

A Survey of Agent-Based Modelling and Simulation Tools for Educational Purpose

Krešimir RAKIĆ, Marko ROSIĆ, Ivica BOLJAT

Abstract: Simulation is an experimentation with the imitation or model of the observed system, and observation of its behaviour over time, with the purpose of better understanding and/or improving the system. It is often used in situations where research cannot include the real system because of its inaccessibility, dangerous or unacceptable involvement, the fact that the system is designed but not built yet, in situations where the system is abstract or simply does not exist. Elements of these systems can be implemented as software entities which can perceive their environment and autonomously react to the stimulation, i.e. intelligent agents. These tools, which allow the research of complex natural, social and technical phenomena and systems, are called agent-based modelling and simulation tools. This paper presents a review of some of these modern computer tools that can be successfully implemented in the teaching process.

Keywords: agent-based modelling and simulation; education; intelligent agents

1 INTRODUCTION

The rapid development of information and communication technology (ICT) in the last few decades gave the opportunity to a wider scientific and professional community to approach the problems in previously unimaginable ways. Many natural, social and technical phenomena and systems are difficult to access or are completely inaccessible. They can be dangerous for observation and exploration or abstract and invisible to the human eye. By applying modern computing technologies, it is possible to model and simulate such phenomena and systems and manipulate them so that the system can be analysed and strategies for its functioning evaluated. Elements of these systems can be implemented as software entities which can perceive their environment and autonomously react to the stimulation. This description largely coincides with the standard definition of an intelligent agent. Therefore, this paper deals with agent-based modelling and simulation (ABMS) tools.

This approach can significantly contribute to the understanding of complex phenomena from different areas of life. In doing so, different groups of users emphasize the different aspects of used software tools. Social scientists, without programming experience, want a simple-to-use tool with an intuitive interface. On the other hand, computer scientists are concerned with the type of license and the code openness that will enable them to manipulate the tool. Teachers, who use modelling and simulation tools in teaching, want easy-to-learn packages that enable students to transfer knowledge and experience to real-world situations [32].

Over the years, many different tools have been developed for different purposes and areas of application. The primary purpose of this paper is to provide a comprehensive overview of the segment of these software tools that can be successfully applied in teaching.

This paper begins by defining the concepts in the field of modelling and simulation in Section 2. In Section 3, we continue with pointing out the possibility and good practice of applying simulation in the teaching process. After that, in Section 4 we brought the basic facts of the agent paradigm. Section 5 reports some related past surveys while in Section 6 we give an overview of the state-of-the-art in ABMS simulation tools for educational purpose.

Finally, in Section 7, the implications of this research and concluding remarks are presented.

2 SIMULATION

In everyday life, we often encounter the concept of simulation. Meteorologists regularly display simulations of weather in the days to come. Children, and sometimes adults, play with physical models of cars and roads, trains and railways, ships and, more recently, aircrafts that, remotely controlled, simulate the behaviour of real vehicles. Numerous computer applications simulate a wide range of human activities, which enables us to test our skills as drivers, researchers, urbanists, etc.

From a historical point of view, the simulations were largely independently developed and used in different areas of human work and life. However, the researches in cybernetics and system theory in the 20th century, combined with the ever-increasing use of computers in all these areas led to a certain unification and systematic presentation of the concept of simulation.

Simulation is today an increasingly present methodology for solving problems from different areas of human activity. It is used to describe and analyse the behaviour of the system, to detect the causalities within the system, as well as to assist in the design of realistic systems, both existing and conceptual.

In its most general sense, the simulation can be defined as an imitation of the system.

The imitation represents the attempt to replicate or copy something else. For example, a forger tries to imitate the work of a great artist or attempts to copy the looks of official money bills. For an athlete who behaves as if the foul was committed on him, though it was not the case, we say that he simulates. Computer-aided design (CAD) systems allow imitation of designs of physical artefacts or entire production facilities. However, the key difference between these imitations and the examples described above is that they do not include the time lapse.

Therefore, there is a difference between the concept of static simulation, that imitates the system at a given moment, and the concept of dynamic simulation, which includes the time lapse [1]. The term simulation is mainly used in the context of dynamic simulation and can be

defined as the imitation of the process or real-world system's function over time [2].

Simulation of the system requires the development of a system model. The system model represents the key characteristics, behaviours and functions of the selected physical or abstract system or process. The model, therefore, represents the system itself, while the simulation represents the functioning of the system over time.

