The model takes into account the time taken by each type of field operations. A *farmer* agent is able to carry out several actions in a day defined as a set of work hours. After finishing a task on a field, a *farmer* agent will tend to work on fields close to the previous one (computed thanks to the field georeferenced shape).
- *Plan update rules*: if it was not possible to sow a crop in a field during the sowing season, the farmer agent can update its plan by replacing the crop by a substitute one. Moreover, each time the rotation of crops for a field is finished, the plan is removed from the intention base and a new choice of plan is triggered.

Concerning the **desires** of the agent, we defined 4 desires (criteria) based on the work of (Dury et al., 2010):

- Maximise the expected profit
- Minimise the financial risks
- Minimise the workload
- Maximise the similarity to the last cropping plans

Maximise the expected profit

A farmer tends to select a plan that potentially maximises its profit. The profit takes into account the price of cropping market price and the cost of the different variable costs (seeds, fertiliser...). The belief functions are shown in Figure 2. These functions depend on the profit expected over the years of the plan (P). P_{max} is the maximal profit obtained with the different plans. D_{max} is the maximal deficit that can be made considering the worst scenario (no plant grown).

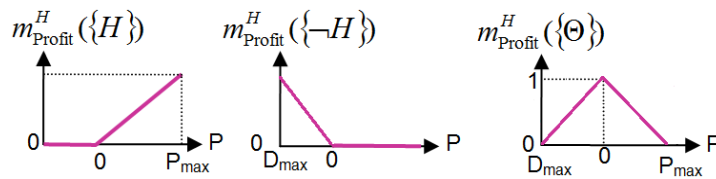


Figure 2. Belief functions for the “maximise profit” desire

Minimise the financial risks

A farmer tends to avoid plans that can lead to a high variation of the expected financial outcomes. The belief functions are shown in Figure 3. These functions depend on the standard deviation of the expected outcomes according to the quantity of crops produced (R). R_{max} is the maximal standard deviation obtained with the different plans.

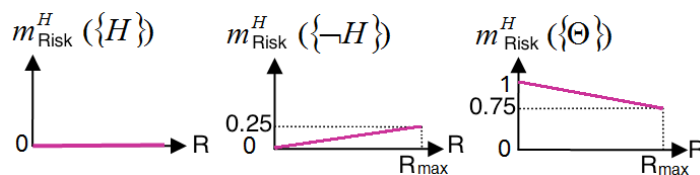


Figure 3. Belief functions for the “minimise risk” desire

Minimise the workload

A farmer tends to select a plan that maximises the number of free days (days without work). The belief functions are shown in Figure 4. These functions depend on the number of free days (F).

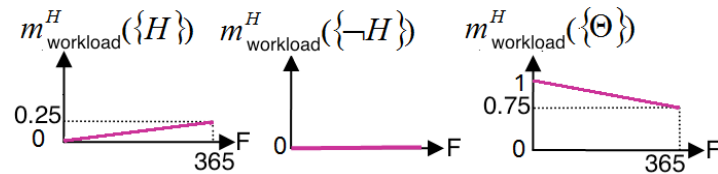


Figure 4. Belief functions for the *minimise workload* desire

Maximise the similarity to the last cropping plans

A farmer tends to always keep the same plan over the years, or at least similar plans. The belief functions are shown in Figure 5. These functions depend on the similarity value compared to the last cropping plan (S), i.e. the rate of similar cropping choices.

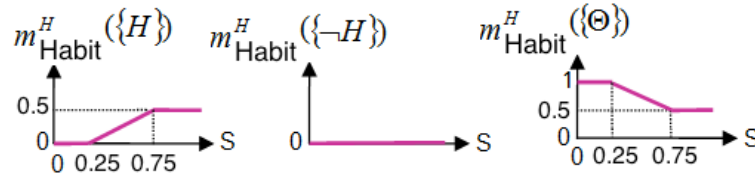


Figure 5. Belief functions for the *maximise similarity to last cropping plan* desire

The **belief** base of a *farmer* agent is composed of the following attributes: *current_cropping_plan*, *last_production_memory*, *last_weather*, *water*, *crop_prices* and *crop_costs*

Details

Input

The data used as input are the following ones:

- The Land Parcel Identification System (LPIS) to 2009: describe the geometries (GIS) of the fields and whether they are irrigable.
- Rotation CSV data (2006-2009): describe the rotation of crops for each field and the owner of the field (provided by processing of LPIS 2006 to 2009).
- Weather data (2000-2010): describe the rain, temperature, evapotranspiration for each day.
- Crop price/cost data (2000-2007): describe crop prices/costs for each year.
- Crop information data: describe the characteristic of the different crops.

