Designing agent behaviour in agent-based simulation through participatory method

Patrick Taillandier, Elodie Buard

Designing Agent Behaviour in Agent-Based Simulation through Participatory Method

Patrick Taillandier[1,2] and Elodie Buard[3,4]

[1] IRD, UMI UMMISCO 209,
32 avenue Henri Varagnat, 93143 Bondy, France
[2] IFI, MSI, UMI 209,
ngo 42 Ta Quang Buu, Ha Noi, Viet Nam
patrick.taillandier@gmail.com
[3] COGIT IGN, 2/4 avenue Pasteur,
94165 Saint-Mandé, France
[4] UMR Géographie Cités,
13 rue du Four, 75006 Paris, France
elodie.buard@ign.fr

Abstract. Agent-based simulation has demonstrated its usefulness for the modelling of complex systems. However, the simulation widely depends on the agent behaviour designing. In order to facilitate the definition of such behaviour, we propose an approach based on a participatory method: a domain expert directly enters his knowledge about entities in a specific environment. In this paper, we propose to formalise the agent behaviour by using a combination of production rules and of a multi-criteria decision making method. An experiment, carried out in the domain of ecological simulation, is presented. This first experiment shows promising results for our approach.

Keywords: multi-agent simulation, agent behaviour design, participatory method, multi-criteria decision making, ecological simulation.

1 Introduction

Agent-based simulations are powerful tools to study complex systems. Indeed they enable to take into account not only different level of granularity but also the heterogeneity of the entities composing the system. A key point of these simulations concerns the designing of the agent's behaviour. Unfortunately, this task can be complex and fastidious. A recent approach to face this difficulty in the context of agent-based simulation consists in using participatory methods [6]. These methods propose to let human actors (e.g. domain experts) directly participate in the agent modelling process through their interactions in the simulation. In this paper, we propose an approach dedicated to the agent behaviour design based on a participatory method. Thus, we propose to let a human expert plays the role of an agent in the simulation and to analyse the logs produced in order to extract knowledge about the expert behaviour. In Section 2, the context of our work is introduced. Section 3 is

J.-J. Yang et al. (Eds.): PRIMA 2009, LNAI 5925, pp. 571–578, 2009.

devoted to the presentation of our agent behaviour design approach. Section 4 describes an application of our approach in the domain of ecological simulations. Section 5 concludes and presents the perspectives of this work.

2 Context

In this paper, we deal with agent behaviour design. For numerous complex systems, designing such behaviour is a complex task, in particular, when no formalised knowledge is available on the system studied. In this context, it becomes necessary to directly obtain the necessary knowledge from domain experts. Unfortunately, extracting this knowledge is often a fastidious and difficult task. To face this difficulty, an approach consists in using Machine Learning techniques. Indeed, these techniques enable to acquire general knowledge from examples. It is then possible to obtain knowledge about the agent behaviour by analysing examples of their behaviour. Two questions arise from the use of such approach: which data to use and how to learn the agent behaviour from these data. An approach to answer these questions consists in using the participatory paradigm. Based on this approach, [4] and [10] propose methods to learn expert decision criteria in the context of rescue simulation. More precisely, they propose methods to learn utility functions representing the agent behaviour by means of interactions between an expert and the system. In [4], the expert observes the behaviour of an agent and has the possibility to correct an action if not relevant. The agent takes into account this intervention to refine its behaviour. In [10], the expert directly takes the control of an agent. Here, the behaviour is learnt *a posteriori* in analysing the produced logs. In this paper, we propose to use the same general approach as in [10]. Thus, the agent behaviour is learnt in analysing the expert activities when this one takes the control of an agent. However, we refine the formalism representing the agent behaviour to get a more powerful tool. Indeed, in contrary to [4] and [10] where a weighted linear combination of criteria was used to represent the agent behaviour, we propose to use a combination of production rules and of a powerful multi-criteria decision making method. This new formalism brings about more complex and reliable agent behaviours. In this context, we put forward a new approach to learn the agent behaviour which is described in the next part.