Another important aspect of the definition is the purpose of the simulation model. In his general discussion of models, Pidd identifies the need for understanding, changing, managing and controlling reality as the purpose for building the model [3]. Simulation models, specifically, should provide a better understanding of the system and detect possible system improvements. A better understanding of the system as well as identification of possible improvements, are important for future decision-making in the real system.

Another important feature of the model is the emphasis on its simplification. The system model rarely includes all aspects of it, but it concentrates on those that are crucial for the functioning of the system. Even if it is possible to capture all the details, it is probably not desirable because it would take too much time to collect and engage all aspects of the system in the model.

When choosing simulations as a method of system modelling, the purpose of modelling needs to be taken into account. The simulation represents an experimental approach to modelling in which user simulates changes of system inputs, explores alternative scenarios, and observes the behaviour of the model until it gains enough understanding of its work and identifies how it can improve the real system. For example, it is possible to foresee the average waiting time for telephone users in the call centre when a certain number of operators are busy, and the task of the person using the simulation model is to change the input (operator number) and observe the simulation model to determine the effect of the change.

In addition to the described essential aspects of simulation, it is possible to define simulation more precisely as experimentation with simulated (computer) imitation of the observed system and observation of its behaviour over time, with the purpose of better understanding and/or improving the system.

Simulation is often used in situations where research cannot include the real system because of its inaccessibility, dangerous or unacceptable involvement, the fact that the system is designed but not built yet, in situations where the system is abstract or simply does not exist [4].

3 COMPUTER SIMULATION IN EDUCATION

Today, computers are used as an integral part of many classrooms and laboratories, as a teaching and learning tool for real or virtual investigations in various subjects. These applications include using computers to facilitate data acquisition, to provide real-time data display, to analyse these data, and to simulate complex phenomena. Several studies show that this approach is as effective as, or even more effective than its non-computer-based counterparts [5-9]. These studies proved that computer simulations could be a learning tool as productive as hands-on

equipment, given the same curricula and educational setting [10]. Students who used computer simulations instead of real equipment performed better on conceptual questions and developed a greater competency in handling real physical artefacts [11].

As mentioned before, simulation can be effectively and efficiently used in describing systems which are inaccessible or dangerous, systems that normally do not accept students' involvement, abstract systems or systems which do not (yet) exist. These systems can bring particular benefits to the process of learning and teaching.

With simulation environments, we are enabled to have our students vary the force of gravity, explore nuclear fission at the molecular level, move tectonic plates while investigating the differences between divergent and convergent boundaries. Computer simulations give us these interactive, authentic and meaningful learning opportunities. One advantage of a computer-based simulation is the ability to make normally unobservable occurrences plainly visible for the student [12]. Students and researchers can observe, explore, recreate, and receive immediate feedback about real objects, phenomena, and processes that would otherwise be too complex, time-consuming, or dangerous [13].

In a simulated environment, time changes can be speeded up or slowed down; abstract concepts can be made concrete and tacit behaviours visible. Teachers can focus students' attention on learning objectives when real-world environments are simplified, a causality of events is clearly explained, and unnecessary cognitive tasks are reduced through a simulation. Otherwise, when students have many choices within simulations, it represents the potential distraction [14], and too many details in the visualisation of the system hamper the ability to notice relevant details [15]. Another useful feature of simulation is the possibility to employ perceptual cues that would direct learner's attention to critical features and events that are functionally invisible [16].

Simulations provide opportunities for immersive understanding and adaptive exploration of the diverse real-world and constructed environments, thus affording a wide range of exploration opportunities, ranging from the scientific to the social and artistic. [17]

For all the reasons mentioned, simulation has recently become one of the most popular instructional tools for delivering quality teaching. When using realistic simulations to study different phenomena, students are encouraged to apply acquired skills and motivated for advanced learning [18-20]. The experience of working with the simulation environment positively affects the correct adoption of the concepts, and successful application of the learned material [21]. The study conducted with high school students in the United Kingdom, who used the simulation in the classroom, demonstrated that this approach has inspired participants to express scientific thinking and approach to analysis and evaluation of the data collected [22]. The results of these and similar researches point out the great potential of applying technology to teaching and its impact on students' achievements [23-24].

Research has shown that participation in technically enriched teaching can positively influence the attitude of students towards science and encourage them to choose

STEM-oriented careers. Such high-tech, computer-assisted cooperative simulation of a real-life situation helped to trigger application learning as well as professional identity/awareness/interest/construction [25].

