Agent-Based Modeling and Simulation

Franziska Klügl, Ana L. C. Bazzan

■ *This article gives an introduction to agent-based modeling and simulation (ABMS). After a general discussion about modeling and simulation, we address the basic concept of ABMS, focusing on its generative and bottom-up nature, its advantages as well as its pitfalls. The subsequent part of the article deals with application-oriented aspects, including selected tools and well-known applications. In order to illustrate the benefits of using ABMS, we focus on several aspects of a well-known area related to simulation of complex systems, namely traffic. At the end, a brief look into future challenges is given.*

How is it possible that a whole ancient civilization disappeared? Was this caused by climate changes? What type of recruitment strategy among social insects is best adapted to their particular environment? How much time does it take to evacuate an airport if people have limited perception caused by smoke as well as restricted mobility? What if many of these people travel in groups or families? How long does it take for commuters to reach their destinations if an important arterial in the Los Angeles area is closed? These are the kind of questions that can be and are answered by agent-based modeling and simulation (ABMS).

In this paradigm, simulated human beings or animals are modeled as agents, interacting with some of their peers as well as with their environment. The environment, as in many multiagent systems, plays a key role and must therefore be carefully taken into account. For instance, passengers seeking to leave the airport just mentioned try to find the shortest way to an exit, which may be partially hindered by debris. These are only some examples of scenarios — also characterized as complex adaptive systems — that can be investigated using ABMS. The core idea here is to use simulated agents for producing a phenomenon that shall be analyzed, reproduced, or predicted. This generative, bottom-up nature of modeling and simulation provides great potential for dealing with problems in which conventional modeling and simulation paradigms have difficulties capturing the core features of the original system.

In what follows, this particular modeling and simulation paradigm, its concept, properties, and application are introduced and discussed. To this end, concepts about modeling and simulation in general, and about ABMS in particular, are introduced and discussed in the next two sections. Then, some popular environments for ABMS are briefly presented. Applications and case studies are then discussed. We remark that, due to lack of space, we have opted to focus on two particular domains: social science simulation (one of the earliest application domains) and traffic simulation (given the increasing interest in transportation- and traffic-related applications). Readers interested in a broader view on applications may refer to Phan and Amblard (2007) or Uhrmacher and Weyns (2009). We then conclude the article with a discussion on future challenges.

Modeling and Simulation in General

For many decades, development, analysis of, and experimentation with models have been a part of the instruments of basically all domains of science and engineering. Modeling is the development of a model as a representative of a system. Simulation can be defined as experimenting or executing a model. There are numerous textbooks on modeling and simulation, such as Law (2007) and Gilbert and Troitzsch (1999), that provide good introductions to modeling and simulation in general or with a focus on particular areas.

Since the idea behind modeling and simulation is to use a model instead of a real system, the best possible correspondence between the former and the latter is essential. Therefore validity is a key issue. An acceptable degree of validity of course clearly depends on the objective of the model and the simulation. Possible objectives may be increasing the understanding of the original system, optimizing it, or predicting the reaction of the system to particular measures.

There are several different approaches for modeling that use different representation formalisms and simulation methods. The best choice of a modeling paradigm depends on properties of the system under investigation and on the goals of the simulation study. Different paradigms can be characterized by the underlying time representation (continuous versus discrete) or the granularity of the model elements (macroscopic, microscopic).

Agent-Based Modeling and Simulation: Concepts and Features

Agent-based modeling and simulation — sometimes also called multiagent simulation or multiagent-based simulation — applies the concept of multiagent systems to the basic structure of simulation models. One may also find the term *agent-directed simulation* used as a more general notion (Ören et al. 2000).

In ABMS, active components or decision makers are conceptualized as agents, being modeled and implemented using agent-related concepts and technologies. Thus, one can define agent-based modeling (ABM) as a representation of an original or reference system that is conceptualized as a multiagent system. In the present text we use the term *ABMS* to refer to the general modeling and simulation paradigm, reserving *ABM* to the particular task of modeling and *agent-based simulation* (*ABS*) for the execution of a model.

The core idea of ABMS is that, instead of merely describing the overall, global phenomenon, this phenomenon can be rather generated from the actions and interactions of the multiagent system. This bottom-up nature is the most important feature of ABMS (Epstein 2007). Thus, ABMS is particularly suitable for the analysis of complex adaptive systems and emergent phenomena in social sciences, traffic, biology, and others. Such emergent phenomena are "unforeseen" patterns or global behaviors that are not derivable from properties of its constituents. Thus, an emergent structure or behavior is generated by locally interacting entities, despite the fact that it is only observable on a global, macroscopic scale, thus being not directly deducible from local behaviors. In ABM, these interacting entities can be naturally associated with agents. Generating the phenomenon from low-level actions and interactions is especially valuable because it helps understand the causes and circumstances of their occurrence.

To create an agent-based model, the following three elements have to be explicitly dealt with. First, the set of agents is the most characteristic element. These agents are autonomous with respect to the other entities within the simulated environment. Next comes the specification of the interactions of the agents among themselves and with their shared environment. Since these interactions are responsible to produce the overall outcome, the design of all involved aspects is central. Interactions need not be explicitly represented within, for example, organizational structures. Rather, they may occur implicitly, as is the case with stigmergic interactions. However, in the implemented ABS, although organizational structures may not be obvious as such, it is important that they be explicitly considered. The third element, the simulated environment, contains all other elements. These may be resources, other objects without active behavior, as well as global properties.

Whereas the previous three elements are part of ABM, for actual execution, it is necessary to have a simulation infrastructure. We remark here that the latter is sometimes misleadingly referred as a simulation environment. In principle, the infrastructure should not influence the outcome of the ABS in the same way that the use of a particular programming language should not influence the result of a particular calculation. Yet in practice technical aspects such as the particular order in which the set of agents is updated in a simulation platform are central. A more detailed discussion of the differences between the simulated environment and the simulation environment is given in Klügl, Fehler, and Herrler (2005).

For the sake of illustration of how the three elements are related, consider a simple ABS with predators (wolves), prey (sheep), and some resource that serve as food for the prey (for example, grass). The set of agents in this ABM are the wolves and the sheep. The behavior of the agents consists of performing a random walk, while interactions occur when two agents are close enough to each other. Those interactions are implemented as parts of the agent behavior. For example, if a wolf encounters a sheep, it eats the sheep and increases its energy level. The possible interactions between agent types are given in table 1.

	wolf	sheep	grass
wolf	reproduce	feed-on	–
sheep	being-eaten	reproduce	feed-on
grass	–	–	–

Table 1. Interactions in a Prey-Predator Model.

