



Using Participatory Paradigm to Learn Human Behaviour

Patrick Taillandier, Thanh-Quang Chu

► To cite this version:

Patrick Taillandier, Thanh-Quang Chu. Using Participatory Paradigm to Learn Human Behaviour. International Conference on Knowledge and Systems Engineering, 2009, Hanoi, Vietnam. pp.55-60. hal-00689151

HAL Id: hal-00689151

<https://hal.science/hal-00689151>

Submitted on 19 Apr 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.

Using Participatory Paradigm to Learn Human Behaviour

Patrick Taillandier, Thanh-Quang Chu
UMI UMMISCO (IRD/IFI)
ngo 42 Ta Quang Buu, Ha Noi, Viet Nam
patrick.taillandier@gmail.com, thanh.quang@gmail.com

Abstract

Since the end of the seventies, the utilisation of multi-agents simulations has spread out. A typical use of these simulations concerns the modelling of human behaviour. In this application case, a key point to ensure the simulation realism is the definition of the agent behaviour. Unfortunately, designing such behaviour is often complex. In order to help the definition of human behaviour, we propose an approach based on the participatory paradigm. In our approach, a human actor directly plays the role of an agent in the simulation. Knowledge about its behaviour is extracted from analysis of the logs. In this context, we propose to formalise the human behaviour by means of utility functions. An experiment, carried out in the domain of rescue simulation, is presented. This first experiment shows promising results for our approach.

1. Introduction

From time immemorial, the study of complex phenomena has been core of numerous researches. Since the end of the seventies, the multi-agent simulation has brought a new way to study it. Multi-agent simulations are based on the agent paradigm. They propose to represent individuals and their interactions in the modelled system. These simulations, which allow to take into account different types of agents as well as an environment explicitly defined, are powerful tools for analysing and understanding the global behaviour of a complex system from the interactions of individual components (the agents). An important utilisation of multi-agent simulations concerns the modelling of human behaviour. Designing such simulations requires to define the behaviour of the agent, i.e. formalise the human behaviour and to express it in a way usable by computers. Unfortunately, this behaviour definition stage, which is a key point of the multi-agent simulation designing, is most of time complex and fastidious.

An approach to face this difficulty is to use role-playing games [2]. Inheriting from this approach, agent-based par-

ticipatory simulations propose to merge multi-agent simulations and role playing games [10]. These simulations propose to design the agents behaviour through the participation of experts. In this paper, we propose an approach dedicated to the learning of human behaviour and based on the participatory paradigm. Indeed, we propose to let a human actor directly plays the role of an agent in the simulation and then to analyse the logs produced in order to extract knowledge about the human actor behaviour.

In Section 2, the general context of our work is introduced. Section 3 is devoted to the presentation of our approach. Section 4 describes an application of our approach in the domain of rescue simulations. We present a first experiment that we carried out as well as its results. Section 5 concludes and presents the perspectives of this work.

2. Context

2.1. Context of the human behaviour learning

2.1.1 Agent action choice as a multi-criteria decision making problem

In this work, we are interested in the designing of the agent behaviour in multi-agent simulations. At each step of the simulation process, the agent has to make a decision concerning its behaviour, i.e. the action he is going to apply. We propose to formulate the problem of agent action choice as a multi-criteria decision making problem: at each step of the simulation process, the agent has to choose an action according to the value of sets of criteria (its perception). In the literature, numerous approaches were proposed to solve this type of multi-criteria decision problems. Among them, several approaches aim at aggregating all criteria in a single criterion (utility function) which is then used to make the decision [12, 13]. Another family of approaches consists in comparing the different possible decisions per pair by the mean of outranking relations [19, 21]. A last family of approaches, which is highly interactive, consists in devising a preliminary solution and in comparing it with other possible solutions to determine the best one [4, 9]. In this work, we

are interested in approaches based on the aggregation of the criteria in a utility function. These approaches have for advantage the clarity of the results obtained [3] that facilitates their validation. At each step of the simulation process, the agent computes the utility of each possible actions thank to utility functions. Then, the agent applies the action that maximises the utility. The next section described the type of utility functions considered in the work.

