Engineering Applications of Artificial Intelligence 26 (2013) 211–226

Contents lists available at SciVerse ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Formalizing the construction of populations in multi-agent simulations

Benoit Lacroix [a,b,*], Philippe Mathieu [b], Andras Kemeny [a]

[a] Renault, Technical Center for Simulation, 1 avenue du Golf, 78288 Guyancourt Cedex, France
[b] University of Lille, Computer Science Department, LIFL, 59655 Villeneuve d'Ascq Cedex, France

ARTICLE INFO

Article history:
Received 12 September 2011
Received in revised form
17 February 2012
Accepted 15 May 2012
Available online 15 June 2012

Keywords:
Agent
Agent-based simulation
Behavior
Variety
Consistency
Road traffic

ABSTRACT

In individual-centered simulations, the variety and consistency of agents' behaviors reinforce the realism and validity of the simulation. Variety increases the diversity of behaviors that users meet during the simulation. Consistency ensures that these behaviors improve the users' feeling of immersion. In this work, we address the issue of the simultaneous influence of these two elements. We propose a formalization of the construction of populations for agent-based simulations, which provides the basis for a generic and non-intrusive tool allowing an out-of-the-agent design. First, the model uses behavioral patterns to describe standards of behaviors for the agents. They provide a behavioral archetype during agents' creation, and are also a compliance reference, that allows to detect deviant behaviors and address them. Then, a specific process instantiates the agents by using the specification provided by the patterns. Finally, inference enables to automate behavioral patterns configuration from real or simulated data. This formalization allows for the easy introduction of variety in agents' behaviors, while controlling the conformity to specifications. We applied the model to traffic simulation, in order to introduce driving styles specified using behavioral patterns (e.g. cautious or aggressive drivers). The behavioral realism of the traffic was therefore improved, and the experimentations we conducted show how the model contributes to increase the variety and the representativeness of the behaviors.

1. Introduction

Variety and consistency are two keys that foster the realism in individual-centered simulations (Maim et al., 2009; Wright et al., 2002). Indeed, repetitive or inconsistent behaviors highly disturb users' immersion. In this work, we propose a solution that enables to improve the agents' behavioral realism during multi-agent simulations. For this purpose, we adopt an approach that explicitly takes into account the variety and consistency of the behaviors.

In this paper, we propose a behavioral differentiation model, which provides the basis for a generic and non-intrusive tool, allowing an out-of-the-agent design. It describes the agents' behavior using behavioral patterns, which describe standards of behaviors, external to the simulation model, that are used to create agents of specific categories. At runtime, the behaviors conformity is checked using the specifications provided by the patterns. The agents are created using a specific process that allows to introduce behavioral irregularities. This process is inspired by an approach proposed by Reynolds (1999) for fuzzy path following. We have generalized and extended it to generate values in a constrained space, and to control the randomness of the process. Finally, the behavioral patterns can be inferred from simulations records or real-world situations. Based on an unsupervised learning technique, the self-organizing maps proposed by Kohonen (1995), inference enables to automatize the model configuration.

The main contribution of this work is a formalization of the construction of populations for agent-based simulations, both ab initio and as generalization from sample data. Moreover, it describes a set of tools that not only enhances the immersion of the final user during the simulation – thanks to the improvement of the realism – but also assists the simulation designer in building scenarios in a more automated way.

We applied this model to traffic simulation in SCANeR^{TM}, the application developed and used for driving simulation by the French car manufacturer Renault (Reymond and Kemeny, 2000; Oktal, 2012). The software modules we developed introduce various driving styles in the traffic (e.g. aggressive, cautious drivers). These driving styles are specified using behavioral patterns. They allow for the easy population of the database in an automated way, and the update of pre-existing scenarios. These developments have already been included in the commercial version of the software. Beyond the subjective improvement in the

* Corresponding author at: University of Lille, Computer Science Department, LIFL, 59655 Villeneuve d'Ascq Cedex, France.
E-mail addresses: lacroix.benoit@gmail.com (B. Lacroix), philippe.mathieu@univ-lille1.fr (P. Mathieu), andras.kemeny@renault.com (A. Kemeny).

0952-1976/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2012.05.011

traffic realism, our experimentations show how the proposed tool increases the variety and representativeness of the behaviors.

This paper is organized as follows. In Section 2, we present related works. Sections 3 and 4 present the grounds for this work, and an overview of the proposed approach. We then describe our model: the data model (Section 5), the generation method and algorithms (Section 7), and the inference techniques used to automate the simulation configuration (Section 8). Section 9 shows how the model addresses our issue: increasing variety in simulation, while insuring the behaviors consistency. Section 10 presents the application of the model to traffic simulation in driving simulators, and how it improves the realism in an industrial application. Section 11 introduces another example, to illustrate the genericity of the approach. Finally, the model is discussed in Section 12, and future works are presented in Section 13.

