Lectures on Modelling and Simulation; A selection from AMSE # 2016-N°2; pp 66-76
Selection at the AMSE Conference Teruel/Spain, July 4-5, 2016

Towards Agent-Based Simulation Support for Training Teachers

*Iván García-Magariño, **Inmaculada Plaza, ***Raúl Igual, ****Andrés S. Lombas

*Department of Computer Science and Engineering of Systems, University of Zaragoza
Spain, E. U. Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, (ivangmg@unizar.es)
** Department of Electronics Engineering and Communications, University of Zaragoza
Spain, E. U. Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, (inmap@unizar)
*** Department of Electrical Engineering, University of Zaragoza
Spain, E. U. Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, (rigual@unizar.es)
**** Department of Psychology and Sociology, University of Zaragoza
Spain, Fac. Ciencias Sociales y Humanas, c/ Atarazana 2, 44003 Teruel, (slombas@unizar.es)

Abstract

The quality of teacher training is crucial as it influences the academic performance of students. The last decades have witnessed a steep increase in the use of information applications for improving the training of teachers. In this line of work, the current approach proposes to use agent-based simulation for supporting the training of teachers. The current work focuses on the capability of teachers in designing appropriate teaching strategies with the corresponding schedules of learning activities. In particular, the current approach uses FTS-SOCI (an agent-based Framework for simulating Teaching Strategies with evolutions of Sociograms) for simulating the social repercussions of certain teaching strategies. Teachers can pursue obtaining groups with high cohesion, which is normally related with a high academic performance. This article illustrates the current approach with the training of a computer science teacher for the programming subject.

Key words

Agent-based simulation, agent-oriented software engineering, multi-agent system, social simulation, teacher training

1. Introduction

Several studies show that the quality of teachers' training influences their productivity in promoting student achievement [1, 2]. The last decades have witnessed an increasing demand of reducing the costs of training teachers through information and communication technologies (ICT) [3]. Most of the current works use general-purpose tools in the training of teachers. Examples of these tools vary from the ones for sharing documents [4] to social media applications [5].

One of the keys for achieving a proper training of teachers is to acquire practice with real students or at least with simulated situations that are similar to the real ones. Before becoming autonomous teachers, candidates can practice by participating in learning activities of some ongoing courses. However, it is less common that teachers' candidates can design a teaching strategy with a schedule of learning activities and put it into real practice. In this particular task of designing teaching strategies, we propose agent-based simulation as a possible tool support for letting teacher candidates get this practice.

Teaching strategies can include both individual and collaborative learning activities. Among others, the latter ones can be discussions in class, teamwork in pairs or larger groups, or role-play activities. These collaborative activities can help students in knowing each other and establishing new social relations. It is well known that the cohesion of a group in terms of social relations can positively influence in the performance of their teamwork [6]. Thus, teachers could promote social relations to have group of students with a high cohesion, which will probably imply a high academic performance.

In this context, the current approach proposes that teachers are trained by designing different teaching strategies and simulating their repercussion in the social relations of the group. For this purpose, Agent-Based Simulators (ABSs) have shown to be especially useful for simulating social repercussions [7]. In particular, the current work proposes to use FTS-SOCI (an agent-based Framework for simulating Teaching Strategies with evolutions of Sociograms) [8] and the associated technique for designing teaching strategies [9]. In this way, teachers can learn to design teaching strategies that will probably promote the class cohesion obtaining a high academic performance in average.

2. Background

2.1 Training teachers with new technologies

The teaching training has been supported with different technologies. Some technologies have been specifically designed for training teachers, while other studies just use general-purpose tools for it.

In the group of works about the former kind of technologies, Ke and Xu [10] present a mixed-reality integrated learning environment where university teaching assistants can practice. Their system uses a Kinect interface, where teachers and students were represented with virtual avatars. They concluded that their system improved the sense of presence and supported the performance of a wide range of virtual teaching tasks and actions. Moreover, Jiménez and O'Shanahan [11] use a web-based tutorial program for reading instruction, called ``Letra'', for training teachers. In the web-based tutorial, teachers can learn how children can reach a good level of phonological awareness. Teachers can access to the digital tutorials from a row of books in a virtual library environment. Brubacher et al. [12] provided online training for teachers in which they learned how to effectively question children in suspicious circumstances to detect child abuse. They practiced simulated interviews with a virtual avatar.

