

The Role of Agent-Based Simulation in Education

Andrew Koster¹(✉), Fernando Koch², Nicolas Assumpção³, and Tiago Primo⁴

¹ Artificial Intelligence Research Institute (IIIA-CSIC),
Universitat Autònoma de Barcelona, Barcelona, Spain
andrew@iiia.csic.es

² Korea University, Seoul, Republic of Korea
fkoch@acm.org

³ Instituto de Pesquisas Eldorado, Campinas, SP, Brazil
nicolasriccieri@yahoo.com.br

⁴ SAMSUNG Research Institute, Campinas, SP, Brazil
tiago.t@samsung.com

Abstract. Agent-Based Modelling and Simulation (ABMS) is a research methodology for studying complex systems that has been used with success in many social sciences. However, it has so far not been applied in education research. We describe some of the challenges for applying ABMS in the area of education, and discuss some of the potential benefits. We describe our proof-of-concept model that uses data collected from tablet-based classroom education to model how students interact with the content dependent on their engagement level and other properties, how we are calibrating that model, and how it can be validated.

1 Introduction

Agent-Based Modelling and Simulation (ABMS) is a methodology for studying complex systems, and is employed in diverse fields such as economics [30], social sciences [9], biology [13] and transport engineering [3]. [17] describe ABMS as being “particularly suitable for the analysis of complex adaptive systems and emergent phenomena”. Despite pedagogy studying what can clearly be described as a complex system [5, 15, 16], ABMS has, to our knowledge, not been applied in studying education.

There are a number of possible reasons for this, but in our opinion the most prominent one is that to apply ABMS, one needs a lot of available data, and a sufficient understanding of intermediate constructs and how they can be extracted from the data. This type of research is performed in the twin fields of Learning Analytics and Educational Data Mining, which have only recently reached sufficient maturity for ABMS to be a valid approach in researching various aspects of education. In this work, we aim to promote ABMS as a methodology for pedagogical research; in particular when studying classroom environments. We will give an overview of the state-of-the-art in ABMS, and describe some ways in which it could be applied to education research. Moreover, we will present a

proof-of-concept ABMS for modelling a classroom education environment, and how we can use it in both the research and practice of education.

2 Background

Agent-Based Modelling and Simulation (ABMS) is a computational research methodology to study complex phenomena. In particular, an *agent* in this methodology is an autonomous (computational) individual that acts within an *environment*. The aim is (usually) to model the agents and their interactions, and study the resulting emergent properties of the system. It has long been known that very simple systems can portray complex behaviour. Probably the most well-known example of this is Conway's game of life [10], in which a cellular automaton with four simple rules can, dependent on the starting scenario, have such complex behaviour that people are still discovering new patterns.

ABMS can be seen as the natural extension of such cellular automata, in that it considers the system as being distributed: rather than the system having a single state, each entity (agent) has a state, and agents decide individually (and asynchronously) on their next action, rather than the system as a whole moving from state to state. Nevertheless, they have in common that both ABMS and cellular automata are bottom-up (aka micro) models, where the behaviour of the system emerges from the behaviour of individuals, as opposed to macro models, which aim to (usually mathematically) describe the system as a whole without worrying about individuals. Because ABMS allows for distinct types of agents within the system, it is rapidly gaining popularity in biology and social science research, where describing the entire system in a set of mathematical equations is often too complex to be of much use, if possible at all, whereas it is possible to build sufficiently realistic models of individuals and how they interact with each other, and study properties of the system that emerge from the behaviour of many individuals.

For the development of an ABM, it is necessary to be able to calibrate and validate it [4]. The former refers to the process of tuning the model to best correspond to real data. In particular, this requires that there is detailed data on individuals' behaviour in order to calibrate the agent models, data on their interactions, in order to calibrate the interaction models, and finally, data on the emergent properties of the system in order to validate the model. The validation of an ABM, or any model for that matter, is a complicated issue and there is ongoing research on how best to approach this [14, 24], but without doubt, it is only possible with sufficient real data against which to validate.