2. Related works

Our work focuses on how we can articulate both behavioral variety and consistency in simulations. These two notions are considered as central issues in various works.

For instance, in crowd simulations, the realism is crucial to improve the users' immersion during the simulation, and to improve the results validity. The realism is directly related to the variety of the agents' looks. Techniques increasing this diversity have thus been developed: for instance, Maim et al. (2009) automatically change virtual humans' appearance by using diverse colors, accessories and shape parameters. Variety in the simulation is increased, which improves users' immersion. However, this approach is based on hard coded parameters in the graphics models, which limits its genericity and flexibility.

In videogames, more and more attention is focused on the improvement of the gameplay. Moreover, in that domain, the non-player agents' behaviors are often implemented as scripts, defined by long and complex list of rules (Tozour, 2002). However, such scripts can contain weaknesses that decrease the player's immersion in the virtual environment. Focusing in particular on the variety and consistency of the behaviors, Spronk et al. (2006) have proposed to dynamically improve agents' behaviors using online learning. The agents' behaviors being defined by sets of rules associated to activation weights, those weights are then dynamically adapted during the game to improve the behaviors. However, during the agent's creation, these weights are randomly selected: the approach does not specifically address this initial instantiation. It could therefore benefit from an automated mechanism introducing behavioral profiles at this level.

In the computer graphics and virtual reality field, various approaches focus on easing the configuration of the virtual environments. For instance, Ulicny et al. (2004) have proposed dedicated tools for the designers, based on a *painting* metaphor. Using a *brush*, the designer can *paint* new pedestrians in a simulation, or visual and behavioral characteristics on existing ones. This approach enables to easily create agents and to increase the behavioral variety, but remains focused on the graphical part of the simulation. In other works, predefined behavioral patterns are proposed to the user (Pellens et al., 2009). After having selected a pattern, the user has to adapt the pattern parameters to the values he desires. However, this approach does not take into account the possibility to automate the construction of whole populations.

Finally, in driving simulation, Wright et al. (2002) created virtual drivers' characters to reinforce the users' immersion during the simulation. The model combines the drivers' speed choices with behavioral parameters (sex, age, aggressiveness, alcoholic intoxication and fatigue). When implemented in a driving simulator, the model improves the realism as perceived by the users. However, this approach lacks flexibility: the modifications have to be integrated within the core of the traffic model. Such changes are not easy to introduce in commercial softwares, due to back-compatibility issues or even because no modification is allowed on these core components.

3. Motivation of the work

Our initial objective was to enhance the simulation realism. In this section, we discuss how this objective led us to a second one, which is to help the designer elaborate simulation scenarios. Then, we describe the constraints introduced by the requirement to integrate the proposed approach in a commercial software, and the design choices needed to meet these requirements.

3.1. Objectives

The main objective of this work was to improve the realism of the agents' behavior in multi-agent simulations. To do so, we focused on the variety and consistency of the behaviors. Indeed, simultaneously taking both these dimensions into account is important for the users' immersion into a simulation. In traffic simulation, the observation of different kind of behaviors (e.g. aggressive or cautious drivers) contributes to the realism feeling (Wright et al., 2002). However, an aggressive driver is characterized not only by a high speed, but also by short security distances and a tendency to disregard speed limits. Consistency among all these properties – and not only a subset of them – characterizes the "aggressive driver" category. To handle these dependencies, we, therefore, proposed to explicitly take both variety and consistency into account.

Furthermore, in most commercial simulation tools, a scenario is designed specifically for one experiment. A designer is in charge of creating and implementing these scenarios. Since the design of the scenario highly influences the simulation realism, he plays a crucial role in the simulation outcomes. For instance, in SCANeR^{TM}, the simulation tool used at Renault for driving simulation, each vehicle had to be manually added in the scenario, and each vehicle parameter had to be manually modified: reproducing the variety usually met in a real traffic was thus a long and repetitive task. Moreover, if the parameter values were not carefully chosen, unrealistic behaviors could appear during the simulation. The consequences were that scenarios were usually created with a small number of vehicles, and that these vehicles kept their (similar) default parameters.

To successfully increase the realism during a simulation, we had thus not only to improve the agents' behaviors themselves, but also to help the scenario designer to introduce these realistic behaviors into the simulation.

3.2. Constraints

Another dimension we had to take into account when designing the model is that it had to be integrated into a commercial software. This introduced two specific constraints: first, the proposed model had to provide a running mode where non-regression with the previous version would be insured; second, the proposed model had to be totally non-intrusive, as no modification of the original source code of the simulation software was allowed by the editor.

To guaranty non-regression, we proposed to use a two-level control mechanism in the agents' creation process. The first level provides an easy way to switch off the improvements introduced by the model, and to return to the previous simulation running