In the case of applying general-purpose tools for teachers' training, Hassan et al. [4] present a training program of teachers that follows a specific way of using certain open source technologies. Among other tools, they used the Google Form, Google Calendar and Google Drive. They show the benefits of using these tools in the training of teacher, who then were able to use these tools with their students. In this line of research, Igual et al. [13] discuss some free software simulation tools that teachers use for preparing the learning activities in engineering higher education. Furthermore, Salminen et al. [5] described the training of nurse teacher candidates with social media applications.

In this context, the current approach belongs to the group of works in which an application is specifically designed for training teachers. More concretely, the current approach is focused on providing a simulation environment in which the teachers can practice different schedules of learning activities. The used application is an ABS, and the bases of this kind of systems are presented in the next section

2.2 Agent-based simulators: foundations and practical applications

A proper introduction of ABSs needs a brief discussion of their precursors, the multi-agent systems (MASs). The paradigm of MASs probably emerged from the need of distributing the techniques of artificial intelligence [14]. These systems were composed of autonomous entities, which were called agents. From the beginning of this paradigm, the definitions of agents were extensively discussed with different definitions. Franklin and Graesser [15] collected the most relevant definitions. For instance, the MuBot definition focused on the autonomy and the ability of reasoning. By contrast, the AIMA definition indicated the agent was an entity that was able to perceive the environment and to act on it. Other definitions, such as the KidSim definition introduced the concept of goals of agents (also referred as agendas). This last kind of agents was aligned with the well known Belief-Desire-Intention (BDI) architectures [16].

The most relevant features of agents were not only their individual features, but also their social activities when they interact among each other [17]. For instance, Fitoussi and Moshe [18] discussed the way of choosing social laws for reaching coordination.

In this context, from a software engineering point of view, the last decades have witnessed the proposal of a large variety of agent-oriented software engineering processes and methodologies. In particular, Cernuzzi et al. [19] discuss different process models that can be followed for developing MASs. Due to the variety of agent-development processes, a tool was developed for specifically defining agent-oriented development processes using a subset of the Software & Systems Process Engineering Metamodel (SPEM) [20]. Regarding the agent-oriented methodologies, the Gaia methodology [21] introduced a well-sound theoretical basis that many of the later works about MASs considered. From a practical point of view, the Ingenias and Prometheus methodologies provided alternative ways of developing agents following a model-driven approach, with respectively the Ingenias Development Kit [22] and the Prometheus Design Tool [23]. Practitioners can define a MAS model with some graphical interface, and a considerable part of the programming code is automatically generated from it, by means of templates and/or model transformations [24]. In the former methodology, the Ingenias Agent Framework was extended to support simulations [25], so the Ingenias language included many aspects that were necessary for fully defining ABSs.

In this context, some MASs were developed for simulating groups of entities with social interactions, and the number of this kind of systems increased until a subfield of MASs appeared with the name of agent-based social simulation [26]. In this subfield, ABSs started to have some specific needs. For instance, they needed to run large number of agents or analyze large amounts of data, like for example when simulating the political elections with the citizens of a city [27], the discussions of experts with the knowledge based on large sets of Wikipedia documents [28], or all the tourists that come to a city in a year [29]. However, these systems do not normally need some features of some non-simulation MASs such as the possibility of physical distribution of agents (i.e. each agent in a different device or server). In this context, we proposed a Process for developing Efficient Agent-Based Simulators (PEABS) [30]. This process is based on the common needs of ABSs. It uses the expressivity of the Ingenias language with its modeling tool, and at the same time it provides a framework for developing simulations with large amounts of agents in a reasonable response time. There are other processes for developing efficient MASs like the one of Di Stefano and Santoro [31], but this process does not use a graphical modeling language specific for ABSs, which is useful in the early design of these systems. Therefore, FTS-SOCI was developed with PEABS, and the next section presents its practical application.

3. Training teachers with FTS-SOCI

In the current approach for training teachers, firstly they must be aware that the cohesion of a group is normally related with its performance. Thus, teachers should know that if most students get on well with each other, they will probably be able to learn and work with enthusiasm and without worries about negative social relations.

