COMPARING SYSTEM DYNAMICS AND AGENT-BASED SIMULATION TO SUPPORT STRATEGIC DECISION-MAKING

Christian Bauer
Leon Barbulovic-Nad
Freimut Bodendorf
Department of Information Systems
University of Erlangen-Nuremberg
Lange Gasse 20, 90403 Nuremberg, Germany
{Christian.Bauer|Freimut.Bodendorf}@wiso.uni-erlangen.de

KEYWORDS

Approach Comparison, System Dynamics, Agent-Based Simulation, Strategic Decision Support

ABSTRACT

System dynamics and agent-based simulation are used to explore the dynamic behaviour of complex technical and socio-economic systems. Despite different paradigms regarding system representation, model creation and simulation, both approaches have been applied successfully to support strategic decision-making. As the selection of the right modeling approach is elementary for the effectiveness of the decision support, it is necessary to understand approach assumptions and limitations.

This paper contributes to the selection of the right modeling approach by depicting differences between system dynamics and agent-based simulation. First, both approaches and the underlying paradigms are analyzed. Second, it is shown how system dynamics and agent-based simulation offer two different modeling perspectives that carry a different burden of accuracy and model complexity. Explanations are fortified with simplified scenarios and study models describing workforce and knowledge dynamics within an organization.

PROBLEM

In simulation experiments, as well as in real settings, decision makers have access to feedback information about the appropriateness of actions. The closer cause and effect are related, the more effective is the use of feedback information. Unfortunately real decision environments mostly lack this closeness between decision and feedback. It often takes a considerable time until the results caused by a decision are perceptible.

Simulation models compress time and space and thereby enable managers to learn about the effects of decisions more quickly. Using simulation models, decision makers can experiment with various strategies and learn from making rounds of decisions in an environment that allows failure and reflection (Bakken et al. 1994). However, modeling and simulation approaches are based on different methodologies so it is important that model developers and decision makers understand approach differences. They must be aware of resulting model limitations, whether simulation is used as a learning tool, training tool, or as a decision aid.

SYSTEM DYNAMICS

System dynamics applies differential equations to model the system of interest. The approach aims at explaining the system structure that causes an observed behavior. Complex systems are seen as an interlocking structure of feedback loops (Forrester 1976). The system under investigation is decomposed and the causal relationships between the identified elements are revealed.

System dynamics uses stock variables to represent the system states. A stock accumulates the influences it receives over time. The change of state that affects a stock at any point in time is described by flow variables. Flows represent the consequences resulting from actions in the system. While stocks and flows are the basis of system dynamics, auxiliary variables reflect how flows are determined. Auxiliaries are used to represent policies that manage a certain stock by controlling its corresponding inflows and outflows.

Numerical integration is used to compute the behavior of modeled systems. Based on differential equations, time is viewed as continuous. Irrespective of the used integration method, simulation is governed entirely by the passage of time. Often referred to as “time step simulation” (Coyle 1996), a number of steps along the time axis are taken during a simulation run. However, the modeler must be aware that the size of the time step influences the simulation accuracy.

AGENT-BASED SIMULATION

In agent-based simulations, individual entities are modeled to interact so that cumulative actions shape the environment that encapsulates this virtual society. An agent may represent an individual but also collectives such as firms or states, or artificial entities (Gotts et al. 2003). Agent types can vary from simple, reactive units to more complex, cognitive agents (Drogoul et al. 2003).

Agent-based models do not have a common formalism. Most formalisms are logic-based but subject to implementation differences. A typical agent-based model consists of agents, interaction environments and governing rules. Agents are usually represented as objects containing internal states and capabilities. Over time, the internal states change due to agent-agent or agent-environment interactions. Simple agent types have capabilities based on predetermined rules of behaviour so interactions are very limited while more complex cognitive agents contain adaptive methods of interaction in form of learning. Such capabilities are usually implemented using evolutionary and genetic algorithms. As interactions occur at discrete points of time a discrete-event view is adapted and implemented either using an event-scheduling or process-interaction approach.