4 INTELLIGENT AGENTS AND MULTI-AGENT SYSTEMS

An agent is a physical and/or virtual system that perceives its environment by various sensors and acts on it through its actuators. Agents work in their environment, and can include people, robots, computer programs, but also some simple gadgets.

The software agent is a persistent, goal-oriented computer program that responds to its environment, works without continuous direct surveillance and performs certain functions for the end user or other computer programs in a particular environment. Such independent action implies the power to choose a suitable action in a particular situation if it exists. The agent is inhibited by other processes and agents, but also can be able to learn from its experience of working in the environment over a long period.

Software agents can be independent or work in collaboration with other agents or humans. In interaction with humans, software agents can have qualities characteristic for humans, such as the understanding of natural language and speech, personality, or humanoid embodiment.

Software agents represent an evolutionary step in comparison to conventional computer programs. They can be self-activated and run and they do not require input or interaction with a human user. Software agents can also start, monitor, and shut down other programs or agents.

All agents are computer programs, but not all programs are agents. Key features that differentiate agents from arbitrary programs are, according to Franklin & Graesser [26], reaction to the environment, autonomy, goal-orientation and persistence.

Related and derived terms are:

- intelligent agents who express a certain aspect of artificial intelligence, e.g. learning or concluding,
- autonomous agents who can modify the way they achieve their goals,
- distributed agents that run on physically different computers,
- multi-agent systems, i.e. the group of distributed agents working together to achieve the goal that an individual acting agent alone cannot achieve.

Thus, an agent is an entity that we can define through its perception of the environment in which it is located, by the built-in sensors, and acting in such an environment and on that environment through actuators.

The agent's perception sequence is a complete history of everything that an agent ever perceives. The agent's choice of action at any time may depend only on the last perception or on entire perception sequence so far. In mathematical terms, we can say that agent behaviour is described by an agent function that maps every perception into the corresponding action.

Russel & Norvig listed five basic types of agents characterized by a degree of expressed intelligent behaviour and ability [27]:

- Simple Reflex Agents that select action on the basis of a current observation, ignoring the history of observation;
- Model-Based Reflex Agents that select the action based on the history of observations;
- Goal-Based Agents who choose the action to achieve a specific goal;
- Utility-Based Agents that select the action that will achieve the maximum degree of success;
- All these agents can also be Learning Agents who analyse the experience to choose the action.

Systems in which intelligent agents work, as well as multi-agent systems, are examples of abstract systems. The study of these systems is hardly feasible at the theoretical level. The particular difficulty is a prediction of system behaviour in different situations that the intelligent agent, or several agents in a multi-agent system, perceives by its sensors, and responds to them depending on the defined causal rules. As such, these systems are very suitable for studying and researching using simulation models.

Over the last few years, with the growing interest in using an ABMS tool, numerous researches of available tools have been conducted. In the next section, we give a review of relevant past surveys.

5 RELEVANT SURVEYS

Widespread use of computer agents that, with ubiquitous digitization, slowly penetrate into all spheres of life also caused the need for investigating the behaviour of the systems in which individual agents or groups of agents act autonomously. The number of environments that enable modelling and simulation of such agent-based systems is constantly increasing.

This motivated numerous authors to conduct screening of available tools, focusing on different aspects of system modelling and simulation. In an extensive research of relevant databases such as IEEE Xplore, Google Scholar, Scopus, ScienceDirect, and SemanticScholar, using the terms: "*agent-based modelling and simulation*", "*agent-based simulation environment*", "*agent toolkit*", "*state-of-the-art*", "*comparison*", "*review*" and "*survey*", we have identified seven surveys that have been conducted in last 20 years, a period of growing interest for research in the field of agent paradigm. We will list those surveys chronologically

In their review, Serenko & Detlor [28] explore tools for modelling and simulation of agent-based systems based on their application as pedagogical tools for teaching purposes. They have classified 20 available tools by looking at four features: the ability to create mobile agents, the ability to develop a multi-agent system, the ability to create different types of agents for different purposes, and the ability to retrieve information. They also explored the basic programming language used in these tools. The authors conducted a user satisfaction survey about tool functionality, performance, and interaction with the user, involving 87 teachers. Our work in this paper is similar in that we are trying to explore tools that can be successfully used in the teaching process.

In 2004, Tobias & Hofmann [29] discussed four open code tools: Repast, Swarm, Quicksilver and VSEit, and evaluated them based on 19 characteristics, including

general criteria, modelling and experimentation, modelling support, and modelling options. They ranked the tools by awarding points according to each criterion. The main limitation of this review is the fact that authors are limited to tools that use Java as the main programming language, and are mostly used by the social sciences community.