Up until very recently, such data was not available for the field of education. However, over the last decade, many areas of education have been making increasing use of technology: (i) on the administrative side to keep track of students, their grades, attendance rates, etc. (ii) at the course level, where Learning Management Systems (LMS) applications help teachers distribute course material, collect assignments and perform many of the day-to-day tasks of administering a course; and (iii) at the class level, mostly in the form of Massive Open Online Courses (MOOC), where teachers provide a digital course for students

to follow worldwide (usually completely separate from the normal curriculum). All such applications generate, and store, detailed data at the level they are designed for. An administration system will store a history of students and classes throughout time, whereas an LMS tracks students through a course, in combination with the material each student accessed, and created, throughout the course. Finally, MOOCs and other systems within a class collect detailed usage data for each student, logging each action a user takes throughout the class. The use of such data to perform data-driven analysis of pedagogical measures is a development of the last few years.

In particular, *Learning Analytics* is defined as: “the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs.” —Siemens [29]. Learning Analytics and Educational Data Mining are both relatively young, but fast-growing areas of research that focus on finding algorithmic methods for quantifying various metrics that can model a student’s learning process. There is a large overlap in the two areas, both in goals and in methodologies, with the differences between the areas being mostly in the philosophical and historical roots of the fields [28]. However, as Gašević et al. [11] point out, in accordance with our own findings, “there has been a dearth of empirical studies that have sought to evaluate the impact and transferability of this initial work across domains and context”, and “instructors expressed their preferences of learning analytics features that offer insights into learning processes and identify student gaps in understanding over simple performance measures”. Eberle et al. [8] discuss 12 grand challenge problems for Technology-Enhanced Learning. Among these challenges, we distinguish the similar themes: they identify challenges on how to model students using continuous temporal data, and how to use the data in providing appropriate feedback to empower teachers. In short, while there is a significant amount of work [7, 25–27] into descriptive and predictive measures, what is missing is how different measures can be combined into a theoretically sound, data-driven model of student learning.

We propose that Agent-Based Modelling for Education can address this question.

3 ABMS for Education

The integration of cognitive theory and agent-based approaches in technology-enhanced learning has so far been the particular domain of Intelligent Tutoring Systems (ITS) [12]. In this approach, the focus is on individual learning, and the student is helped by an artificial tutor. Cognitive theories are employed to model user knowledge, and a diverse range of AI techniques, including intelligent agents and recommender systems are used to decide how the tutor should act, what learning object to present to the user, etc. An example of a modern ITS approach is SimStudent [22], which uses machine learning techniques to update a rule-based agent system to adapt to students’ needs. A similar approach is presented by López Bedoya et al. [21], which uses a combination of computational techniques to plan, and replan coursework based on heterogeneous characteristics

of the students. Such systems serve a different purpose from what we propose: rather than attempting to model students as socio-cognitive systems in order to understand them, these systems model students as parameters for adaptive computational tutors.

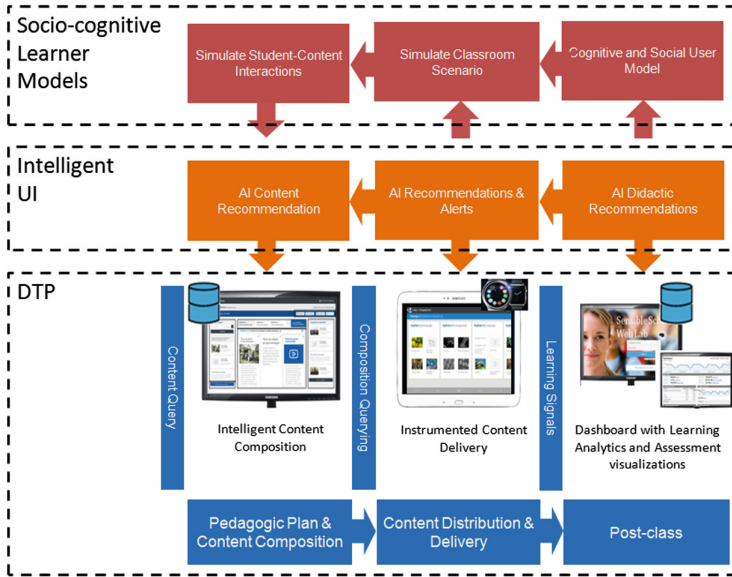


Fig. 1. Overview of components of a digital teaching platform with intelligence models