2.1.2 Description of the utility function set

We propose to formalise the behaviour of the agent by a set of utility functions. Each of these utility functions is used to compute the utility of one type of actions. Indeed, we make the assumption that the behaviour of some complex agents, able to perform several types of actions, can not be expressed by a unique utility function. Typically, a set of criteria can be used to characterise each type of actions. We formulate each utility function as a set of regression rules, in which each regression rule is associated to a weighted linear combination of criteria. The interest of such representation is to allow to define expressive utility function and at the same time to be easily interpretable by domain experts and thus to facilitate the function validation and update. Let S be the whole possible states of the modelled system. Let A_s be the set of possible actions that the agent can apply for a state s belonging to S . We set the constraint that A_s is finite. However, the total number of possible actions for all states ($\bigcup_{s \in S} A_s$) can be infinite. Let C be the set of criteria charactering a type of actions. We note w_i the weight associated to the criteria i and $Val_i(a)$, the value of the criteria i for the action a belonging to A_s . The criteria are defined such as:

$$\forall s \in S, \forall a \in A_s, \forall i \in C, \\ VAL_MIN \leq val_i(a) \leq VAL_MAX$$

with VAL_MIN and VAL_MAX real.

Each regression rule has the following format:

$$\text{if condition then} \\ utility(a) = \sum_{i \in C} \frac{1}{w_i} \times \sum_{i \in C} w_i \times Val_i(a)$$

2.1.3 Problem of the utility function set learning

The agent behaviour learning problem can be modelled as a utility function set definition problem in which the goal is to define the most pertinent regression rules (their condition and their associated criterion weights). Two questions arise from this learning problem: which data used to learn the utility function set and how learning utility functions from data. An approach to answer the first question consists in

using the participatory paradigm in order to directly capture the behaviour of human actors. Concerning the second question, the domain of machine learning provides many techniques that can be used to learn a general model (such as our utility function set) from particular examples. In the next section, we describe different works that are directly linked to these two questions.

2.2. Related works

The utilisation of the participatory paradigm to learn human behaviour is a recent research topic. However, some works have to be noted. Thus, reference [15] proposes to learn a user model by analysing the user activity logs. Specific domain knowledge is used to generate explanation for the user behaviour. Reference [20] proposes an approach in which human actors play their own role in an agent-mediated simulation and interact with artificial agents that question their behaviours. The objective is to use these interactions to stimulate the human actor reactions and thus to reveal hidden knowledge. At last, reference [5] proposes an approach aiming at learning the expert behaviour by means of interactions between the expert and the system. Thus, the expert observes the behaviour of an agent and has the possibility to correct it if this one is not pertinent. The agent takes into account this intervention to refine its behaviour. Reference [5] proposes an implementation of this approach in the context of rescue simulations. In this paper, we propose an approach that takes place in the continuity of the latter. Actually, we propose an approach based as well on the learning of the agent behaviour through the participation of a human actor. However, we do not seek to learn the agent behaviour by means of the human actor intervention but let him directly plays the role of an agent in the simulation. Thus, we do not have to refine step by step the agent behaviour but to directly learn it by analysing the logs. Another difference concerns the formalisation used to express the agent behaviour: as in [5], we propose to express the agent behaviour by means of utility functions. Yet, in [5], the utility function consists in a weighted linear combination of criteria. In this paper, we propose to extend the utility function definition domain by defining the utility function as a set of regression rules, in which each rule is associated to a different weighted linear combination of criteria and by giving the possibility to define one utility function per type of actions (cf. Section 2.1.2).

3. Proposed Approach

3.1. General approach

As stated in Section 2.2, we seek to learn human behaviour through the direct participation of a human actor in the simulation.

Our approach is composed of two stages: the first one consists in producing data concerning the behaviour of the human actor, the second one in analysing these data in order to learn the human behaviour, i.e. the utility function set (cf. Section 2.1.3). In the following sections, we described each of these steps.