3 Approach Proposed

3.1 General Agent Behaviour Design Approach

In this part, we carry out the designing of agent behaviours in multi-agent simulations. We define the agent behaviour as the function that the agent uses to choose, at each step of the simulation process, an action to apply between a set of possible actions. As stated in the introduction, we propose to learn this function by using a participatory method: an expert directly plays the role of an agent in the simulation and the behaviour of this one is learnt by analysis of the expert behaviour. Our general approach is composed of two stages: the first one consists in producing data concerning the behaviour of the human expert, the second one in analysing these data in order to learn about the expert behaviour. The method used for this second stage deeply depends on the formalism used to represent the agent behaviour. In Section 3.2, we describe the data gathering in the acquisition stage. As for the learning stage, in Section 3.3, we present the formalism we used to represent the agent behaviour and in Section 3.4 the learning method we propose.

3.2 Production of Data Concerning the Expert Behaviour

This stage, which consists in producing data representing the expert behaviour, is similar as the one presented in [10]. In order to produce these data, we propose to use a participatory method: an expert directly interacts in the simulation by playing the role of an agent. At each step of the simulation process, the action he chooses is logged. The actions can be of different types. Typically, some actions can refer to displacement toward an objective, other to communications, other to the modification of the environment, and so on. We propose to characterise the actions of each type by sets of criteria. We assume that, for each type of actions, a set of criteria is defined to characterise the actions of this type and that a set of criteria is defined to characterise the state of the agent.

3.3 Formalisation of the Agent Behaviour

In this paper, we propose to solve the action choice problem by decomposing it into two sub-problems. Indeed, instead of directly choosing the action that is the most relevant between the complete set of actions, the agent will choose, in a first step, the type of action that is relatively more appropriate in regards with its state, then, in a second step, the action of this type that is the most relevant. The interest of this decomposition is to reduce the complexity of the agent behaviour design problem.

Concerning the action type choice problem, we propose to formalise the knowledge used to make this choice by a set of production rules. The advantage of this kind of representation is to be easily interpretable by domain experts and thus to facilitate its validation and update.

For the choice of the most relevant of the action type considered, we propose to formulate the problem as a multi-criteria decision making problem. The goal is to choose, according to criteria specific to the selected action type, the action that is the most relevant. In the literature, numerous approaches are proposed to solve this type of multi-criteria decision making problems. In this work, we propose to use a method based on the ELECTRE methods [8]. The ELECTRE methods are well-established multi-criteria decision making methods based on partial aggregation. They enable to make a decision from incomparable criteria [2]. These methods were used with success to solve numerous problems [9]. Their principle is to compare the possible actions by pair. These methods analyses the possible outranking relation (noted S) existing between two actions. An action outranks another if this one is at least as good as the other one.

Our decision making method requires defining several parameters for each criterion:

- The *weight* of the criterion: importance of the criterion in the action choice.
- The *preference threshold*: represents the threshold from which the difference between two criterion values allows to prefer one action over another.
- The *indifference threshold*: represents the threshold from which the difference between two criterion values is considered significant.
- The *veto threshold*: represents the threshold from which the difference between two criterion values disqualifies the action that obtained the smaller value.

A last parameter to define is the cutting level of the fuzzy relation λ. It defines the reference threshold for the action comparison. The higher this threshold, the more the establishment of the relation A_1SA_2 requires unanimity from the criteria concerning the fact that the action A_1 is superior to the action A_2.

The first step of our method consists in computing, for each pair of actions (A_1, A_2) and for each criterion j, the concordances $c_j(A_1SA_2)$ and $c_j(A_2\ S\ A_1)$ between the two actions as well as the discordances $d_j(A_1SA_2)$ and $d_j(A_2SA_1)$. The concordance and the discordance are computed from the difference of values of the criterion j for the two actions. We note A^j, the value of the criterion j for the action A. Figure 2 illustrates how to compute the concordance and the discordance values from the difference of values of the actions for the criterion j.