4.2 Experiments

Context

The model was implemented with the GAMA simulation platform (Taillandier et al., 2012). The tests were carried out with a laptop computer running under Mac OS X, with an i7 processor and 4Go of RAM. We carried out an experiment aiming at simulating the cropping plan decision making of the farmers of two water basins (landscape size: 125×100 km) between 2005 and 2009 (1825 simulation steps). The experiment integrates more than 2700 *farmer* agents and 20 700 *field* agents. The first goal of this experiment was to test the scalability of our architecture. The second goal was to test the relevance of the built model. We do not use data to initialize the *current_crop* attribute of the *field* agent (set to *null*): the rotation data were only used to validate the model.

Results

A first result of this experiment concerns the scalability of our architecture: the 1825 simulation steps took less than 22 minutes, which is rather promising given the laptop used and that we did not try to optimise computation times of the model. A second result concerns the relevance of the model. For each year, we compare the percentage of field area occupied by each crop type in the real data and in the simulation results; the mean similarity over the five years between both data is equal to 76,5%. This result is rather promising knowing that in our experiment only

one profile of farmers was used. In further experiments, we plan to define several profiles to improve the simulation results. At last, in terms of understandability, the architecture proved to be simple enough to be understood by domain-experts to allow them to participate to the definition of the *farmer* agent behaviour (in particular, the belief functions).

5 CONCLUSION

In this paper, we propose a new BDI architecture, based on the belief theory, dedicated to cognitive agents. We present an application of this formalism to a simulation dedicated to cropping plan decision-making. First experiments carried out showed that our architecture allows to simulate thousands of agents in a reasonable amount of time. Moreover, they showed that the model built with our architecture allows to obtain interesting results, close to the real system results. The next step of the MAELIA project will consist in coupling this model with a hydrological model and enriching it with new dynamics and agents such as a *government* agent (able to create norms), the land market or technology evolution.

REFERENCES

- Adam C., B. Gaudou, S. Hickmott and D. Scerri, Agents BDI et simulations sociales. Unis pour le meilleur et pour le pire, in *RIA*, pp. 11-43, 2011.
- Behzadian, M., R. Kazemzadeh; A. Albadvi and M. Aghdasi. PROMETHEE: A comprehensive literature review on methodologies and applications. *EJOR* 200(1), 198-215, 2010.
- Ben Mena, S. Introduction aux méthodes multicritères d'aide à la décision, *Biotechnol. Agro. Soc. Env.* 4(2). 83-93, 2000.
- Bordini R. H., J.F. Hübner and M. Wooldridge, Programming Multi-Agent Systems in AgentSpeak Using Jason", in *Agent Technology*, John Wiley & Sons, 2007.
- Dempster, A. Upper and lower probabilities induced by multivalued mapping". *An. of Math. Statistics* 38, 325-339, 1967.
- Dury, J., F. Garcia, A. Reynaud, O. Therond and J.E. Bergez, Modelling the Complexity of the Cropping Plan Decision-making", *IEMS*, Canada, 2010.
- Grimm, V. et al.: A standard protocol for describing individual-based and agent-based models. *Ecological Modelling* 198 (1-2), 115-126, 2006.
- Huber M.J., JAM: A BDI-Theoretic Mobile Agent Architecture, in *Agents*, pp. 236-243, 1999.
- Jacquet-Lagrece, E. and J. Siskos, Assessing a set of additive utility functions for multicriteria decision making, the UTA method. *EJOR* 10(2), 151-164, 1982.
- MAELIA project, <http://maelia1.wordpress.com/> [Accessed 1 February 2012], 2012.
- Rao A.S., and M.P. Georgeff, Modeling Rational Agents within a BDI-Architecture, in *KR'91*, pp. 473-484, 1991.
- Robert G., and A. Guillot, A motivational architecture of action selection for non-player characters in dynamic environments, in *IJIGS*, vol. 4, pp. 1-12, 2005.
- Shafer, G. 1976. *A mathematical theory of evidence*. Princeton University Press.
- Smets, P. and R. Kennes, The transferable belief model. *Artificial Intelligence* 66(2) : 191-234, 1994.
- Smets, P. Constructing the pignistic probability function in a context of uncertainty. *Uncertainty in Artificial Intelligence* 5: 29-39, 1990.
- 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.
- Taillandier P., A. Drogoul, D.A. Vo and E. Amouroux, GAMA : a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control, in *PRIMA*, pp. 575-587, Kolkata, India, 2012.
- Taillandier P., E. Amouroux, D.A. Vo, and A.M. Olteanu-Raimond, Using Belief Theory to formalize the agent behavior : application to the simulation of avian flu propagation, in *PRACSYS*, pp. 242-258, Kolkata, India, 2012.