3.2. Production of data concerning the human behaviour

This stage consists in producing data representing the human actor behaviour. In order to produce these data, we propose to use the participatory paradigm: a human actor directly interacts in the simulation by playing the role of an agent. At each step of the simulation, the action he chooses is logged. The data describing his behaviour will thus be composed of a set of examples represented by couples A_s , a_{HA} with A_s representing the set of possible actions to apply in the state s of the system, and a_{HA} , the action chosen by the human actor. Each action of A_s is characterised by the values of the set of criteria corresponding to the type of actions concerned. A key point of this stage is the choice of the system states that will be used to log the human actor behaviour.

In order to face this system states choice problem, we define the notion of scenario: a scenario is a sequence of steps of the simulation. Each step represents a state of the modelled system. Thus, choosing system states for the logging stage means choosing scenarios. Learning an agent behaviour really in adequacy with the human actor behaviour requires to use pertinent scenarios. In particular, these scenarios have to propose a wide panel of situations, and among them, complex conflicting situations in which the decision making requires to establish priorities between different criteria.

3.3. Learning of the utility function set

The second stage of our approach consists in analysing the data produced during the last stage in order to learn the agent behaviour, i.e. the utility function set. The goal is to build utility function set that allow to choose, for all examples, the same action as the one chosen by the human actor. We remind that at each step, the utility of each action is computed thanks to the utility function set and that the agent chooses to apply the action that maximised its utility (cf. Section 2.1.1). Building the utility functions consists, for each utility function, in defining a set of regression rules (condition of the rule + weight values) from the data produced, i.e. the human actor behaviour. In order to carry out this building, we propose to use an approach based on the search of the best weights and on the partitioning of the criteria sets (which correspond to the addition of new regression rules) as presented in Figure 1.

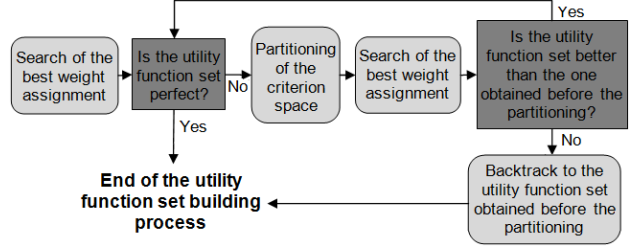


Figure 1. Partitioning method

At the initial stage, all utility functions are composed of only one regression rule, such as the criterion space is composed of only one partition. At the first step, the system searches a weight assignment that maximises the adequacy between the utility functions and the human actor behaviour. If this weight assignment is in total adequacy with the human actor behaviour, the process ends; the utility functions are composed of only one regression rule. Otherwise, new regression rules are introduced: for each utility function arising problems, the system computes partitions of the criteria set in order to detect the parts of the criteria set that are not compatible with the others. Then, a new weight assignment is searched again for the whole regression rules and for all utility functions, by considering all partitions built at the same time. If the weight assignment obtained after the partitioning allows to get a better result than the previous one, it is kept. Otherwise, the system backtracks to the previous utility functions and ends the utility function set building process. This partitioning procedure is recursively repeated until the learnt utility functions allow to obtain a behaviour in total adequacy with the human actor behaviour or until there is no more improvement of the utility functions.

3.3.1 Search of the best weight assignment

We propose to formulate the problem of the best weight assignment as a minimisation problem. We define a global error function that represents the inadequacy between the utility function set (and thus the weight assignment) and the human actor behaviour. The goal of the best weights assignment search is to find the weights that allow to minimise the global error function.

Let F be the current set of utility functions. Let $f_{ut}(a, F)$ be the function that computes the utility of an action a according to the utility function set F . We make the assumption that the action a is automatically evaluated by the utility function of F that corresponds to the type of action of which a belongs.