Figure 1 gives the outline of components in a contemporary system for technology-enhanced learning and outlines what kind of models can help improve our understanding at each stage of education. We see that at the individual level, the modelling follows a cognitive modelling approach [1], and the aim, as best as possible, is to model the psychology of learning, and understand such concepts as creativity, learning styles, engagement and performance. This usually treats students as individuals in isolation, relying for instance on data from MOOCs where students can mostly be considered to be working through a course alone; interacting only with the content and not (much) with their fellow students or the educator. A system comprised of multiple agents like this could model a classroom. In this model, in addition to individual properties, the social interactions come into play. We distinguish between implicit and explicit social interactions. Implicit interactions are those that arise from individual actions that predicate a response. For instance, the teacher changes pages, and expects the students to follow suit. Explicit social interactions are those actions that are directly aimed at a social goal, such as discussing an item, asking a question, solving an exercise together. MOOCs and LMS systems attempt to capture many such interactions, although it is often impossible to do so: if interactions happen outside the system, the data is usually not available. The emergent properties from such a system

can also be studied at two levels: firstly, the interactions can have an effect on individual students, and secondly we can study properties of the classroom, both as aggregates over the individuals, like average performance or engagement, and properties that only arise at the level of a classroom, such as sociability. Finally, we can study how such systems are situated in the environment; the environment is generally described as everything that is modelled, but is not an agent. In the education scenario this is the course material and the possibilities for students and educator to interact with it, but also external parameters, such as noise level, light level, temperature, time of day and day of week, etc. With regards to the content, the data that is most easily manipulated by a computational model is the metadata, which describes properties of the actual course material, such as whether it is textual, a video, or interactive content, what subject(s) the content is related to, and for what level it is intended. By modelling the content, and other aspects of the environment, one can answer other types of questions, such as whether the same exact class at a different time of day would have led to better overall performance? Or what effect switching textual content for video content might have. This is the use of ABMS we have started to study, and we present our proof-of-concept class simulator in the next Section. However, we can think of other potential uses of ABMS for education.

ABMS can also be applied at a higher level of abstraction, in order to assist in policy decisions about educational institutions, teaching methodologies and content. Students might be modeled as having to decide what institute to study at, taking into account the policies and teaching philosophies. Or the institutes themselves can be modeled as agents. The modelling of such macro-level phenomena requires more administrative and demographic data.

A different, but complementary, approach for using ABMS in education is to use the ABMS as an educational tool for modelling other systems (e.g. a biological ecosystem). While that is not the use of ABMS we are interested in this work, it may nevertheless be insightful to give students control over an ABMS simulating their own situation, and allow them to experiment with what types of interventions might improve their own, their class's or their school's performance.

4 Proof-of-Concept Model of a Classroom

We built the proof-of-concept Classroom Simulator to model a classroom in which the students are following the teacher during a lecture. We model students interactions' based on different educational contents and teacher characteristics, however we do not take explicit social interactions into account. In simulating a classroom, the model is composed of three different steps: (I) Initialization, (II) Generate Signals, (III) Content Analysis and optimization. This aims to simulate the actual data that is *output* by a tablet-based digital teaching platform that we developed [18] for use in classrooms. This system generates detailed usage logs for each student, and the teacher. The idea of the model is to be able to generate such usage logs artificially, and simulate some specific types of interventions.

In particular, we aim at modelling teacher-oriented interventions, such as changing the type of material or teaching style. The simulator focuses on a subset of the log data, specifically those related to navigation through the content.

Step (I), generates the variation of students signals' based on the characteristic of the teacher, the content and a class profile. Input in this step are the duration of a class, the educational content profile, the students' profile and the teacher profile. The educational content profile is the number of different contents, their difficulty, their type (image, video, text or assessment) and their order. The teacher profile is related to this content: how orderly he goes through the material and what types of content he spends more time on. Students' profiles are generated stochastically, by drawing random samples from a class profile. The class profile defines the number of students, and normal distributions for engagement (the amount of time a student spends working on the same content as the teacher), activity (the number of actions per time step), and orderliness (whether they navigate through the content linearly from start to finish, or a haphazardly fashion that jumps between contents and navigates backwards) of the students. Furthermore, we simulate eye tracking time series data about whether an individual student is looking at, or away from the tablet at every time stamp. All of these parameters can also be extracted from the real usage logs generated by our tablet-based system. In addition to these profiles, a set of rules are generated (also adding random noise in the generation process) based roughly on the *possible* values each of the different parameters might take for each student, and the states these create within the system, the next state for any student. An example of such a rule, for a student, is the following:

IF

the student has the profile of being highly engaged, medium active and highly orderly;
 the student's recent history of actions are that he or she is looking towards the tablet, and has been studying the first content (a difficult video) for at least 10 time steps;
 and the teacher is looking at the second content (an image)

THEN

the student will change pages to the second content.