The simulated environment would consist of the spatial representation where grass objects are scattered. Note that grass is not affected by the interaction with wolves. It would also contain some global variables associated with temperature and humidity, which change according to some function and influence the availability of grass. Hence, its appearance and disappearance is an environmental process. The simulation infrastructure could be any of those that appear later in this text (see the section on simulation tools). The infrastructure is responsible for the time advance, for visualization, as well as for generating and exporting data such as the number of wolves over time.

Advantages and Relation to Conventional Simulation Techniques

ABMS opens up many opportunities and has advantages with respect to conventional modeling and simulation paradigms. Due to the explanatory power that arises from its generative nature, it allows observation and analysis of model dynamics on at least two levels: the local agent and the macroscopic level, the latter being generated from the actions and interactions on the former. To this aim, a modeler can use arbitrarily complex agent designs, meaning that there is no restriction on the complexity of the agent reasoning, on the sophistication of its internal structure, or on its interaction abilities. This freedom of design also includes heterogeneity of the agent population or of the environment. Also, multiagent learning methods, explicit optimization, reorganization, and evolutionary processes may be integrated. Hence, ABMS also offers new opportunities in cases that have so far been successfully tackled by conventional methods. A good example is how to model human operators and their influence in simulation of manufacturing processes. In conventional models, probability distributions for delays and errors are the established means to model that influence. Using ABMS, one may include simulated humans as agents into the model. These agents cause not only random delays but also may use "intelligent" strategies to cope with unforeseen situations. This last example also illustrates that ABMS can be seen as an intuitive paradigm since an actor in the original system may be directly modeled as a simulated agent. The representation gap between the original system and the model is smaller than in conventional modeling, as one does not need to transform entities in probability distributions or aggregate them in variables. This also affects the visualization of the model: the entities that a human observer is familiar with in the real world are explicitly captured and visualized in the model.

Nowadays, in most application domains that use ABMS, modeling and simulation in general has been acknowledged before as a useful tool. In these domains, successful macro- and microscopic simulations were developed using partial differential equations, cellular automata, queuing networks, Petri nets, object-oriented simulations, econometric models, and others.

Differences between ABMS and macroscopic approaches are quite obvious. The basic idea behind the latter is that the complete system is taken as one object whose state is represented by state variables, which are updated with time. The macroscopic approach has many advantages. First, once the set of mathematical formulas describing the system is known, the complete model is fully determined in a clear language. Second, the result can be reproduced easily, once the integration algorithm and its parameters are known. However, the mathematical apparatus underlying macroscopic models is only accessible to particularly trained people. Besides, some underlying assumptions made are spatial homogeneity and homogeneity among the individuals in the population, and that it is not necessary to formulate a conditional behavior or other singularities. These assumptions are acceptable if the system consists of a large number of individuals so that differences are averaged out or an explicit treatment of heterogeneity does not lead to further gains. These assumptions may lead to oversimplifications resulting in irrelevant simulation output. Parunak, Savit, and Riolo (1998) and Bagni, Berchi, and Cariallo (2002) give detailed comparison between macro- or equation-based simulations and ABMS.

In addition to comparing agent-based to macroscopic approaches, another question that arises is what distinguishes ABMS from other conventional microscopic approaches to simulation, given that these two are quite related. ABMS also has advantages over the several existing conventional simulation approaches for microscopic models. Klügl et al. (2004), focusing on a comparison between queuing networks, Petri nets, and ABS, concluded that the core advantage of ABMS in comparison to these well-known and precisely defined frameworks lies in the capability of formulating truly flexible actor behavior. The main difference between agent-based approaches and cellular automata is grounded on the idea that in cellular automata, the dynamics are spatially bound, uniform, and based on a fixed neighborhood of each cell. On the other hand, in ABMS, although an agent is also an entity that is situated in some spatial environment, its connections to the neighbors are not necessarily hard wired. Moreover, althought a cell is fixed, the agent may move.

Finally, since an implemented model may be seen as software with a specific set of requirements, the differences between ABMS and standard agent-oriented programming need to be clarified. In simulation, there is no system to design anew, but a given reference system whose behavior and structure should be analyzed or predicted. Although developing a simulation might involve similar forms of activities when compared to developing and implementing software, the basic goals of construction versus reproduction are different. Simulation analysis has to do with abstraction of all relevant elements of the overall system. Whereas in standard agent-oriented programming the environment sets the constraints for the agents, in simulation the environment is a major part of the model, containing an explicit representation of space. Also, the treatment of the virtual or simulated time is mostly not related to real time.

Potential Pitfalls of ABMS

The previously mentioned advantages come at a cost. The high degree of freedom in the design phase may pose serious challenges to less experienced modelers. A critical issue is often that the necessary level of detail is unclear. Which elements have to be part of the model, which modules can be left out or abstracted using, for example, a probability distribution? The fewer details a model has, the fewer assumptions have to be justified. Also, fewer parameters have to be calibrated, and the implementation and handling of the model become easier. However, if the model is too simplistic, the significance of the results might be compromised (Edmonds and Moss 2004).

Due to the generative nature of ABMS, the development process seems to be more often based on exploring alternative designs than a conventional simulation. There is currently a lot of research on transferring agent-oriented software engineering approaches to developing ABSs (see, for example, Gomez-Sanz, Fernandez, and Arroyo [2010] or Klügl and Bernon [2011]). General modeling processes that are adapted to ABMS to some extents have been suggested in Drogoul, Vanbergue, and Meurisse (2002), in Gilbert (2007), and in North and Macal (2007). The only suggested ABS methodology can be found in Kubera, Mathieu, and Picault (2011). They focus especially on reactive agents and start from a detailed analysis of their interactions. There are particular inherent problems in the development of an ABS that can only be handled partially by the currently existing methods. One example that makes tuning and calibration of ABSs difficult is brittleness of the model outcome. Changes in one parameter value may affect the full population of agents leading to a completely different overall result. Izquierdo and Polhill (2006) characterized such critical parameters as "knife edge parameters." Examples are sharp thresholds responsible for a change in the agent behavior, such as an energy threshold when it comes to triggering the reproduction of a predator agent.