The survey conducted by Railsback, Lytinen & Jackson, published in 2006 [30], compares four major platforms: NetLogo, Mason, Repast and Swarm. The authors have created a template, called *StupidModel*, which they used to evaluate and compare these tools on multiple levels. With each new level, they added more features and, through a total of 15 levels, authors investigated different features such as environmental issues, model structure, agent scheduling, file entry and output, random number generation, and statistical capabilities. The main limitation of this review is a small number of tested platforms and their similar nature. In this paper, we will expand the number of described tools, avoid their ranking and allow the user to choose which tool is most appropriate for his project.

In the next major survey, conducted in 2006, Castle & Crooks [31] investigated eight simulation platforms. Particularly they focus on the assessment of their geospatial capabilities. They also paid attention to the date of creation, the implementation language, required programming experience and the availability of demonstration models and tutorials. This review also compares similar tools specialized for use in social science researches.

Significantly increasing the number of observed simulation tools, Nikolai & Madey [32] examine their five characteristics. With a total of 53 tools, they compared the programming language required to create a model or simulation, the operating system required to run the tool, the type of license, the primary domain for which the tool is intended, and the degree of support available to the user of the tool. Their intent was to provide users with enough information to choose the most suitable tool in the form of an easy-to-use compendium.

Based on personal experience and available information, Allen [33] recalled the significant features of existing tools. The author has divided the tools into the simulation environments for individual intelligent agents (31 tools) and those designed to simulate multi-agent systems (13 tools). It points to important shortcomings such as: difficult of use; insufficient tools for building, especially tools for representing space; insufficient tools for executing and observing simulation experiments; and the lack of tools for documenting and communicating software. This survey also lists general application areas: Biology and Medicine, Physics and Chemistry, Security and Cyber Security, Environment, Social and Economic Modelling, and Supply Network and Transport Optimisation.

In the most comprehensive survey of ABMS tools, Abar et al. [34] reviewed 85 existing tools. The authors present a concise characterization of existing tools, including the underlying programming language, the type of agent that the system supports, the complexity of the end user system, and the domain for which the system is primarily intended, highlighting their advantages and disadvantages. In this way, they provide a useful reference

for engineers, researchers, students and teaching staff for choosing the appropriate ABMS tool, when designing and developing their models and prototype systems.

Referring to most of the above-mentioned reviews, Wikipedia [35] provides another updated overview of available tools, including their primary domain, institution developing them, programming language and licensing, user documentation, 3D and GIS capabilities, and compliance with applicable standards.

A brief overview of the above-mentioned surveys is given in Tab. 1.

Table 1 Overview of Previous Surveys

| Authors | Year | Number of Tools | Purpose of Tools |
|---|---------------|-----------------|-------------------------|
| Serenko, Detlor | 2002 | 20 | Education |
| Tobias, Hoffman | 2004 | 4 | Social Sciences |
| Railsback, Lytinen, Jackson | 2006 | 4 | General Purpose |
| Castle, Crooks | 2006 | 8 | Geospatial Capabilities |
| Nikolai, Madey | 2009 | 53 | General Purpose |
| Allen | 2010 | 31+13 | General Purpose |
| Abar, Theodoropoulos, Lemariner, O'Hare | 2017 | 85 | General Purpose |
| Wikipedia | 2019 (online) | 25 | General Purpose |

The above-mentioned researches provide useful information on a wide range of ABMS tools. They also give an insight into the period of active use of certain tools, which also indicates their objective quality. In this review, we will pay special attention to tools that can be successfully used for teaching purposes, when students are introduced to abstract concepts of the agent paradigm.

6 SURVEY OF ABMS TOOLS IN EDUCATION

As a basis for the selection of ABMS tools for this survey, we took the latest and most comprehensive surveys mentioned earlier [34, 35], as well as some tools that are not covered by these surveys, but seem interesting and very applicable. From the wide range of available software platforms, we have selected those tools that list education as a possible field of application.

Guided by the goal that this survey can be used as an ABMS tool catalogue, and knowing the limitations of using software tools for teaching purposes, we will list the basic characteristics of each tool and classify them by several categories that, by our experience, play an important role in choosing the teaching and learning software.

One of the categories is a type of software tool license. This feature is of great importance due to the financial aspect of organizing the instructions.

Other important categories are the programming language for modelling and simulation of the system, as well as the development environment of the tool. Getting acquainted with the programming language with which students did not have any contact until then would be time-consuming. A complicated development environment would also require too much time. In this way, there would

not be enough time for a quality teaching of the agent paradigm, which would be the focus of attention for this kind of instruction practice.