Secondly, teachers need to get used to the user interface of FTS-SOCI [8]. In the main interface, teachers or candidate teachers select the number of students of each type. The available types of students are quiet students (also known as passive), participant students (also known as active), tangent students (i.e. they derive discussions to tangent topics that are irrelevant for the subject), joker students (they make jokes cheering the class) and occasional participant students (with an intermediate activity level). In the training phase, normally teachers do not have a real group of students yet, so they should simulate common distribution of student types. For example, in a group of 20 students, the current approach suggests a common distribution of five quiet students, four participant students, one tangent student, one joker student, two obstructive students and seven occasional participants. However, there can be other common distributions regarding their institutions, the subjects, the academic level, and so on. Thus, teachers can use other distributions if for example they know the common proportions of students in their universities and subjects.

In addition, the number of iterations must be set to the number of class hours. If they are training for a specific subject, they should enter the number of class hours of the subject. Otherwise, teachers are recommended to select 60, which is a common number of hours for a subject.

Regarding the simulation speed, in a first phase of training they are recommended to select between 0.5 to 2 iterations per second. In this manner, teachers can slowly observe the evolution of the sociometric status of the class represented with a sociogram. They can understand the particular influences of certain learning activities in different parts of the subject schedule. In the initial phase of training, teachers are recommended to run only one or very few simulations at each execution.

In the selection of the teaching strategy in the interface, teachers should first try to simulate one of the default teaching strategies. Then, teachers are recommended to design their own teaching strategies. They should choose the number of learning activities and their types (i.e. teamwork in pairs or larger groups, discussions and role play activities). In the case of teamwork, they should also select the number of members for each group. They need to think about the content of these activities so that the teaching strategy is feasible and aligned with the goals of the particular subject. After designing a teaching strategy, the teacher defines this strategy by means of the framework of FTS-SOCI by simply extending a Java object-oriented class and implementing one method where they can call the inherited methods in the appropriate iterations. The interested readers can read further about the definition of strategies with FTS-SOCI in our previous work [8].

When teachers simulate different possible teaching strategies, they should observe the evolution to understand common repercussions of certain kinds of activities. They should also assess the final cohesion (i.e. the IAg sociometric). As some recommended standards, teachers can consider an appropriate cohesion when $IAg \geq 0.040$ in groups around 20 students or $IAg \geq 0.060$ in groups around 10 students.

Since FTS-SOCI is nondeterministic, teachers can run a battery simulations with the same input to reduce the bias of the results occurred by chance, in a second phase of practice. In fact, the obtained average cohesion is more reliable than the particular cohesion of each singular simulation. In this phase, teachers are recommended to set the number of simulations to a number from 10 to 100, and increase the simulation speed to the maximum available option of the tool (currently the maximum available speed is 200 iterations per second).

4. Case study: training of a computer science teacher

The current approach was applied to train a teacher of computer science. The teacher designed several teaching strategies for a Programming subject of the first grade of Computer Science. He analyzed the advantages and drawbacks of each strategy by considering both (a) his own opinion about the right way of teaching the subject with reasonable schedules of activities, and (b) the evolutions and outputs of FTS-SOCI.

Finally, he chose one teaching strategy and he concluded it was probably the most appropriate strategy from the ones he had considered. The schedule of this strategy contained teamwork in pairs from almost the beginning. He observed that the sooner the collaborative activities start, the more cohesion is achieved. The individual activities were kept to minimum, and these were orally presented to the classmates in classes so there were some interactions between students. There was also a final discussion between all the students some days before finishing the subject.

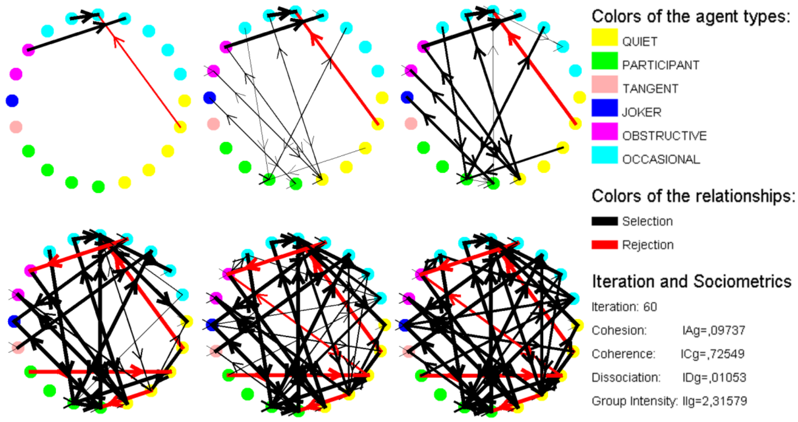

Fig. 1 Evolution of the simulated sociometric status with a teaching strategy for the Programming subject and the recommended distribution of student types