Let A_s , a_{HA} be an example representing that the human actor chose to apply the action a_{HA} when he had to choose between the action set A_s . We define the function $error(A_s, a_{HA}, F)$ that return the error value for an example A_s , a_{HA} and an utility function set F . This

function is defined as:

$$error(\{A_s, a_{HA}\}, F) = \min \left(1, \frac{\max_{a \in A_s} (f_u(a, F)) - f_u(a_{HA}, F)}{VAL_MAX - VAL_MIN} + val_{er} \right)$$

In this function, we integrated a parameter val_{er} that represents the minimum importance of an error whatever the values of the utility for the two actions are. The higher the value of this parameter, the more important it will be to minimise the number of incompatible examples. This function has for value a real ranged between 0 (utility function set in total adequacy with the human actor behaviour for this example) and 1 (utility function set in total inadequacy with the human actor behaviour for this example).

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

$$Error(F, Data) = \sum_{A_s, a_{HA} \in Data} error(A_s, a_{HA}, F)$$

The aims of the weight assignment step is to find a weight assignment that minimises $Error(F, Data)$. 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. In the literature, numerous metaheuristic were proposed [14, 7]. In this paper, we propose to use genetic algorithms [11] with are particularly effective when the search space is well-structured as it is in our search problem.

3.3.2 Partitioning of the measure space

For some agent behaviour definition problems, it will not be possible to find a weight assignment compatible with all examples of the data set. Thus, we propose to partition the criteria set space and to define for each partition a regression rule with its own weight assignment.

We propose to base our partitioning method on the utilisation of supervised learning techniques. The goal is to search the parts of the criterion spaces that have a different behaviour in terms of utility functions. Thus, we search to detect, for each utility function, the parts of the criterion space which contain action linked to an incompatible example.

For each utility function, we built a learning set composed of couples action, conclusion. The actions are described by the values of the criteria set. The conclusion could be either "compatible" if the example which contains the action is compatible with the utility functions or "incompatible" if it is not. Then, a supervised learning algorithm is used to partition the criterion space linked to the considered utility function. We remind that we proposed to express the partition in the form of rules. Thus, it is necessary to use a supervised learning algorithm that allows to build a

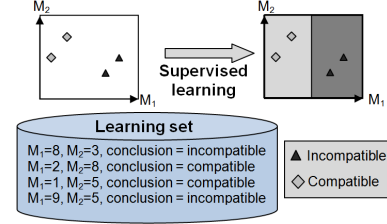


Figure 2. Partitioning method

predictive model expressed by rules. Different algorithms could be used for this partitioning problem such as RIDOR [8] or C4.5 [18]. In this paper, we propose to use the well-established and efficient RIPPER algorithm [6]. Figure 2 presents an example of partitioning for a criterion space composed of two criteria (M_1 and M_2).

Once the partitioning is carried out, the user need definition module performs a new search of the best weight assignment. All partitions of all utility functions are considered at the same time for this search. If the weight assignment found is better (in terms of minimisation of the global error value) than the assignment obtained before the partitioning, the new utility function is kept. Otherwise, the module keeps the previously obtained utility function set and the utility function set building process ends.

4. Case study

4.1. General Context

4.1.1 Rescue simulation

We propose to apply our approach in the domain of rescue simulations and more particularly in the context of emergency responses (rescue management).

The problem of emergency responses to disasters is a very serious and complex social issue. It involves a large number of heterogeneous actors that have to work together in a hostile environment. In the recent years, many research works proposed to study this problem through agent-based simulations [16, 17]. Indeed, agent-based simulations are powerful tools to analyse large-scale urban disasters and the emergency responses resulting from it. They can take into account a large amount of information and manage heterogeneous agents. However, in order to implement pertinent rescue simulations, the definition of credible agents is particularly important. In this context, approaches such as ours are very interesting.

4.1.2 Implemented rescue simulation

We build a rescue simulation on the GAMA simulation platform [1] This platform, which was developed by the MSI team, aims at providing a complete modelling and simulation development environment for building spatially explicit multi-agent simulations.