Rules for the teacher are simpler, as they only take the teacher's actions into account, whereas we assume that students are influenced by what the teacher is doing (while the reverse is also possible, we disregard that form of feedback at this stage). Even so, for both the students and the teacher, there is the possibility for combinatorial explosion in creating the rules, so we instead generate them randomly without overlap until a sufficient (parameter of the model) number have been generated.

In Step (II), the class is simulated. All students, and the teacher, are initialized on page 0. For each time step, the teacher and each student is evaluated: for each agent the list of rules are run through, and a distance from the agent's state

to the state described in the precondition of the rule is computed. The rule with the minimum distance is selected, and if this distance is lower than a threshold, the rule is executed and the agent transitions to a new state. The output of the simulation is a log of the events that occur during the simulation run.

In Step (III), we can *calibrate* (and also *validate*) the model by analysing the simulated log with regards to student engagement, orderliness and activity, and see if these correspond to the student profiles generated in Step (I). The calibration can be done by many different optimization algorithms, but in particular, we believe Genetic Algorithms (GA) are appropriate, with as fitness criterion a distance measure between the generated student profiles, and the input student profiles. We start by generating different class, teacher and material profiles and generate a population of rule sets. The rules are evaluated through simulation, and then a subsequent population of rules is generated. The best rule set is selected. This rule set can then be tested with *real* data: compute the engagement, orderliness and activity level of a real classroom based on the logs from a session of use of the tablet-based education platform. Then use the rule set to generate an artificial log (which could also be compared to the *real* log for additional validation), and based on the artificial log, compute the different profiles again. We can then see how far they differ.

4.1 Implementation

In order to test this idea, we have implemented the simulation above. In order to test our initial assumptions, we generate a random set of student agents, and a single teacher agent. The students are generated from the class profile in Table 1.

Table 1. Class profile used to generate student agents

#students	Activity	Orderliness	eye tracking	at tablet
5	$80(\sigma = 5)$	$80(\sigma = 5)$	$80(\sigma = 5)$	$50(\sigma = 5)$
10	$30(\sigma = 10)$	$40(\sigma = 10)$	$40(\sigma = 10)$	$90(\sigma = 5)$

In this table, we see that there are two different *profiles* for students in the class, with the first profile used to generate 5 students, and the second to generate 10. Each student agent’s behaviour is governed by its profile, generated at random from the Gaussian distributions with the parameters as in the table. For instance, this can result in a student with profile 1 having an activity level of 85.25, an orderliness of 83.5, “eye tracking” parameter of 82.0, and “at tablet” parameter of 49.25. Each of these values governs different aspects of the agent’s behavior. A high activity level means that the agent is more likely to click, change pages, or perform other activities, at each time stamp. An agent with high orderliness is likely to work through the material sequentially, whereas a low orderliness indicates a propensity to jump around in the material haphazardly. The “eye tracking” and “at tablet” parameters together govern what the student

looks at. “Eye tracking” gives the percentage of time that a student spends looking at either the tablet, or the teacher (as opposed to elsewhere), while “at tablet” is used to divide that time between the tablet and the teacher.

The teacher agent has a similar profile, but there is no need for eye tracking, so it only consists of activity level and orderliness. Finally, we generate a material profile, which specifies a set of learning objects, and their order and organization by chapters. Each learning object is modeled with a type of object and its difficulty level.

Independently of the class, teacher and material profiles, the main engine of the simulator is the rule-based system for each agent. All student agents use the same rules (or plans), and the teacher agent has a separate set. Because we do not know how to model such rules, we calibrate the rules with a genetic algorithm (GA). As stated in the previous section, the rules are used to define each agent’s behaviour, given the state of the world: what learning object the agent is at (and for how long), what learning object the teacher is at (and for how long), the characteristics of both learning objects, whether it is currently looking at the tablet or the teacher, and the intrinsic parameters defined in the agent’s profile). The agent matches the plan rule that is nearest to its current state, and that rule specifies the parameters for the stochastic method of choosing the next state. In our GA, the initial population consists of 200 models (individuals), for each of which we generate a random set of 650 rules for students and a further 150 for the teacher (or approximately 800 rules out of a possible $2exp13$). We then run each model to generate the events, which in turn are analysed and result in 15 student profiles and a teacher profile. The fitness of this profile is simply how near it is to the original profile (using Euclidean distance).