Implementation of a model is still a problem despite the number of tools for ABS (see the next section). Sometimes the model is simply too large. Size here relates either to the complexity of the behavioral models of the agents, or to the number of agents, or to the size of the environment. Whereas the last two go back to implementation issues that can be addressed by using high-performance and distributed computing, the first may pose conceptual problems related to justifying assumptions, clean handling of parameters, and others.

Reproducibility of results was identified as a major problem (Axtell et al. 1996), mostly arising from incomplete documentation of the agent-based models, even if reproduction of results is the basic ingredient in all scientific works. Wilensky and Rand (2007) advise how to prepare a model so that it supports reproduction. In fact, weak and insufficient documentation has led to the introduction of the ODD Protocol (Overview, Design concepts, Details) for model documentation

(Grimm et al. 2006). Due to the variety of agent-based models, an approach such as ODD, which is more a guideline on the necessary types of information rather than on languages to use, is an appropriate basis for model documentation.

Finally, validation poses severe problems for ABMS from both a scientific and practical point of view. Only in rare cases and domains are there enough empirical data available for a full validation. More often, tackling a high level of detail poses problems when it comes to proving that the model captures the most important properties of the original system. This is even more serious regarding models that involve humans, their reasoning, and decision making. Without reliable data, the modeler must go back to simpler plausibility checking. As validation is such a central issue, it has been discussed in a number of contributions. Klügl (2008) suggests different phases involving at least two levels: the agent level and the macrolevel. Windrum, Fagiolo, and Moneta (2007) analyze general methods for empirical validation and their application to ABMS, whereas Barreteau and others (2003) count on the involvement and repeated reviewing by system experts and stakeholders. Also, participatory ABMS — see for example (Guyot and Honiden 2006) — can support testing and validation of an ABS. Yet, it can also be used for model elicitation, that is, for supporting the identification of actors, their goals, and behaviors.

When to Use ABMS

Considering the advantages and pitfalls, one can conclude that ABMS is particularly appropriate for systems that present the following characteristics, partially inspired by the requirements listed in Hare and Deadman (2004):

Systems that draw their dynamics from flexible and local interaction. Variable population sizes, structures, and interactions can be difficult to consider in other simulation paradigms.

Systems that require the representation of heterogeneity regarding not only states but also behavioral rules. This may be hard to model in paradigms that assume homogeneity.

Multilevel systems that require observation on several levels, especially when there is no connection between them. This is the case with emergent phenomena.

Systems where decision making happens on different levels of aggregation. Microlevel decision making concerns the behavior of the individuals, whereas higher-level decision making is done by some regulatory authority or entity. Feedback loops affect individuals as well as aggregate levels.

Systems that include learning or evolutionary processes at individual and at population level.

Systems that incorporate intelligent human behavior such as sociotechnical systems, where flexible team or group tasks have to be modeled.

Systems in which the assumptions necessary for an equilibrium-based modeling are too strong. Such assumptions may refer to homogeneity of space, uniform decision making, perfect information, rationality, and others. According to Epstein (2007), agent-based simulations "... may have the effect of decoupling individual rationality from macroscopic equilibrium."

Systems in which the focus is not on a stationary equilibrium but rather on the phenomena and behaviors that lead to it. Thus the transient dynamics must be analyzed.

Tools for ABS

In this section some established tools for ABS are introduced. Note that although they all do well when it comes to ABS, their support during the modeling phase varies greatly. Despite the existence of several tools, the focus here is on general-purpose and freely available ones. A more extensive treatment of some tools mentioned below can be found in Railsback, Lytinen, and Jackson (2006). Also in JASSS[1] from time to time surveys of current ABS tools are published.

Swarm

Swarm[2] is one of the earliest tools for implementation of ABSs and complex systems. It is in fact composed of libraries that provide the core from where developers can build their ABSs, as well as perform collection and analysis of data, display, and control parameters of the model. The original libraries were based on Objective-C but currently Java can also be used. Although libraries are provided, a user without programming skills may have to spend some time with the coding. There is no explicit default representation of the environment. In Swarm, there is the possibility of an agent being a swarm itself, in which case the behavior of this agent emerges from the behavior of the agents inside it. This way, hierarchical models can be built by grouping swarms.

Recursive Porous Agent Simulation Toolkit

The Recursive Porous Agent Simulation Toolkit (Repast)[3] is also a platform based on Java. In the same spirit as Swarm, Repast provides a library of classes for the most common tasks associated with the implementation of an ABS. Besides, since the initial focus of Repast was social science, it includes some tools that are useful in this domain such as network analysis. Recently, the Repast Symphony was introduced, which is a visual modeling tool based on state charts.

Shell for Simulated Agent Systems

The Shell for Simulated Agent Systems (SeSAm)[4] provides a fully visual interface for the development of ABMS. Contrarily to the requirements Swarm puts on its users, the user of SeSAm does not need to know any programming language. The code (even regarding data presentation, definition of plots, and other kinds of analyses) is assembled together by means of a graphical interface. The kernel of a SeSAm simulation is the system's model itself, which is built using activity diagrams. Predefined primitives for agent actions, perceptions, and their processing can be enhanced by user-defined functions. A special type of agent is the "World" that determines how the environment behaves and may manage different kinds of spatial representations.

MASON

MASON,[5] a library based on Java, aims at facilitating the programming of large-scale simulations. The reasoning underlying MASON is to be compact in order to gain in performance. Hence, it is especially attractive for performance-demanding applications, but it requires programming skills. MASON supports serialization and not only two but also three-dimensional visualization, separated from the simulation kernel.

NetLogo

NetLogo[6] was designed with the end user in mind. It has basically three interfaces. The first is a kind of editor for programming the model itself, where the language resembles Starlogo. NetLogo's second interface permits the visualization of the environment and its parameters and also allows the user to play with the model parameters by means of sliders. The third interface contains a structured documentation. NetLogo is turning increasingly popular due to its extensive documentation, the existence of good tutorials, and a large library of preexisting models.