We also paid attention to the platform on which ABMS tools are running, as well as the required operating system. These characteristics represent another significant item from the organizational point of view.

As the last characteristic, for each of the tools, we will indicate the level of complexity of the model development. This feature also has a significant impact on the liability to engage some ABMS tool in teaching and learning process.

We are aware that the selection of listed characteristics is, at the same time, a limitation of this survey. Emphasis on the general characteristics of this group of software tools has potentially omitted the more subtle features indigenous to each package. Sometimes, in research of specific domains and with specific modelling requirements, these particular characteristics can be crucial for selecting a particular tool. However, we hope that this survey will serve as a starting point for educators who want to include some of the ABMS tools in their teaching process.

Before the brief overview of the selected AMBS tools' characteristics, we will give some basic information for each of them. The tools we choose on the basis for their applicability as a teaching tool are:

- **AgentScript** is a minimalist system-modelling framework, based on agents. This tool belongs to a group of tools based on NetLogo agents' semantics. Its goal is to promote an agent-oriented programming model in a deployable CoffeeScript/JavaScript implementation.
- **AgentSheets** is an authoring tool that empowers casual computer users with no formal programming training to build and publish web-based interactive simulation. This tool combines Java authoring, end-user programmable agents and spreadsheet technology.
- **BehaviourComposer** is a web-based tool for constructing, running, visualising, analysing, and sharing agent-based models. These models can be constructed by non-experts by composing pre-built modular components called microbehaviours, small, coherent, and independent program fragments.
- **Cellular** is a block-based programming environment for creating agent-based simulations. Different types of agents can be created, and their behaviour controlled by constructing scripts using a drag-and-drop interface.
- **ExtendSim** is a simulation environment for modelling continuous, discrete event, discrete rate, and agent-based systems. Its design facilitates every phase of the simulation project, from creating, validating, and verifying the model, to the construction of a user interface. An integrated database enables faster and handling with simulation models.
- **Framsticks** is a system built to support a wide range of experiments, and to provide all of its functionality to users, who may use this open system in a variety of ways. This tool tries to fill the gap between advanced artificial life models with their consequences, and advanced simulation tools with their realism.

- **JAS-mine** is a Java-based computational platform that features tools to support the development of large-scale, data-driven, discrete-event simulations. The platform is specifically designed for both agent-based and microsimulation modelling, anticipating a convergence between the two approaches.
- **MIMOSE** is a modelling and simulation software system that consists of a model description language and an experimental frame for simulation of the described models. The purpose of MIMOSE project was the development of a modelling language that considers special demands of modelling in social science and supports the creation of structured, homogeneous simulation models.
- **MOBIDYC** is a system that helps the user in the four main steps involved in the construction and the use of models for population dynamics: defining biological entities, defining their environment, launching and controlling simulations, editing results. It includes tools for every aspect of agent functioning.
- **NetLogo** is a multi-agent programming language and modelling environment for simulating natural and social phenomena. It is particularly well suited for modelling complex systems evolving over time. Despite the entry-level programming interface, NetLogo is capable of quite sophisticated modelling and allows experienced users to add their own Java extensions.
- **Scratch** is a visual programming environment that lets users create interactive, media-rich projects. These projects include animated stories, games, music videos, science projects, tutorials, simulations, and sensor-driven art projects.
- **SeSAM** provides a generic environment for modelling and experimenting with agent-based systems. Based on a declarative, explicit model representation and visual programming, it allows implementing models on specification level, including the easy construction of complex models.
- **SimSketch** is an integrated drawing and modelling tool that allows students to create sketches and apply behaviours to elements of their drawing. A multi-agent simulation engine interprets and executes the model, thus building an intuitively usable and motivating learning environment and confronting the learner with the results and consequences of his externalised mental model.
- **SIMUL8** is a software tool for discrete event simulation that allows user to create a visual model of the system under investigation by drawing objects directly on the screen. The tool incorporates programming language and model visualization capabilities that enable it to create accurate, flexible, and robust simulation more rapidly.
- **StarLogoNova** is a block-based programming environment that allows users to create 3D simulations and games. This environment is designed for users to explore the collective behaviour of decentralized agents. They can drag and drop blocks to construct graphical scripts that control how individual agents respond to stimuli.
- **StarLogo TNG** is designed to facilitate the development and study of simulated systems in

classrooms. It includes a graphical programming language and a game-inspired three-dimensional world. The graphical programming environment reduces the learning curve of programming, while 3D graphics gives more immersive and engaging experience for users.