For creating the next generation we use an elitist approach: the 20% best models of each generation are copied over as is, with a 1% chance for mutation. Furthermore, these serve as the basis for crossover (random pairs are selected) which creates a further 20% of the next population. The remaining 60% is generated randomly. To perform crossover we pick two parents randomly, and for each rule, either parent has an equal chance to contribute theirs (and if one parent has more rules than the other, these are also all copied over). Our mutation operator works as follows: there is a chance a set of rules is mutated, in which case one rule (chosen randomly) is generated again randomly. For further information on genetic algorithms and these operators we refer to Mitchell’s work [23].

4.2 Experiments

We implemented the simulator and genetic algorithm (GA) as described above in Python. The rules, class profile, teacher profile and material profile are all represented in JSON format. Subsequently we ran the GA for 200 generations, but the results plateaued after 40 generations. To compare whether crossover and mutation were contributing anything over random search, we also ran the algorithm with a selection of 40%, and no crossover or mutation. We then compare the top 40% of each generation; the results are in Fig. 2.

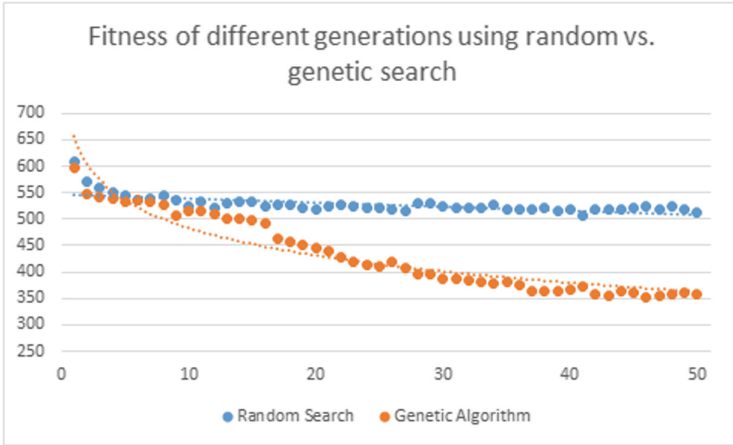


Fig. 2. For each generation, a comparison between pure random search, and the GA with crossover and mutation. The dotted lines are the trend lines (logarithmic for the GA, and linear for the random search)

We see that the GA improves the performance significantly over a standard random search, indicating that our crossover and mutation operators work. Nevertheless, the error still seems high, even in the best case. However, that is if we expect the simulation process to predict the exact same set of students (and thus have a distance of 0). What we actually aim for is to simulate at a class level, and thus what we expect is that the process results in a set of students from the same class profile. The expected distance¹ between two sets of agents generated using the profiles of the table above, is 232. The best model in the last generation from the GA has a score of 272, or in percentages, it is 17% away from this optimal score. Whereas the best result from the random search has a score of 385, or 66% away from the optimal score. This indicates that our GA is on the right track, and we can probably improve it, by fine tuning the parameters.

In order to validate the result, we also have to test how well the simulation works on real data. For this, we test the 20 best results of our last generation on real data collected in field trials of the Digital Teaching Platform [18, 19]. However, the real data is logs of actions in the tablet-based system, and not the resulting profiles. As such, we should compare the simulated logs with the real one. In order to do this, we extract the class, material and teacher profiles from the real log data, and use this profile in each of the 20 models. This generates a log

¹ For each student, the expected distance is the expected distance between two samples from the same normal distribution (as defined by the class profile), which is: $\sqrt{\sum_{p \in \text{parameters}} 4\sigma_p^2/\pi}$, with σ_p^2 the variance of the distribution for parameter p . Summing this distance over all students gives us the expected distance between two sets of student agents generated from the same class profile.

of actions, which we can compare to the original log. However, this comparison is not trivial. Our first impression is that the simulated log and the real log are very different. For starters, the simulated log contains over 100,000 actions, whereas the real log contains only 8,000. This also results in the Levenshtein (edit) distance (the minimal number of insertions, deletions and substitutions to make two strings equal) [20] between the two logs being approximately the same as the length of the simulated log, and thus not a useful measure. Instead, we look at the frequency of event types, and the χ^2 distance measure [6] between them. This results in an average distance of 0.27, which is a more useful measure: it shows that while the number of events are different, the general frequency of events is fairly similar. As a measure of similarity between logs, we therefore suggest to use the geometric mean of the edit distance between the actual logs and the χ^2 difference between their histograms.