4.2. Case study context

4.2.1 Ambulance agent

As mentioned in the last section, agent-based emergency response simulation often implies heterogeneous agents. As a case-study, we propose to learn the behaviour of ambulances. Ambulances have for assignment to rescue the injured, i.e. to provide assistance to injured victims while minimising the number of deaths. In the simulation context, we assume that the role of ambulances is take injured victims to hospitals. Each ambulance has a maximal of transported victims. An ambulance cannot take more victims at the same time than its capacity. In the same way, each hospital has a maximal capacity of injured victims of which it can take care.

Ambulances have two types of actions:

- Displacement toward a victim: the ambulance can move in direction of each localised victim. We assume that the ambulance know the shortest path to reach them.
- Displacement toward a hospital: the ambulance can move in direction of each hospital.

When an ambulance reaches a victim, the ambulance can load the victim. When an ambulance reaches a hospital, the victims transported by the ambulance are unloaded and the hospital takes care of the victims.

We define two sets of criteria linked to each type of actions.

- Criteria used for both action types:
 - Distance between the ambulance and the victim/hospital.
 - Number of close victims.
 - Maximum of the victim injury seriousness among the close victims.
 - Maximum of the victim injury seriousness among the victims loaded in the ambulance.
 - Number of victims that can be still loaded in the ambulance.
- Criterion specific to action type "Displacement toward a victim":
 - Seriousness of the victim injury.
- Criterion specific to action type "Displacement toward a hospital":
 - Number of victims of which the hospital can take care.

As a first experiment of our approach, we propose to use a simple rescue context where only one ambulance is as stake in the simulation. This experimental context, very basic, allows to give a first evaluation of our approach. Carrying out more complex experiments is one of our perspectives (cf. Section 5). We used, as GIS data for the experiment, the Ba-Dinh district of Hanoi.

4.2.2 Defined scenarios

In this paper, we propose three scenarios based on conflicting situations in which the ambulance has to make complex decision.

- Importance of Injury level: in this scenario, the ambulance has the choice to rescue in priority several victims with light or medium injury that are located in the same area or to rescue in priority one victim with serious or very serious injury located in another area.
- Ambulance capacity problem: in this scenario, two groups of victims are located in the two different areas. One of this group is composed of a number of victims lower than the ambulance capacity while the other groups of victims is composed of a number of victims higher than the ambulance capacity.
- Hospital capacity problem: in this scenario, a group of victims is defined such as their number is lower than the ambulance capacity. The hospital that is the nearest to this group of victims is almost full, i.e. it cannot take care of all victims of the group. Another hospital, farer and in opposite direction from the first hospital, has the capacity to take care of all victims of the group.

4.2.3 Test protocol

As a test protocol, we propose to use our approach to learn an utility function set (*Learnt Fct*) for the ambulance agents and then to compare it to one defined by an human actor (*Human Fct*). The goal is to determine if our approach can allow to define a better utility function set than the one directly defined by the human actor. Concerning the learning stage, we used the three scenarios defined in the last section. Thus, for this stage, the human actor played three times each of these scenarios with different parameters (learning data). For the test part, we let the human actor play the role of the ambulance during three sessions based on scenarios built randomly, in which, victims, with a random level of injury, appear randomly on the map (test data). In each session, 10 victims appear in the simulation. We used these random scenarios to evaluate the adequacy between the actions chosen by the human actor and the ones chosen by the utility function set.

Table 1. Global error for the learnt utility functions

Data set	Error(F, Data)	
	Human Fct	Learnt Fct
Learning_data	0.12	0.04
Test_data	0.15	0.09

4.3. Results

Results presented Table 1 show that our approach allowed to learn a good utility function set. Indeed, the learnt utility function set obtained significantly better results on both learning data and test data than the one defined by the human actor.

However, the results obtained are not perfect. Actually, the global error rate is not nil. An explanation is the lack of criteria to characterise the system state. Indeed, Human beings often use complex spatial information to make decision. In our experiment, we only used simple spatial criteria that did not allow to understand some complex decisions made by the human actor. In order to learn a more accurate behaviour, additional criteria are needed.