- **Sugarscape** is a bottom-up defined world of agents, where agents are heterogeneous from the point of view

of individual abilities (vision) and needs (metabolism). It has most of the important components of social simulation. Agents in SugarScape have a number of attributes such as vision and metabolism and are capable of adapting to varying environments.

An overview of the key features of the ABMS tools listed above is given in Tab. 2.

Table 2 Overview of ABMS Tools Key Features

| ABMS Tool | License Type | Coding Language / API | IDE | Operating system | Platform | Complexity |
|---------------------------|--------------|---------------------------|------------------------------|---|----------------------------------|-------------|
| AgentScript | LGPL | Model libraries | CoffeeScript within browser | Windows, Mac OS X, Linux, Unix | Desktop computers, Smart tablets | Simple/Easy |
| AgentSheets | Proprietary | Visual AgenTalk | Visual Programming Interface | Windows, Mac OS X, Linux, Unix | Desktop computers | Simple/Easy |
| BehaviourComposer (2D/3D) | BSD | Model libraries, Ajax | Visual Model Development | Windows, Mac OS X, Linux, Unix | Desktop computers, Smart tablets | Simple/Easy |
| Cellular (2D/3D) | AGPLv3 | Graphical blocks | Visual Programming Interface | Windows, Mac OS X | Desktop computers | Simple/Easy |
| ExtendSim (2D/3D) | Proprietary | Model libraries | Interactive GUI | Windows, Mac OS X | Desktop computers, Workstations | Moderate |
| Framsticks (2D/3D) | GPL, LGPL | FramScript | Visual GUI | Windows, Mac OS X, Linux | Desktop computers, Workstations | Simple/Easy |
| JAS-mine | LGPL | Built-in libraries | Eclipse | Any Java-enabled OS | Desktop computers | Simple/Easy |
| MIMOSE | GPL | Model description | GUI | Windows, Linux, Solaris | Desktop computers | Moderate |
| MOBYDIC | GPL | Smalltalk | Template-based GUI | Windows, Mac OS X, Linux, Unix | Desktop computers | Simple/Easy |
| NetLogo (2D/3D) | GPL | Model libraries | N/A | Windows, Mac OS X, Linux, Unix | Desktop computers | Simple/Easy |
| Scratch (2D/3D) | CC-BY-SA 2.0 | Graphical blocks | Visual Programming Interface | Any browser with Adobe Flash Player | Desktop computers | Simple/Easy |
| SeSAM | LGPL | Visual modelling language | N/A | Windows, Mac OS X, Linux | Desktop computers | Simple/Easy |
| SimSketch | MIT | Java Web Start | N/A | Windows, Mac OS X, Linux | Desktop computers | Simple/Easy |
| Simul8 | Proprietary | Visual Logic | Drag & Drop Interface | Windows, Mac OS X with VM | Desktop computers, Workstations | Moderate |
| StarLogo Nova | CC BY-SA 4.0 | Graphical blocks | Visual Programming Interface | Windows, Mac OS X, Linux, Unix, Solaris | Desktop computer | Simple/Easy |
| StarLogo TNG | SLTNG v1.0 | Graphical blocks | Visual Programming Interface | Windows, Mac OS X | Desktop computer | Simple/Easy |
| Sugarscape | GPL | Java | Visual Model Development | Any Java-enabled platform | Desktop computer | Simple/Easy |

This survey is part of a broader research of teaching agent paradigm using simulation environments. Taking into account the above characteristics of ABMS tools, as well as the specific needs of our research, from the group of listed tools we selected Cellular and NetLogo. The main reason for this choice was the fact that students were already familiar with the programming languages of the same or very similar syntax.

With this choice, of course, we do not want to claim that the chosen ABMS tools are better than other previously mentioned tools. We leave the choice of appropriate ABMS tool to each researcher individually, in accordance with his needs and the purpose of the research.

7 CONCLUSION

The increasing use of ICT in all aspects of life as well as the need to optimize different systems and processes using computer models and simulations, in which intelligent agents play a key role, justify the use of ABMS tools in education, to give future experts an insight into the possibilities of these software solutions. This paper provides an insight into the range of available modelling and simulation tools available for agents that can be used

in the learning and teaching process. The paper also outlines the essential features of these tools that, according to our experience, can play a key role in choosing one of them. That is why we believe this work can serve as a reference book for the teachers at all levels of the education system when choosing the most appropriate tool for their work.