Other Tools

Many other tools are commonly listed in repositories related to ABMS.[7] However many of them were specifically designed for particular purposes. For illustration we arbitrarily selected the following two. MadKit for instance builds upon an organizational model of agents' societies. Therefore it is useful in domains where one aims at simulating intra- and interorganizational processes. However, if the problem at hand is not necessarily focused on such an organizational model, it may not be the best tool for the problem. Similarly, CORMAS is a programming environment that targets natural resources management. During the last years, more and more agent platforms were used for implementing simulation applications. Although they support the agent-based concepts, they miss the infrastructure that is specific for simulations, such as integration of input data, handling virtual time, model instrumentation, data collection, and others. Depending on the particular focus and objective of the simulation (for example, support belief desire intention [BDI] agent architecture), it might indeed be a good idea to use standard agent-programming tools.

Comparison of the Tools

The previously mentioned general-purpose tools can be compared regarding the following dimensions: programming language, underlying simulation model, interface, predefined primitives, and type of agents.

Regarding the programming language for actually implementing the models, as mentioned, most of the tools assume Java knowledge. Two exceptions are SeSAm and NetLogo. Regarding the underlying simulation elements (agents, interactions, environment), in all frameworks, these have to be separately modeled and brought together in the actual simulation, following the bottom-up generative approach of ABMS. In Swarm, the agents that integrate the model have to be defined before the environment. In SeSAm, the focus is both on the environment and on the agents (defining attributes, goals, activities, and others.).

All frameworks provide predefined primitives for ABS, but ABM is facilitated to different extents depending on the tool. A graphical user interface helps the user to build the model (but in some cases it has to be specified elsewhere). When such an interface is not provided, the programming load increases, eventually preventing the use by nonexperts. Such a graphical interface for modeling is well explored in SeSAm. In NetLogo, the graphical user interface is present but it serves basically as an editor where the code must be written.

Regarding agents' types of behaviors, all tackle reactive agents and none directly supports planning or other cognitive tasks. This is accomplished only by specific toolkits and not by the general frameworks mentioned. However, especially for social simulation of anthropological systems, features to model mental states in agents are highly desirable.

Applications

Nowadays, ABMS concepts are applied in basically all domains. ABMS can be seen as a possible "killer application" of agent technology. Because of the wide range of application areas (from archeology to zoology), it is impossible to give a comprehensive overview. Therefore, we next focus on two areas: social science (as the first application domain), and traffic simulation (as one that is receiving a lot of attention).

Applications in Social Sciences

Schelling (1969) was basically the first to consider individuals in a simulation. In his scenario dealing with residential segregation, he has proposed a neighborhood modeled by means of squares in a grid, where in each there can be an individual (household). For each household, the eight neighbors are observed regarding how many share a given household characteristic (in Schelling's case, skin color). If this number is below a given threshold, then this household is transferred to a randomly selected and unoccupied position in the grid. This process is repeated until all households no longer move.

This kind of approach seems quite obvious to us now, 40 years later. However, it represents the breaking of a paradigm and has opened up the way for ABMS. For instance, using the then prevailing paradigm of differential equations, Schelling would have been able to model the time component, but not the spatial interactions.

Sugarscape

Sugarscape (Epstein and Axtell 1996) is a milestone in ABMS due to the fact that with a simple model the authors were able to show the emergence of various social phenomena. It is an artificial system consisting of a landscape (the sugarscape proper) and agents. The landscape consists of regions with different quantities of sugar. Agents have a vision (the distance they can see when foraging for sugar), a metabolism (consuming sugar), and other attributes coded in their artificial genetic code. These agents have local rules to decide about their movement in the landscape, about trading behavior, combat, and other kinds of interaction with the environment. By moving in the landscape, agents harvest sugar and burn it at a rate that is given by their own metabolism. If all sugar in the individual storage is burned, then the agent simply dies. Sugarscape agents also engage in sexual reproduction, transmitting genes that are responsible for, for example, vision as well as parts of the storage.

With this basic model the authors were able to generate, for example, a skewed distribution of wealth. Other phenomena reported later as extensions were introduced, such as spatial segregation (for example, two tribes), a second resource (spice), and environmental factors such as pollution caused by some economical activity. Regarding the former, because of population growth in one or both tribes, agents are forced to forage further away thus initiating some interaction between tribes. The introduction of spice allows trade among the agents (metaphor for an economic market), while pollution yields changes in the environment, affecting foraging and others.

The Artificial Anasazi Project models the Kayenta Anasazi of Long House Valley (Arizona, USA) over the period 800 to 1300 A.D., at which point the Anasazi mysteriously disappeared. One issue of the Anasazi enigma is whether environmental factors alone can account for their disappearance. According to Epstein, "in bringing agents to bear on this controversy, we have the benefits of (a) a very accurate reconstruction of the physical environment (hydrology, aggradation, maize potential, and drought severity) on a square hectare basis for each year of the study period, and (b) an excellent reconstruction of household numbers and locations." Due to a collaboration with anthropologists, plausible rules of agent behavior were identified (see Dean et al. [2000)] for a report of phase I of the project, which includes those rules).

One result of the project has been the demonstration that the environmental rules accounted for important features of the Anasazi's demography (for example, decline in population around 1300). However, they did not generate the outright disappearance that occurred. Authors' interpretation is that subsistence considerations alone do not fully explain the enigma. Despite this, the authors make the point that ABMS permits a new kind of empirical research, while also allowing a novel kind of interdisciplinary collaboration.

Other Case Studies in Social Sciences

Relevant studies are related to anthropology, political science, and economics. Epstein (2007) presents a list that includes, for example, the reproduction of the alignment of 17 nations regarding alliances during the Second World War, and the generation of the relevant statistical distribution of prices in an agent-based trading model. Regarding economical sciences in particular, ABMS differs, for example, from experimental economics whose aim is to understand why specific rules are applied by humans, and from models based on dynamic stochastic general equilibrium. The latter are often criticized. For instance, a recent article in *The Economist* magazine[8] mentions that conventional models "perform well enough in a business-as-usual economy. They do badly in a crisis however, …, as there is no equilibrium during crashes." This article raises the question about using a single, conventional model based on rational expectations and top-down design to model increasingly complex markets, given that there are alternatives. Although the article emphasizes that there is less agreement on what should replace those models, it mentions ABMS as one promising alternative. This issue is more extensively discussed in the framework that concerns agent-based computational economics; see for instance Tesfatsion and Judd (2006).