Future work is to adapt the simulator using a two-step GA to evolve a model that is both calibrated (in the sense outlined in the first experiment) and valid (using the distance measure we just discussed).

5 Discussion

Agent-Based Modelling and Simulation is a powerful tool for doing research into complex systems, such as education, and while it has so far not been used in this field, we believe the data is now available to take advantage of this methodology. Moreover, data-driven research into education is a fast-growing field of research, and ABMS can take advantage of increasingly sophisticated quantitative metrics of education to incorporate into the agent models. Once agent-based models are sufficiently robust, they can even help validate these metrics, by quantifying how much explanatory power they add in the model, in comparison to the added complexity. This can be an invariable tool in quantitative research into education, by helping to hone in on what metrics best explain and predict student learning, and present such information to the students, teachers, parents and other stakeholders. Moreover, simulations can be designed to test what type of interventions work best.

Our own agent-based model, although still in an early stage, is intended in such a capacity. The model as described thus far is a first step, and we have specific extensions planned to better take advantage of the unique possibilities of agent-based approaches: the rules that are created to differentiate between agents' behaviour do not take social interactions into account, and the student profile does not take motivational factors or learning styles into account. Moreover, we are still working on creating a valid model based on the data we gather, but the potential is clear: we can take real class profiles and test how they perform using a different content profile. If the students are more engaged than their input profile suggested, this is an indication that perhaps that type of content works better with that class. We can design experiments to test such hypotheses and use a methodology like *design-based research* [2] to iteratively improve the model and test its predictions.

Acknowledgements. This research was implemented with support from Samsung Research Institute Brazil, project “Integrated Ecosystem of Digital Education”. Fernando Koch, Andrew Koster and Nicolas Assumpção were working with Samsung Research Institute Brazil during the time of the implementation of this project. Fernando Koch received a CNPq Productivity in Technology and Innovation Award, process CNPq 307275/2015-9. Andrew Koster is a Marie Skłodowska-Curie COFUND Fellow in the P-Sphere project, funded by the European Unions Horizon 2020 research and innovation programme under grant agreement No. 665919. Andrew Koster also thanks the Generalitat de Catalunya (Grant: 2014 SGR 118).

References

1. Anderson, B.: Computational neuroscience and cognitive modelling: a student’s introduction to methods and procedures. Sage (2014)
2. Bakker, A., van Eerde, D.: An introduction to design-based research with an example from statistics education. In: Approaches to Qualitative Research in Mathematics Education, pp. 429–466. Springer, Dordrecht (2015)
3. Bazzan, A.L., Kliegl, F.: A review on agent-based technology for traffic and transportation. *Knowl. Eng. Rev.* **29**(03), 375–403 (2014)
4. Bonabeau, E.: Agent-based modeling: methods and techniques for simulating human systems. *Proc. Nat. Acad. Sci.* **99**(suppl 3), 7280–7287 (2002)
5. Burns, A., Knox, J.: Classrooms as complex adaptive systems: a relational model. *TESL-EJ* **15**(1), 1–25 (2011)
6. Chardy, P., Glemarec, M., Laurec, A.: Application of inertia methods of benthic marine ecology: practical implications of the basic options. *Estuar. Coast. Mar. Sci.* **4**, 179–205 (1976)
7. Chrysafiadi, K., Virvou, M.: Student modeling approaches: a literature review for the last decade. *Expert Syst. Appl.* **40**(11), 4715–4729 (2013)
8. Eberle, J., Lund, K., Tchounikine, P., Fischer, F.: Grand challenge problems in technology-enhanced learning II: MOOCs and beyond. In: Perspectives for Research, Practice, and Policy Making Developed at the Alpine Rendez-Vous in Villard-de-Lans. Springer (2016)
9. Epstein, J.M.: Generative Social Science: Studies in Agent-Based Computational Modeling. Princeton University Press, Princeton (2006)
10. Gardner, M.: Mathematical games: the fantastic combinations of john conway’s new solitaire game “life”. *Sci. Am.* **223**, 120–123 (1970)
11. Gašević, D., Dawson, S., Siemens, G.: Lets not forget: learning analytics are about learning. *TechTrends* **59**(1), 64–71 (2015)
12. Greer, J.E., McCalla, G.I.: Student Modelling: The Key to Individualized Knowledge-Based Instruction. Computer and Systems Sciences, vol. 125. Springer, Heidelberg (1994)
13. Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S.K., Huse, G., et al.: A standard protocol for describing individual-based and agent-based models. *Ecol. Model.* **198**(1), 115–126 (2006)
14. Herd, B., Miles, S., McBurney, P., Luck, M.: MC²MABS: a monte carlo model checker for multiagent-based simulations. In: Gaudou, B., Sichman, S.J. (eds.) MABS 2015. LNCS, vol. 9568, pp. 37–54. Springer, Cham (2016)