The second step of our method consists in computing, for each pair of actions (A_1, A_2), the *concordance indexes* $C(A_1SA_2)$ and $C(A_2SA_1)$ between the two actions. These indexes represent the mean concordance obtained for the whole criterion set weighted by the criterion weight. It enables to estimate the part of criteria for which an action is at least as good as another one. Let w_j be the weight of criterion j. For two actions A_1 and A_2, the global concordance index is equal to:

$$C(A_1SA_2) = \frac{\sum_{j\in\{criterion\}} w_j \times c_j(A_1SA_2)}{\sum_{j\in\{criterion\}} w_j} \tag{1}$$

The third step consists in computing, for each pair of actions (A_1, A_2), the *credibility indexes* $\rho(A_1\ S\ A_2)$ and $\rho(A_2\ S\ A_1)$ between the two actions. The credibility indexes are computed as follows:

$$\rho(A_1SA_2) = C(A_1SA_2) \times \prod_{j\in\{l\in\{criterion\}/d_l(A_1SA_2)>C(A_1SA_2)\}} \frac{1 - d_j(A_1SA_2)}{1 - c(A_1SA_2)} \tag{2}$$

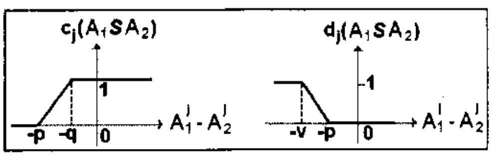

Fig. 1. Computation of the concordance and of the discordance

This index represents the degree with which the action A_1 is at least as good as the action A_2. It corresponds to the *concordance index* weaken by the possible effect of the veto.

The fourth step consists in establishing, according to the values of the *credibility indexes*, the relation between each pair of actions (A_1, A_2). There are four types of possible relations:

- $\rho(A_1SA_2)<\lambda$ and $\rho(A_2SA_1)<\lambda \Rightarrow A_1PA_2$: means that the actions are incomparable.
- $\rho(A_1SA_2)\geq\lambda$ and $\rho(A_2SA_1)<\lambda \Rightarrow A_1PA_2$: means that the action A_1 is better than A_2.
- $\rho(A_1SA_2)<\lambda$ and $\rho(A_2SA_1)\geq\lambda \Rightarrow A_2PA_1$: means that the action A_2 is better than A_1.
- $\rho(A_1SA_2)\geq\lambda$ and $\rho(A_2SA_1)\geq\lambda \Rightarrow A_1IA_2$: means that the two actions are as good.

The last step consists in selecting the best action among the action set. For each action A_1 belonging to the action set *Act*, we compute the preference index $P(A_1)$ that represents the number of times this action was preferred over another action minus the number of times another actions was preferred over this action.

$$P(A_1) = \left|\{A_2 \in Act \,/\, A_1PA_2\}\right| - \left|\{A_2 \in Act \,/\, A_2PA_1\}\right| \quad (3)$$

The action selected is the one that maximises the preference index.

3.4 Learning of the Agent Behaviour

Learning of the action type selection rules. The goal of this step is to learn rules that define, according to the agent state (characterised by a set of criteria), the type of actions that is the most relevant to apply. We remind that we have gathered data concerning the expert behaviour. These data are composed of a set of example (s_{ag}, A_s, a_{exp}) with s_{ag} the agent state, and a_{exp} the action the expert chose to apply when he had to choose between the action set A_s. It is thus easy to build, from these data, a learning set for the action type selection. The attributes are the criteria charactering the agent state. The value is the action type chosen by the expert. Several algorithms can be used to learn rules from such learning set. In this paper, we propose to use the RIPPER algorithm. This one enables to learn relevant rules and has a good generalisation power [5].

Learning from the parameter values of the ELECTRE I method. We propose to formulate the problem of the definition of the best values for the ELECTRE I method parameters as a minimisation problem. We define a global error function that represents the inadequacy between the agent behaviour (and thus the ELECTRE I method parameters values) and the expert behaviour. The goal of this step is to find the parameter values enabling to minimise the global error.

Let P be the current set of utility functions. Let (s_{ag}, A_s, a_{exp}) be an example representing that the expert chose to apply the action a_{exp} when he had to choose between the action set A_s and when the agent state is s_{ag}.

We define the function $error((s_{ag}, A_s, a_{exp})\ P)$ that determines, for an example (A_s, a_{exp}), if the agent behaviour induced by the parameter value P is compatible with the expert behaviour, i.e. if the action applied by the expert behaviour would have been applied taking into account the parameter values. If the agent behaviour is compatible, $error((s_{ag}, A_s, a_{exp}), P) = 0$, otherwise, $error((s_{ag}, A_s, a_{exp}), P) = 1$.