Simulating Games

Metaphors stemming from game theory can be used to investigate social interactions. In minority games for instance one is interested in studying situations that involve coordination of many agents in social systems (also known as collectives), in which most of the individual decisions are not independent. In such cases, there is no a priori best strategy for a single player since the outcome of a game depends on others. This was the focus of B. Arthur (1994), who introduced a coordination game called the El Farol Bar Problem (EFBP), later generalized by Challet and Zhang (1997). The interest in simulating minority games is explained by the fact that they resemble binary decisions that happen in our daily lives, such as route choice in traffic, sell-or-buy stocks, and others. Given that these scenarios are inherently distributed, they are appealing candidates for ABMS. Another popular subject in ABMS has been the simulation of the prisoner's dilemma (for example, Epstein [2007], chapter 9).

Applications in Traffic Simulation

In traffic-related simulation, ABMS's advantage is manifold. First, it is able to capture necessary details at entity level as well as to reproduce the bottom-up way of generating phenomena as real traffic participants do. Second, it enables modeling of complex decision making considering multiple factors and dynamic information (including learning and en route adaptive behavior). Third, the behavior of individual drivers or platoons of vehicles can be visualized, monitored, and validated, thus facilitating testing and debugging. Fourth, ABS supports complex travel decisions on different levels of granularity in space and time. Finally, one of the major advantages of ABMS is its ability to handle heterogeneity or idiosyncrasies at the level of individual agents. This can be used, for example, to model individual drivers' behavior. In route choice scenarios in particular, essentially, each driver has a strategy to pick the best route. However, commuting increasingly depends on information broadcast, which can have a serious impact on stability of traffic conditions (Wahle et al. 2000). Thus ABMS is an important tool in understanding such impacts.

On the other hand, traffic simulation poses interesting challenges to ABMS, concerning modeling and design of interaction among autonomous decision-making agents, as well as between an agent and its (complex, dynamic) environment. Traffic simulation is also interesting from a methodological point of view, by dealing with the necessary level of detail and required knowledge, and by integrating the different levels of abstraction present in traffic systems.

In the remainder of this section, examples of case studies in this domain are discussed. Becausse of lack of space we restrict ourselves to simulation of demand assignment (for example, route choice) but remark that ABMS has been used in several cases related to simulation of traffic flow, control (for example, using traffic lights), management, testing new paradigms, air-traffic control, and pedestrian and crowd simulation. Practically all these works were implemented using some kind of general-purpose simulation tool such as MASON as in Balan and Luke (2006), or SeSAm as in the works by Bazzan and Klügl. Some have developed and used specific traffic simulation tools such as MATSim (Balmer et al. 2008), DRACULA (Rossetti et al. 2002), and ITSUMO (Bazzan, Oliveira, and Silva 2010) to investigate issues that are well related to multiagent system such as multiagent learning (Bazzan 2009; Desjardins, Laumonier, and Chaib-draa 2009). Finally, some use standard object-oriented languages to construct simulators to test new management and control paradigms such as the reservation-based approach by Dresner and Stone (2004) or the market-based one (Vasirani and Ossowski 2011).

Travel Demand Generation

Determining the travel demand is normally the first phase in traffic simulation. The output is the number of trips from a given origin to a given destination. Determining travel demand is traditionally a data-driven activity based on demographic data and interviews, statistics on workplaces and households, car ownership, and others. Whereas this kind of trip-based approach prevailed in the past, agent-based approaches are mostly related to activities (activity-based approach). Here, the daily schedule of a typical human that belongs to a particular behavior class is reproduced. It consists of activities that happen in particular locations, and trips that cause location changes. Origin, destination, and also departure time are determined based on the respective activity/trip. Here, the motivations for using agents are manifold: the ability to represent complex socioeconomic properties and other sources of heterogeneity; individual adaptation and learning of daily plans; integration of other levels of decision making such as mode and route choice. Therefore it is not surprising that a number of works use ABMS for travel demand studies.

The research team led by Arentze and Timmermans has developed activity-based models for demand generation grounded on existing theories in psychology and economics. In Arentze and Timmermans (2005) the use of Bayesian networks is proposed for developing a model of mental maps as an individual representation of the user's environment, which typically contains incomplete and incorrect information. Rindsfüser, Klügl, and

Freudenstein (2004) proposed a model of an intelligent agent for adapting a daily activity schedule with respect to external events. Starting from the definition of a habitual daily program as a coarse pattern, this pattern is extended and adapted on demand, as a reaction to the traffic situation and nonhabitual activities. While this tool works as a first prototype, MATSim (Balmer et al. 2008),[9] has produced positive outcomes in large-scale cases, as for instance the simulation of a full day of traffic in the complete Zürich area. In MATSim each agent possesses a set of complete daily schedules, including details of the route choice. Schedules of all agents are simulated, evaluated, and optimized by means of genetic algorithms for adapting the plans.

Traffic-Related Choice Processes

A number of publications suggest the application of ABMS to different travel-related choice processes such as route and mode choice. Agent-based approaches seem to be particularly relevant when networks are dynamic or when dynamic information is available. In the following we present a number of works that illustrate the use of ABMS especially applied to choices regarding traffic (mode, route, departure time, and others.), focusing on case studies that aim at determining the influence of providing information to drivers (by broadcast, embedded devices, or Internet) on their behaviors. These fit mainly two categories: the scenario is abstracted using metaphors from game theory, or the choices occur in more fine-grained scenarios. We start with the former.

Abstract scenarios are mostly inspired by congestion or minority games. The basic idea is that agents have to decide simultaneously between two or more routes; those that select the less crowded one receive a higher reward. Agents' repeated decision making is coupled to some adaptation or learning strategy so that the next choice is adapted to the reward feedback. Based on this, the user equilibrium may be reached, which means that no agent can improve its reward by switching routes without worsening any other agent. Examples of such abstract two-route scenarios can be found in Klügl and Bazzan (2004) and in Chmura and Pitz (2007). In both cases, reinforcement learning is used to let agents learn which route to take. The reward is aligned with the global goal (balanced use of both roads) so that the theoretical equilibrium is reached. However, and more significantly, both were validated against data coming from laboratory experiments conducted to analyze the behavior and the learning schemes used by subjects in a commuting scenario, and both were able to reproduce the general pattern of route choices made by those subjects. Going beyond the laboratory experiments, Klügl and Bazzan (2004) have tested the effects of adding a second phase in decision making by the agents. In this, the initial decision acts as input to a computation that yields a forecast, which is then forwarded to all or part of the agents. Thus, the agents not only learn about selecting a route in the first phase, but also how to evaluate the information received in the second phase. One important result of this study was that a certain share of agents that ignore traffic forecasts turned out to be necessary for an efficient agent adaptation. This has consequences, for instance, for the deployment of route-guidance devices such as those that come with navigation systems: information must be given in a very selective or personalized way. Departing from simple two-route scenarios, Bazzan and Klügl (2005) showed that providing information can be also useful in the context of the Braess paradox (adding a new link in the network may actually end up increasing the travel time for everyone).