15. Jordan, M., Schallert, D.L., Cheng, A., Park, Y., Lee, H., Chen, Y., Chang, Y.: Seeking self-organization in classroom computer-mediated discussion through a complex adaptive systems lens. In: Yearbook of the National Reading Conference, vol. 56, pp. 39–53 (2007)
16. Keshavarz, N., Nutbeam, D., Rowling, L., Khavarpour, F.: Schools as social complex adaptive systems: a new way to understand the challenges of introducing the health promoting schools concept. *Soc. Sci. Med.* **70**(10), 1467–1474 (2010)
17. Klügl, F., Bazzan, A.L.: Agent-based modeling and simulation. *AI Mag.* **33**(3), 29 (2012)
18. Koster, A., Primo, T., Koch, F., Oliveira, A., Chung, H.: Towards an educator-centred digital teaching platform: the ground conditions for a data-driven approach. In: Proceedings of the 15th IEEE Conference on Advanced Learning Technologies (ICALT), Hualien, Taiwan, pp. 74–75. IEEE (2015)
19. Koster, A., Zilse, R., Primo, T., Oliveira, Á., Souza, M., Azevedo, D., Maciel, F., Koch, F.: Towards a digital teaching platform in Brazil: findings from UX experiments. In: Zaphiris, P., Ioannou, A. (eds.) LCT 2016. LNCS, vol. 9753, pp. 685–694. Springer, Cham (2016). doi:[10.1007/978-3-319-39483-1_62](https://doi.org/10.1007/978-3-319-39483-1_62)
20. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl.* **10**(8), 707–710 (1966). Original in Russian in *Dokl. Akad. Nauk SSSR* 163, 4, 845–848 (1965)
21. López Bedoya, K.L., Duque Méndez, N.D., Brochero Bueno, D.: Replanificación de actividades en cursos virtuales personalizados con árboles de decisión, lógica difusa y colonias de hormigas. *Avances en Sistemas e Informática* **8**(1), 71–84 (2011)
22. Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., Cohen, W.W., Stylianides, G.J., Koedinger, K.R.: Cognitive anatomy of tutor learning: lessons learned with simstudent. *J. Educ. Psychol.* **105**(4), 1152–1163 (2013)
23. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT press, Cambridge (1998)
24. Niazi, M.A., Hussain, A., Kolberg, M.: Verification and validation of agent based simulations using the VOMAS (virtual overlay multi-agent system) approach. In: Proceedings of the Second Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW), Turin, Italy, CEUR-WS (2009)
25. Papamitsiou, Z., Economides, A.A.: Learning analytics and educational data mining in practice: a systematic literature review of empirical evidence. *J. Educ. Technol. Soc.* **17**(4), 49–64 (2014)
26. Peña-Ayala, A.: Educational data mining: a survey and a data mining-based analysis of recent works. *Expert Syst. Appl.* **41**(4), 1432–1462 (2014)
27. Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **40**(6), 601–618 (2010)
28. Siemens, G., Baker, R.S.J.: Learning analytics and educational data mining: towards communication and collaboration. In: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, pp. 252–254. ACM (2012)
29. Siemens, G., Long, P.: Penetrating the fog: analytics in learning and education. *EDUCAUSE Rev.* **46**(5), 30 (2011)
30. Tesfatsion, L., Judd, K.L.: *Handbook of computational economics: agent-based computational economics*, vol. 2. Elsevier (2006)