8 REFERENCES

- [1] Law, A. M. & Kelton, W. D. (2000). *Simulation Modeling and Analysis*. New York, NY: McGrawHill.
- [2] Banks, J., Carson, J., Nelson, B., & Nicol, D. (2001). *Discrete-Event System Simulation*. Upper Saddle River, NJ: Prentice Hall.
- [3] Pidd, M. (2003). *Tools for Thinking: Modelling in Management Science*. Hoboken, NJ: John Wiley & Sons.
- [4] Sokolowski, J. A. & Banks, C. M. (2009). *Principles of Modeling and Simulation*. Hoboken, NJ: John Wiley & Sons. <https://doi.org/10.1002/9780470403563>
- [5] Steinberg, R. N., Oberem, G. E., & McDermott, L. C. (1996). Development of a Computer-Based Tutorial on the Photoelectric Effect. *American Journal of Physics*, 64(11), 1370-1379. <https://doi.org/10.1119/1.18360>
- [6] Zacharia, Z. & Anderson, O. R. (2003). The Effects of an Interactive Computer-Based Simulation prior to Performing

- a Laboratory Inquiry-Based Experiment on Students' Conceptual Understanding of Physics. *American Journal of Physics*, 71(6), 618-629. <https://doi.org/10.1119/1.1566427>
- [7] Liu, A., Newsom, J., Schunn, C., & Shoop, R. (2013). Students Learn Programming Faster Through Robotic Simulation. *Tech Directions*, 72(8), 16-19.
- [8] Liu, C. C., Cheng, Y. B., & Huang, C. W. (2011). The Effect of Simulation Games on the Learning of Computational Problem Solving. *Computers & Education*, 57(3), 1907-1918. <https://doi.org/10.1016/j.compedu.2011.04.002>
- [9] Rutten, N., Van Joolingen, W. R., & Van Der Veen, J. T. (2012). The Learning Effects of Computer Simulations in Science Education. *Computers & Education*, 58(1), 136-153. <https://doi.org/10.1016/j.compedu.2011.07.017>
- [10] Triona, L. M. & Klahr, D. (2003). Point and Click or Grab and Heft: Comparing the Influence of Physical and Virtual Instructional Materials on Elementary School Students' Ability to Design Experiments. *Cognition and Instruction*, 21(2), 149-173. https://doi.org/10.1207/S1532690XCI2102_02
- [11] Finkelstein, N. D., Adams, W. K., Keller, C. J., Kohl, P. B., Perkins, K. K., Podolefsky, N. S., Reid, S., & LeMaster, R. (2005). When Learning about the Real World is Better Done Virtually: A Study of Substituting Computer Simulations for Laboratory Equipment. *Physical Review Special Topics: Physics Education Research*, 1(1), 1-8. <https://doi.org/10.1103/PhysRevSTPER.1.010103>
- [12] Windschitl, M. A. (1995). Using Computer Simulations to Enhance Conceptual Change: The Roles of Constructivist Instruction and Student Epistemological Beliefs (Doctoral dissertation). Retrieved from: <https://lib.dr.iastate.edu/rtd/15946>
- [13] Dwyer, F. M. (1978). *Strategies for Improving Visual Learning: A handbook for the Effective Selection, Design, and Use of Visualized Materials*. State College, PA: Learning Services.
- [14] Chandler, T. N. & Chaille, C. (1993). Process Highlighters in a Computer Simulation: Facilitation of Theory-Oriented Problem Solving. *Journal of Educational Computing Research*, 9(2), 237-263. <https://doi.org/10.2190/3GGG-K9CC-19XC-9G0W>
- [15] Bell, R. L. & Smetana, L. K. (2008). Using Computer Simulations to Enhance Science Teaching and Learning. *National Science Teachers Association: Technology in the Secondary Science Classroom*, 3, 23-32.
- [16] Carlsen, D. D. & Andre, T. (1992). Use of a Microcomputer Simulation and Conceptual Change Text to Overcome Student Preconceptions about Electric Circuits. *Journal of Computer-Based Instruction*, 19(4), 105-109.
- [17] Woolf, B. P. (2010). A Roadmap for Education Technology. Retrieved from: <https://hal.archives-ouvertes.fr/hal-00588291/document>
- [18] Hsu, Y.-S. & Thomas, R. A. (2002). The Impacts of a Web-Aided Instructional Simulation on Science Learning. *International Journal of Science Education*, 24(9), 955-979. <https://doi.org/10.1080/09500690110095258>
- [19] Lewis, E., Stern, J., & Linn, M. (1993). The Effect of Computer Simulations on Introductory Thermodynamics Understanding. *Educational Technology*, 33(1), 45-58.