In these game-theoretic scenarios, the reward of agents when selecting a route is calculated by an abstract function that considers the number of all agents that selected that alternative. This is a very abstract view and does not resemble the actual dynamics of traffic situations. Therefore it is only useful in coarse-grained studies such as pretrip route choice. Objectives such as finding the appropriate form of information that leads to equilibrium states cannot be accomplished with such abstract scenarios. Therefore we next discuss works that were implemented in a less abstracted level.

With the same goal as Klügl and Bazzan (2004), but departing from a two-route choice, Yamashita and Kurumatami (2009) have proposed a cooperative car navigation system with route information sharing, in which each vehicle transmits its current position, destination, and route to a route information server. This estimates future traffic congestion using the current congestion information. Estimates are then fed back to each vehicle, which uses this estimation to replan its route.

Classical simulation tools from transportation engineering have also been used, combined with techniques from AI and multiagent systems. Panwei and Dia (2006) use a fuzzy neural architecture where socioeconomic parameters are represented as fuzzy variables for simulating decision making about whether or not to keep their initial route decision when new information is available. This extends previous works by the group, which were based on a BDI agent architecture. A BDI architecture was previously used in Rossetti et al. (2002) — here implemented in Agentspeak(L) — to extend the DRACULA simulator, and to compare decision making in situations using no information, using information given only before the start of the travel, and using information given both before and during the travel.

Of course the increase in complexity regarding the architecture of the agents raises the issue that creating explicitly layered agent architectures for modeling drivers in traffic simulations is a natural extension. Therefore this also appears in Burmeister, Doormann, and Matylis (1997), one of the first agent architectures for traffic simulation; in Ehlert and Rothkrantz (2001) for individual driving styles; in Bazzan, Wahle, and Klügl (1999), which combines a BDI-based layer for route choice with a reactive layer for actual driving; in Balmer et al. (2004), which combines physical mobility and mental decision-making layers; and in DesJardins, Laumonier, and Chaib-draa (2009), combining an action layer for vehicle control actions and a coordination layer for action choice.

Challenges for the Future

As stated in the previously mentioned article from *The Economist,* there are alternatives to economic models based on dynamic stochastic general equilibrium. For instance, Farmer and Axtell have proposed the construction of an immense agent-based model of the entire global economy. Similar considerations apply for modeling global pandemics and their possible containments as suggested by Epstein (2009). This could be accomplished by one or both of the following measures. First, an enormous simulation effort could be set, fed by a massive, real-time amount of data. Second, there could be a suite of agent-based models, each projecting a possible future.

In both cases central issues that arise are: where the data would come from; how to guarantee anonymity; validation; how to handle distribution (of data and processes) and combine models. Regarding the latter, although there is a lot of work carried out in the area of general distributed simulation, the combination of different agent-based models developed by different authors as well as their simulation in a massively distributed setup has still to be solved for realistic models. Capturing complex large-scale phenomena can also be supported by new hybrid modeling approaches combining partial models on multiple levels of aggregation or detail. Different modules could use different modeling paradigms, as for instance the combination of an agent-based approach (to deal with the parts where many details have to be captured), with a macroscopic approach (to cover parts where the spatial dynamics or heterogeneity are not relevant for the simulated scenario under investigation). There are specific models, such as the one described by Lepagnot and Hutzler (2009), that combine multiple paradigms. Yet for a general applicability, a great deal of research has still to be done. For this purpose, the simulation of traffic-related systems offers great potential for the combinations of paradigms just mentioned, since many different methodologies have been successfully employed to parts of each component of such a system. A vision would be to have a library of building blocks, each of them capturing a partial model of a particular agent group, a particular environmental model, or just one specific agent, or even an activity. The development of such a library was recently mentioned in a wish list for the future of ABMS (Hamill 2010).

For large-scale agent-based models to turn into reality, massive amounts of data sets for input and empirical validation are necessary. Hence, new techniques for gathering such data have to be developed and applied. In the case of traffic simulation for example, this starts to be reality with GPS and other mobile devices for route tracking. Also, participatory simulation approaches involving stakeholders and experts into an ongoing simulation may be further developed toward new validation techniques resulting in more reliable and believable models.

As more and more researchers from the area of agent-oriented software engineering become interested in simulation applications, we expect also a methodological advance for ABMS in the future. This includes all steps from a precise formulation of the simulation objective to automatic generation of model documentation. However, for really being able to provide useful methodological support, the modeling process per se has to be better understood. Because of the generative, bottom-up nature of ABMS, emerging processes are a challenge that is often addressed by trial and error. Teaching ABMS should not be restricted to teaching how to use a particular tool or program in a particular language. Thus, Hamill (2010) identifies the availability of easy-to-use tools as one of the key challenges for the next 15 years. In addition to the previously mentioned building blocks, she sees the necessity for convincing models that are useful not only for research, but also in practice, as one of the great issues behind the development of ABMS as an established modeling and simulation paradigm.

Acknowledgements

We would like to thank the reviewers for their valuable suggestions. Ana Bazzan is partially supported by CNPq; Franziska Klügl is grateful to VINNOVA for funding her research in the VINNMER program.

Notes

1. jasss.soc.surrey.ac.uk .
2. www.swarm.org/index.php/Swarm main page.
3. www.repast.sourceforge.net.
4. www.simsesam.org.
5. cs.gmu.edu/~eclab/projects/mason.
6. ccl.northwestern.edu/netlogo.

7. see, for example, www.agent-based-models.com/blog/resources/simulators and www2.econ.iastate. edu/tesfatsi/acecode.htm.

8. www.economist.com/node/16636121.

9. www.matsim.org.