The global error function proposed corresponds to the sum of all errors obtained for each example of the data set *Data*:

$$Error(Data, P) = \sum_{(A_s, a_{\exp}) \in Data} error((s_{ag}, A_s, a_{\exp}), P) \tag{4}$$

The aims of this step is to find the set of parameter values that minimises *Error(Data,P)*. The size of the search space will be most of time too high to carry out a complete search. Thus, it will be necessary to proceed by incomplete search. In this context, we propose to use a metaheuristic to find the best weight assignment. We propose to use genetic algorithms [7] which are particularly effective when the search space is well-structured as it is in our search problem.

4 Case Study

4.1 Case Study Context

The national park of Hwange. In this case study, we study the link between elephant populations and available natural resources. Indeed, the activity of elephants during the day is often related to the utilisation of these resources. Making decision concerning the resources implies to understand the impact of natural resources on the animal populations. In this context we propose to study the activities of elephant populations in the Hwange National Park, located in Zimbabwe. This park presents one of the highest elephant densities of the world [3]. The low availability of water during the dry season has lead to the installation of water pumps. The issue of the park manager is to manage these pumps, in particular to define the flow of the pumps to maintain the elephant demography neither too high nor too low.

Implemented simulation. In this work, we chose to develop our simulation with the GAMA platform [1]. This platform provides a complete modelling and simulation development environment for building spatially explicit multi-agent simulations. The main advantage of this platform is the simplicity to define a model with it. In our simulation, the environment is represented by a grid. We defined six types of cells: void space, grass, bush, forest, natural water tap and pump water tap. We modelled each group of elephant by an agent. An *elephant group* agent represents a group of elephants composed of 20 to 30 elephants. The behaviour of an *elephant group* agent depends on its state and its perceptions. At each step of the simulation, each *elephant group* agent chooses an objective, then a cell to carry out this objective. We defined four objectives for this agent: *Eating*, *Drinking*, *Bathing* and *Sleeping*. In the context of our agent behaviour design approach, the objectives represent the action types. Thus, we have four possible action types. Concerning the target cells, they represent the possible actions: for each step of the simulation, each possible target cell is an action.

4.2 Test Protocol

As a test protocol, we propose to use our approach to learn the behaviour of the *elephant group* agent from a set of examples (the *learning set*) and then to compare

Table 1. *Error rate* obtained on the *learning* and *test* set by the behaviour defined "manually" by the expert and the behaviour defined by our approach

Example set	Behaviour defined "manually" by the expert	Behaviour defined by our approach
Learning set	0.27	0.19
Testing set	0.31	0.25

the learnt behaviour with a behaviour defined by an ecologist expert on another set of examples (the *testing set*). The expert chosen is studying the elephant behavior in the park. He knows the field features and how the elephant move according to attraction points such as water or vegetation. The *learning set* and the *testing set* are composed of 50 examples that represent various situations. For each defined behaviour, we compute on both example sets the *rate of errors*, i.e. the number of actions chosen by the agent (with the considered behaviour) that are different from the ones chosen by the expert.

4.3 Results

As shown on Table 1, the behaviour defined by our approach has obtained better results than the one defined "manually" by the expert. Indeed, the error rates obtained with our approach are lower than the ones obtained with the behaviour defined "manually" by the expert on both example sets. Another advantage of our approach concerns its simplicity to define the agent behaviour. Indeed, with our approach, the expert has just to play the role of an agent in the simulation. In contrary, the "manual" definition of the behaviour requires a long and fastidious tuning process. We can note that even if our approach get good results, these results are not perfect (the error rates are not null). An explanation is the lack of criteria to characterise the agent state and the different actions. Indeed, in our experiment, we only used simple criteria that did not allow to understand some complex decisions made by the expert. In order to learn a more accurate behaviour, additional criteria are needed.

5 Conclusion

In this paper, we presented an approach dedicated to the designing of agent behaviour through the participation of expert playing the role of the agents in the simulation. Our approach is based on the logging of the expert behaviour when this one is confronted to predefined scenarios and on the extraction of knowledge from logs analysis. We proposed to formalise the agent behaviour by using a combination of production rules and of a multi-criteria decision making method. This formalism enables to design complex agent behaviours. We presented a first experiment in the context of ecological simulations that shows promising results for our approach. Indeed, this experiment showed that our approach allow to easily and quickly design a relevant agent behaviour, better than one defined "manually" by an expert. Further experiments need to be carried out in order to study the effectiveness and the