Fig. 1 shows the evolution of the sociometric status with this teaching strategy in FTS-SOCI. One can observe that the simulated cohesion was higher than the recommended standard, so the results advocated that the teaching strategy was appropriate. This was further assessed by running 100 simulations. The average cohesion was 0.0736 (SD=0.0205) in these simulations. Thus, the previous results were confirmed since the simulated average cohesion was also higher than the proposed standard.

4. Conclusions and future work

The current approach supports teachers in training for the particular task of designing teaching strategies in terms of scheduling learning activities. Since candidate teachers normally cannot implement their teaching strategies, simulations can probably be one of the most appropriate solutions.

The current work is planned to be extended in several ways. First, the current approach will be further experienced in more than one teacher, in order to obtain more reliable results about the utility of the current approach. Second, we will explore other interfaces for defining teaching strategies, since the current way of defining these might not be intuitive or easy enough for non-technical teachers.

Acknowledgments

This work has been supported by the project "Diseño curricular de asignaturas con actividades colaborativas asistido con un simulador basado en agentes" with reference PIIDUZ_15_193 and funded by the University of Zaragoza. The current work has also been supported by the program "Estancias de movilidad en el extranjero José Castillejo para jóvenes doctores" funded by the Spanish Ministry of Education, Culture and Sport, in the context of the project "Sistema basado en agentes para simular y trabajar las emociones en los programas de mindfulness" with reference CAS16/00075. This work has also been done in the context of the project "Soluciones tecnológicas para facilitar la práctica de mindfulness: caminando hacia mHealth" with reference TEC2013-50049-EXP and supported by the Spanish Ministry of Economy and Competitiveness.

References

1. Douglas N Harris and Tim R Sass. Teacher training, teacher quality and student achievement. Journal of public economics, 95(7):798-812, 2011.
2. Jere Brophy. Teacher influences on student achievement. American Psychologist, 41(10):1069, 1986.
3. Insung Jung. ICT-pedagogy integration in teacher training: Application cases worldwide. Educational Technology & Society, 8(2):94-101, 2005.
4. Mahbub Hasan, Shahadat Hossain Khan, and Che Kum Clement. Emerging trends of using open source technology for sustainable teacher training programme in Bangladesh. Procedia-Social and Behavioral Sciences, 195:862-871, 2015.
5. Leena Salminen, Marja-Liisa Gustafsson, Liisa Vilén, Pilar Fuster, Natalja Istomina, and Evridiki Papastavrou. Nurse teacher candidates learned to use social media during the international teacher training course. Nurse Education Today, 36:354-359, 2016.
6. Heng-Li Yang and Jih-Hsin Tang. Team structure and team performance in IS development: a social network perspective. Information & Management, 41(3):335-349, 2004.
7. Emilio Serrano and Carlos A Iglesias. Validating viral marketing strategies in twitter via agent-based social simulation. Expert Systems with Applications, 50:140-150, 2016.