References

Arentze, T., and Timmermans, H. 2005. Representing Mental Maps and Cognitive Learning in Micro-Simulation Models of Activity-Travel Choice Dynamics. *Transportation* 32(4): 321–340.

Arthur, B. 1994. Inductive Reasoning, Bounded Rationality and the Bar Problem. Technical Report 94–03–014, Santa Fe Institute, Santa Fe, NM.

Axtell, R.; Axelrod, R.; Cohen, J. E.; and Cohen, M. D. 1996. Aligning Simulation Models — A Case Study and Results. *Computational and Mathematical Organization Theory* 1(2): 123–141.

Bagni, R.; Berchi, R.; and Cariallo, P. 2002. A Comparison of Simulation Models Applied to Epidemics. *Journal of Artificial Societies and Social Simulation* 5(3).

Balan, G., and Luke, S. 2006. History-Based Traffic Control. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems,* 616–621. New York: Association for Computing Machinery.

Balmer, M.; Cetin, N.; Nagel, K.; and Raney, B. 2004. Towards Truly Agent-Based Traffic and Mobility Simulations. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems,* 60–67. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Balmer, M.; Meister, K.; Rieser, M.; Nagel, K.; and Axhausen, K. W. 2008. Agent-Based Simulation of Travel Demand: Structure and Computational Performance of MATSim-T. Paper Presented at the 2nd TRB Conference on Innovations in Travel Modeling, Portland, June 22–24.

Barreteau, O., et al. 2003. Our Companion Modeling Approach. *Journal of Artificial Societies and Social Simulation* 6(2).

Bazzan, A. L. C. 2009. Opportunities for Multiagent Systems and Multiagent Reinforcement Learning in Traffic Control. *Autonomous Agents and Multiagent Systems* 18(3): 342–375.

Bazzan, A. L. C., and Klügl, F. 2005. Case Studies on the Braess Paradox: Simulating Route Recommendation and Learning in Abstract and Microscopic Models. *Transportation Research* Part C 13(4): 299–319.

Bazzan, A. L. C.; Oliveira, D.; and Silva, B. C. 2010. Learning in Groups of Traffic Signals. *Engineering Applications of Artificial Intelligence* 23(4): 560–568.

Bazzan, A. L. C.; Wahle, J.; and Klügl, F. 1999. Agents in Traffic Modelling: From Reactive to Social Behavior. In *Advances in Artificial Intelligence,* Lecture Notes in Artificial Intelligence 1701, 303–306. Berlin: Springer.

Burmeister, B.; Doormann, J.; and Matylis, G. 1997. Agent-Oriented Traffic Simulation. *Transactions of the Society for Computer Simulation* 14(2): 79–86.

Challet, D., and Zhang, Y. C. 1997. Emergence of Cooperation and Organization in an Evolutionary Game. *Physica A* 246(3–4): 407–418.

Chmura, T., and Pitz, T. 2007. An Extended Reinforcement Algorithm for Estimation of Human Behavior in Congestion Games. *Journal of Artificial Societies and Social Simulation* 10(2).

Dean, J. S.; Gumerman, G. J.; Epstein, J. M.; Axtell, R. L.; Swedlund, A. C.; Parker, M. T.; and McCarroll, S. 2000. Understanding Anasazi Culture Change Through Agent-Based Modeling. In *Dynamics in Human and Primate Societies: Agent-Based Modeling of Social and Spatial Processes,* 179–205. Oxford, UK: Oxford University Press.

Desjardins, C.; Laumonier, J.; and Chaib-draa, B. 2009. Learning Agents for Collaborative Driving. In *Multi-Agent Systems for Traffic and Transportation,* ed. A. L. C. Bazzan and F. Klügl, 240–260. Hershey, PA: IGI Global.

Dresner, K., and Stone, P. 2004. Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems,* 530–537. Los Alamitos, CA: IEEE Computer Society.

Drogoul, A.; Vanbergue, D.; and Meurisse, T. 2002. Multi-Agent Based Simulation: Where Are the Agents? In *Proceedings of the Third International Multi-Agent-Based Simulation Workshop* (MABS), Lecture Notes in Computer Science 2581, 1–15. Berlin: Springer.

Edmonds, B., and Moss, S. 2004. From KISS to KIDS — An Anti-Simplistic Modelling Approach. In *Proceedings of the Multi-Agent and Multi-Agent-Based Simulation Joint Workshop,* Lecture Notes in Computer Science 3415, 130–144. Berlin: Springer.

Ehlert, P. A. M., and Rothkrantz, L. J. M. 2001. A Reactive Driving Agent for Microscopic Traffic Simulation. Paper presented at the 15th European Simulation Conference, Delft, The Netherlands, 26–29 October.

Epstein, J., and Axtell, R. 1996. *Growing Artificial Societies: Social Science from the Bottom Up*. Cambridge, MA: The MIT Press.

Epstein, J. M. 2009. Modelling to Contain Pandemics. *Nature* 460 (6 August): 687.

Epstein, J. M. 2007. *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton, NJ: Princeton University Press.

Gilbert, N. 2007. *Agent-Based Models, Quantitative Applications in the Social Sciences,* volume 153. Thousand Oaks, CA: Sage Publications.

Gilbert, N., and Troitzsch, K. G. 1999. *Simulation for the Social Scientist*. London: Open University Press.

Gomez-Sanz, J. J.; Fernandez, C. R.; and Arroyo, J. 2010. Model Driven Development and Simulations with the Ingenias Agent Framework. *Simulation Modelling Practice and Theory* 18(10): 1468–1482.

Grimm, V., et al. 2006. A Standard Protocol for Describing Individual-Based and Agent-Based Models. *Ecological Modelling* 198(1-2): 115–126.

Guyot, P., and Honiden, S. 2006. Agent-Based Participatory Simulations: Merging Multi-Agent Systems and Role-Playing Games. *Journal of Artificial Societies and Social Simulation* 9(4).

Hamill, L. 2010. Agent-Based Modelling: The Next 15 Years. *Journal of Artificial Societies and Social Simulation* 13(4).

Hare, M., and Deadman, P. 2004. Further Towards a Taxonomy of Agent-Based Simulation Models in Environmental Management. *Mathematics and Computer in Simulation* 64(1): 25–40.

Izquierdo, L. R., and Polhill, J. G. 2006. Is Your Model Susceptible to Floating-Point Errors? *Journal of Artificial Societies and Social Simulation* 9(4).