- [20] Weller, J. M. (2004). Simulation in Undergraduate Medical Education: Bridging the Gap between Theory and Practice. *Medical Education*, 38, 32-38. <https://doi.org/10.1111/j.1365-2923.2004.01739.x>
- [21] Spinello, E. F. & Fischbach, R. (2004). Problem-Based Learning in Public Health Instruction: A Pilot Study of an Online Simulation as a Problem-Based Learning Approach. *Education for Health*, 17(3), 365-373. <https://doi.org/10.1080/13576280400002783>
- [22] Jarvis, T. & Pell, A. (2005). Secondary Pupils of Different Abilities Response to an e-Mission Simulation of the Montserrat Volcanic Eruption. Paper presented at the American Education Research Association, Montreal, CA.
- [23] Chen, C. H. & Howard, B. (2010). Effect of Live Simulation on Middle School Students' Attitudes and Learning toward Science. *Educational Technology & Society*, 13(1), 133-139.
- [24] Jonassen, D. H. (2003). Using Cognitive Tools to Represent Problems. *Journal of Research on Technology in Education*, 35(3), 362-379. <https://doi.org/10.1080/15391523.2003.10782391>
- [25] Kim, B. & Reeves, T. C. (2007). Reframing Research on Learning with Technology: In Search of the Meaning of Cognitive Tools. *Instructional Science*, 35(3), 207-256. <https://doi.org/10.1007/s11251-006-9005-2>
- [26] Franklin, S. & Graesser, A. C. (1996). Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents, In Müller, J. P., Wooldridge, M. J., & Jennings, N. R. (Eds.) *Intelligent agent III: Agent Theories, Architectures, and Languages ECAI'96 Workshop*, 21-35. Budapest, Hungary: Springer. <https://doi.org/10.1007/BFb0013570>
- [27] Russell, S. J. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. New Jersey, NJ: Prentice Hall.
- [28] Serenko, A. & Detlor, B. (2002). Agent Toolkits: A General Overview of the Market and an Assessment of Instructor Satisfaction with Utilizing Toolkits in the Classroom (Working paper No. 445). School of Business, McMaster University, Hamilton, CA.
- [29] Tobias, R. & Hofmann, C. (2004). Evaluation of Free Java-Libraries for Social-Scientific Agent Based Simulation. *Journal of Artificial Societies and Social Simulation*, 7(1).
- [30] Railsback, S. F., Lytinen, S. L., & Jackson, S. K. (2006). Agent-Based Simulation Platforms: Review and Development Recommendations. *Simulation*, 82(9), 609-623. <https://doi.org/10.1177/0037549706073695>
- [31] Castle, C. J. & Crooks, A. T. (2006). Principles and Concepts of Agent-Based Modelling for Developing Geospatial Simulations (Working paper No. 110). Centre for Advanced Spatial Analysis, University College London, London, UK.
- [32] Nikolai, C. & Madey, G. (2009). Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation*, 12(2).
- [33] Allan, R. J. (2010). Survey of Agent Based Modelling and Simulation Tools (Technical report, ISSN 1362-0207). Science & Technology Facilities Council, Warrington, UK.
- [34] Abar, S., Theodoropoulos, G. K., Lemarini, P., & O'Hare, G. M. (2017). Agent Based Modelling and Simulation Tools: A Review of the State-of-Art Software. *Computer Science Review*, 24, 13-33. <https://doi.org/10.1016/j.cosrev.2017.03.001>
- [35] Comparison of Agent-Based Modeling Software. Retrieved from: https://en.wikipedia.org/wiki/Comparison_of_agent-based_modeling_software

Contact information:**Krešimir RAKIĆ**

(Corresponding author)

Faculty of Mechanical Engineering, Computing and Electrical Engineering,
University of Mostar,
Matice hrvatske b.b., 88000 Mostar, Bosnia and Herzegovina
E-mail: kresimir.rakic@fsre.sum.ba

Marko ROSIĆ, Full Professor

Faculty of Science, Department of Informatics, University of Split,
Ruđera Boškovića 33, 21000 Split, Croatia
E-mail: marko.rosic@pmfst.hr

Ivica BOLJAT, Associate Professor

Faculty of Science, Department of Informatics, University of Split,
Ruđera Boškovića 33, 21000 Split, Croatia
E-mail: ivica.boljat@pmfst.hr