8. Iván García-Magariño and Inmaculada Plaza. FTS-SOCI: An agent-based framework for simulating teaching strategies with evolutions of sociograms. Simulation Modelling Practice and Theory, 57:161-178, 2015.
9. Iván García-Magariño, Inmaculada Plaza, Raúl Igual, and Andrés S Lombas. Designing teaching strategies with an agent-based simulator. In Modeling and Simulation in Engineering, Economics and Management, volume 254 of Lecture Notes in Business Information Processing, pages 176-185. Springer, 2016.
10. Fengfeng Ke, Sungwoong Lee, and Xinhao Xu. Teaching training in a mixed-reality integrated learning environment. Computers in Human Behavior, 62:212-220, 2016.
11. Juan E Jiménez and Isabel O'Shanahan. Effects of web-based training on Spanish pre-service and in-service teacher knowledge and implicit beliefs on learning to read. Teaching and Teacher Education, 55:175-187, 2016.
12. Sonja P Brubacher, Martine Powell, Helen Skouteris, and Belinda Guadagno. The effects of e-simulation interview training on teachers use of open-ended questions. Child abuse & neglect, 43:95-103, 2015.
13. Raúl Igual, Juan José Marcuello, Inmaculada Plaza, Iván García-Magariño, and Francisco Javier Arcega. Experiences using free software simulation tools in engineering higher education. Edulearn16 Proceedings, pages 8653-8662, 2016.
14. Jacques Ferber. Multi-agent systems: an introduction to distributed artificial intelligence. Harlow, United Kingdom: Addison-Wesley Reading, 1999.
15. Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In International Workshop on Agent Theories, Architectures, and Languages, volume 1193 of Lecture Notes in Computer Science, pages 21-35. Springer, 1996.
16. Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. The belief-desire-intention model of agency. In International Workshop on Agent Theories, Architectures, and Languages, volume 1555 of Lecture Notes in Computer Science, pages 1-10. Springer, 1998.
17. Ron Sun. Cognition and multi-agent interaction: From cognitive modeling to social simulation. New York, NY: Cambridge University Press, 2006.
18. David Fitoussi and Moshe Tennenholtz. Choosing social laws for multi-agent systems: Minimality and simplicity. Artificial Intelligence, 119(1):61-101, 2000.

19. Luca Cernuzzi, Massimo Cossentino, and Franco Zambonelli. Process models for agent-based development. Engineering Applications of Artificial Intelligence, 18(2):205-222, 2005.
20. Iván García-Magariño, Alma Gómez-Rodríguez, and Juan C González-Moreno. Definition of process models for agent-based development. In International Workshop on Agent-Oriented Software Engineering, volume 5386 of Lecture Notes in Computer Science, pages 60-73. Springer, 2008.
21. Michael Wooldridge, Nicholas R Jennings, and David Kinny. The Gaia methodology for agent-oriented analysis and design. Autonomous Agents and multi-agent systems, 3(3):285-312, 2000.
22. Iván García-Magariño, Celia Gutiérrez, and Rubén Fuentes-Fernández. The INGENIAS Development Kit: a practical application for crisis-management. In International Work-Conference on Artificial Neural Networks, volume 5517 of Lecture Notes in Computer Science, pages 537-544. Springer, 2009.
23. John Thangarajah, Lin Padgham, and Michael Winikoff. Prometheus Design Tool. In Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, pages 127-128. ACM, 2005.
24. Iván García-Magariño, Jorge Gómez-Sanz, and Rubén Fuentes-Fernández. Ingenias development assisted with model transformation by-example: A practical case. In 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), pages 40-49. Springer, 2009.
25. Jorge J Gómez-Sanz, Carlos R Fernández, and Javier Arroyo. Model driven development and simulations with the INGENIAS agent framework. Simulation Modelling Practice and Theory, 18(10):1468-1482, 2010.
26. Paul Davidsson. Agent based social simulation: A computer science view. Journal of artificial societies and social simulation, 5(1), 2002.
27. Emilio Serrano, Pablo Moncada, Mercedes Garijo, and Carlos A Iglesias. Evaluating social choice techniques into intelligent environments by agent based social simulation. Information Sciences, 286:102-124, 2014.
28. Iván García-Magariño, Jorge J Gómez-Sanz, and José R Pérez-Agüera. A multi-agent based implementation of a Delphi process. In Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3, pages 1543-1546. International Foundation for Autonomous Agents and Multi-agent Systems, 2008.

29. Iván García-Magariño. ABSTUR: an agent-based simulator for tourist urban routes. Expert Systems with Applications, 42(12):5287-5302, 2015.
30. Iván García-Magariño, Alma Gómez-Rodríguez, Juan C González-Moreno, and Guillermo Palacios-Navarro. PEABS: A Process for developing Efficient Agent-Based Simulators. Engineering Applications of Artificial Intelligence, 46:104-112, 2015.
31. Antonella Di Stefano and Corrado Santoro. Supporting agent development in Erlang through the eXAT platform. In Software Agent-Based Applications, Platforms and Development Kits, Whitestein Series in Software Agent Technologies, pages 47-71. Springer, 2005.