Klügl, F. 2008. A Validation Methodology for Agent-Based Simulations. In *Proceedings of the 2008 ACM Symposium on Applied Computing,* 39–43. New York: Association for Computing Machinery.

Klügl, F., and Bazzan, A. L. C. 2004. Route Decision Behaviour in a Commuting Scenario. *Journal of Artificial Societies and Social Simulation* 7(1).

Klügl, F., and Bernon, C. 2011. Self-Adaptive Agents for Debugging Multi-Agent Simulations. Paper presented at the 3rd International Conference on Adaptive and Self-Adaptive Systems and Applications, (ADAPTIVE 2011), Rome, Italy, September.

Klügl, F.; Oechslein, C.; Puppe, F.; and Dornhaus, A. 2004. Multi-Agent Modelling in Comparison to Standard Modelling. *Simulation News Europe* (40): 3–9.

Klügl, F.; Fehler, M.; and Herrler, R. 2005. About the Role of the Environment in Multi-Agent Simulations. In *Environments for Multi-Agent Systems,* Lecture Notes in Computer Science 3374, ed. D. Weyns, H. V. D. Parunak, and F. Michel, 127–149. Berlin: Springer.

Kubera, Y.; Mathieu, P.; and Picault, S. 2011. IODA: An Interaction-Oriented Approach for Multi-Agent Based Simulations. *Autonomous Agents and Multi-Agent Systems* 23(3): 303–343.

Law, A. M. 2007. *Simulation Modeling and Analysis*, 4th edition. New York: McGraw-Hill.

Lepagnot, J., and Hutzler, G. 2009. A Multi-Scale Agent-Based Model for the Simulation of a Vascular Tumor Growth. *Journal of Biological Physics and Chemistry* 9(1): 17–25.

North, M., and Macal, C. 2007. *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation.* Oxford, UK: Oxford University Press.

Ören, T.; Numrich, S. K.; Uhrmacher, A. M.; Wilson, L. F.; and Gelenbe, E. 2000. Agent-Directed Simulation: Challenges to Meet Defense and Civilian Requirements. In *Proceedings of the 2000 Winter Simulation Conference* 2000, 1757–1762. New York: Association for Computing Machinery.

Panwei, S., and Dia, H. 2006. A Fuzzy Neural Approach to Modeling Behavioural Rules in Agent-Based Route Choice Simulations. Paper presented at the 4th Workshop on Agents in Traffic and Transportation, Hakodate, Japan, 9 May.

Parunak, H.; Savit, R.; and Riolo, R. 1998. Agent-Based Modeling Vs Equation-Based Modeling: A Case Study and Users' Guide. In *Proceedings of the First International Workshop on Multi-Agent Systems and Agent-Based Simulation,* ed. J. S. Sichman, R. Conte, and N. Gilbert, 10–25. Berlin: Springer.

Phan, D., and Amblard, F., eds. 2007. *Agent-Based Modelling and Simulation in the Social and Human Sciences*. Oxford, UK: The Bardwell Press.

Railsback, S. F.; Lytinen, S. L.; and Jackson, S. K. 2006. Agent-Based Simulation Platforms: Review and Development Recommendations. *Simulation* 82(9): 609–623.

Rindsfüser, G.; Klügl, F.; and Freudenstein, J. 2004. Multi-Agent Simulation for the Generation of Individual Activity Programs. In *Application of Agent Technology in Traffic and Transportation,* ed. F. Klügl, A. L. C. Bazzan, and S. Ossowski,. Basel, Switzerland: Birkhäuser, 165–180.

Rossetti, R. J. F.; Bordini, R. H.; Bazzan, A. L. C.; Bampi, S.; Liu, R.; and Van Vliet, D. 2002. Using BDI Agents to Improve Driver Modelling in a Commuter Scenario. *Transportation Research*. Part C: Emerging Technologies 10(5–6): 47–72.

Schelling, T. C. 1969. Models of Segregation. *The American Economic Review* 59(2): 488–493.

Tesfatsion, L., and Judd, K. L., eds. 2006. *Handbook of Computational Economics Vol. 2: Agent-Based Computational Economics*. Amsterdam, The Netherlands: Elsevier.

Uhrmacher, A., and Weyns, D., eds. 2009. *Multi-Agent Systems: Simulation and Applications*. Boca Raton, FL: CRC Press.

Vasirani, M., and Ossowski, S. 2011. A Computational Market for Distributed Control of Urban Road Traffic Systems. *IEEE Transactions on Intelligent Transportation Systems* 12(2): 313–321.

Wahle, J.; Bazzan, A. L. C.; Klügl, F.; and Schreckenberg, M. 2000. Decision Dynamics in a Traffic Scenario. *Physica A* 287(3–4): 669–681.

Wilensky, U., and Rand, W. 2007. Making Models Match: Replicating an Agent-Based Model. *Journal of Artificial Societies and Social Simulation* 10(4).

Windrum, P.; Fagiolo, G.; and Moneta, A. 2007. Empirical Validation of Agent-Based Models: Alternatives and Prospects. *Journal of Artificial Societies and Social Simulation* 10(2).

Yamashita, T., and Kurumatani, K. 2009. New Approach to Smooth Traffic Flow with Route Information Sharing. In *Multi-Agent Systems for Traffic and Transportation,* ed. A. Bazzan, and F. Klügl, 291–306. Hershey, PA: IGI Global.

Franziska Klügl is a full professor in information technology at Örebro University where she heads the Multiagent Simulation Group at the Modeling and Simulation Research Center. She holds a Ph,D. in computer sience from the University of Würzburg, Germany. Her research interests are in methodologies and tools for agent-based simulation including topics such as learning and adaptive agents or visual programming for modeling.

Ana Bazzan is an associate professor at the University of Rio Grande do Sul (UFRGS) in Brazil. She received her Ph.D. in 1997 from the University of Karlsruhe, Germany. From 1997 to 1998, she was a postdoctoral research associate in the Multi-Agent Systems Laboratory at the University of Massachusetts, Amherst. She has served at the IFAAMAS board and is currently associate editor of the journals *Autonomous Agents and Multiagent Systems* and *Advances in Complex Systems*. Her research interests include game-theoretic paradigms for coordination and learning in multiagent systems, and agent-based modeling and simulation.