5. Conclusion

In this paper, we presented an approach dedicated to the learning of human behaviour through the participation of human actors playing their own role in the simulation. Our approach is based on the logging of the human actor behaviour when this one is confronted to predefined scenarios and on the learning, by logs analysis, of utility functions representing the human actor behaviour. We presented a first experiment in the context of rescue simulations that shows promising result for our approach.

In this first experiment, we concentrated our attention on a single agent without considering its interaction with others agents. Yet, these interactions play a key role in the human behaviour. Thus, a perspective of this work is to test our approach in the context where several agents interact with each others. In this context, the work of [10] proposing to distributed the participatory simulation in order to let several human actors play in the simulation at the same time, could be particularly useful.

Concerning the exploration part as well as the partitioning part, we tested one search algorithm and one supervised learning algorithm. An interesting study could be to test others algorithms and to compare the results with the ones obtained.

A last perspective is to pass from a utility functions acquisition problem to utility functions revision problem. Indeed, it could be interesting to take into account initial utility functions and to refine them rather than learning new ones from scratch.

References

- [1] E. Amouroux, T.-C. Chu, B. A., and D. A. Gama: an environment for implementing and running spatially explicit multi-agent simulations. In *PRIMA*, 2007.
- [2] O. Barreteau, M. Antona, P. D'Aquino, S. Aubert, S. Boissau, F. Bousquet, W. Dar, M. Etienne, C. Le Page, R. Mathévet, T. G., and W. J. Our companion modelling approach. *JASSS*, 6(1), 2003.
- [3] S. Ben Mena. Introduction aux méthodes multicritères d'aide la décision. *Bio. Agro. Soc. Environ.*, 4:83–93, 2000.
- [4] R. Benayoun, O. Laritchev, J. de Mongolfier, and J. Tegny. Linear programming with multiple objective functions: Step method (stem). *Math. Program.*, 1(3):366375, 1971.
- [5] T. Chu, A. Boucher, A. Drogoul, D. A. Vo, H. Nguyen, and J. Zucker. Interactive learning of expert criteria for rescue simulations. In *PRIMA*, 2008.
- [6] W. Cohen. Fast effective rule induction. In *ICML*, 1995.
- [7] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, 2004.
- [8] B. Gaines and P. Compton. Induction of ripple-down rules applied to modeling large databases. *JGIS*, 5:211–228, 1995.
- [9] A. Geoffrion, J. Dyer, and A. Feinberg. An interactive approach for multicriterion optimisation with an application to the operation of an academic department. *Manage. Sci.*, 19:357368, 1972.
- [10] P. Guyot, A. Drogoul, and S. Honiden. Power and negotiation: lessons from agent-based participatory simulations. In *AAMAS*, 2006.
- [11] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [12] J. Ignizio. A review of goal programming: a tool for multi objective analysis. *J. Oper. Res. Soc.*, 29(11):–, 1978.
- [13] E. Jacquet-Lagrange and J. Siskos. Assessing a set of additive utility functions for multicriteria decision making, the uta method. *Eur. J. Oper. Res.*, 10(2):151164, 1982.
- [14] S. Kirkpatrick, C. Gelatt, and V. M.P. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [15] Y. Murakami, S. Y., and T. Ishida. Modeling human behavior for virtual training systems. In *AAAI*, 2005.
- [16] G. Narzisi, V. Mysore, and B. Mishra. Multi-objective evolutionary optimization of agent-based models: An application to emergency response planning. In *IASTED*, 2006.
- [17] S. Paquet, B. Chaib-draa, and S. Ross. Hybrid pomdp algorithms. In *Workshop MSDM*, 2006.
- [18] J. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.
- [19] B. Roy. The outranking approach and the foundations of elective methods. *Theory and Decision*, 31:141–160, 1991.
- [20] F. Semp, M. Nguyen-Duc, A. Boucher, and A. Drogoul. An artificial maieutic approach for eliciting experts' knowledge in multi-agent simulations. In *AAMAS*, 2005.
- [21] G. Siskos, J. Wscher and H. Winkels. A bibliography of outranking approaches. Technical report, Cahiers du Lamsade, Paris Dauphine, 1983.