APPROACH COMPARISON

Origin of Dynamics

In system dynamics, the basic building blocks are stocks. Being part of feedback structures they determine the system behavior. The accumulation process captured by stocks is central to the system dynamics approach. Stocks accumulate past events through inflows and outflows. Hence, actual stock value reflects the totality of all past events. This accumulation causes inertia. Assuming limited flow rates, the stock value determines how fast a given state can be changed. Creating a delay stocks absorb the difference between inflows and outflows. Stocks are often used as buffers leveling outflow rates against fluctuating inputs so stock values vary. As decision making is based on stock information, varying stocks often lead to disequilibrium dynamics being characterized by erratic system states.

In case of agent-based models, the dynamics is due to agent-agent and agent-environment interactions. Agents are the basic building blocks. Following specified rules, agents interact within their environment and thus generate the overall system behavior due to emergence. The macro-level system behavior is a result of the micro-level interactions of individual heterogeneous agents. Events that trigger reactions are the source of dynamics in agent-based simulation models. Properties of reactions are specified by the interaction rules. As agent interactions occur at discrete points of time, scheduling of events is of great importance for the emergence of the overall system behavior.

Top-Down vs. Bottom-Up Perspective

System dynamics modeling follows a top-down systems view. The high-level structure of the system is sketched providing a conceptualization of aggregate key elements and relationships. Usually the main stocks in the system are identified first, followed by the flows and the relationships that determine the flow rates. During model development initial stocks are gradually decomposed until all relevant feedback loops are captured. System dynamics seeks an endogenous explanation for a given phenomenon based on the identification of dominant feedback structures. Models developed capture emergence by modeling the phenomenon itself (Schieritz and Milling 2003).

The level of necessary model detail depends on the purpose of the model. As an example, to capture the workforce dynamics of an organization it is common to split the stock representing the company's overall employees into a promotion chain. Applying the aging chain archetype, a structure used to model items that are age-dependent, the promotion chain represents different levels in the staff hierarchy of an organization (Sterman 2000). Figure 1 depicts the structure of a two level promotion chain.

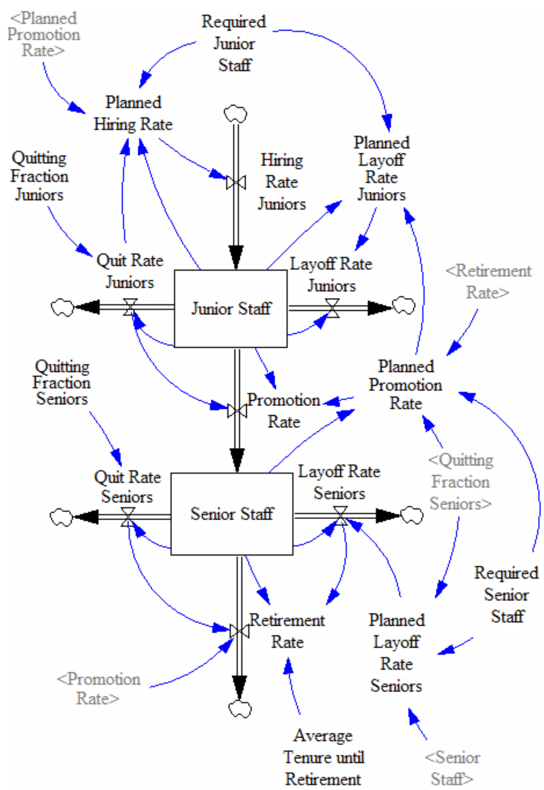

Figure 1: Capturing Workforce Dynamics Using a Promotion Chain Structure

In contrast to the top-down approach, agent-based models follow a bottom-up approach. Individuals are the most basic modeling units. The behavior of individual agents is modeled. Agents usually play different roles and it is possible that one agent is assigned multiple roles or the same agent changes roles during existence. Interactions occur according to the specified interaction rules. Groups of agents can interact with other groups creating a new level of emerging dynamics that is seen as the society behavior. In other words, characteristics of the population evolve during simulation. The bottom-up approach is a source of emergence due to interaction among agents on a particular hierarchical level – the emergence on one level causes an emergent behavior on the level above and so on.

To capture different types of agents, roles and environments, Parunak and Odell suggest the use of an UML class diagram and swimlanes (Parunak and Odell 2002). Relations are shown in a table where vertical swimlanes specify group aggregation while horizontal swimlanes specify object instantiation. An example of such a class-swimlanes diagram is shown in Figure 2.

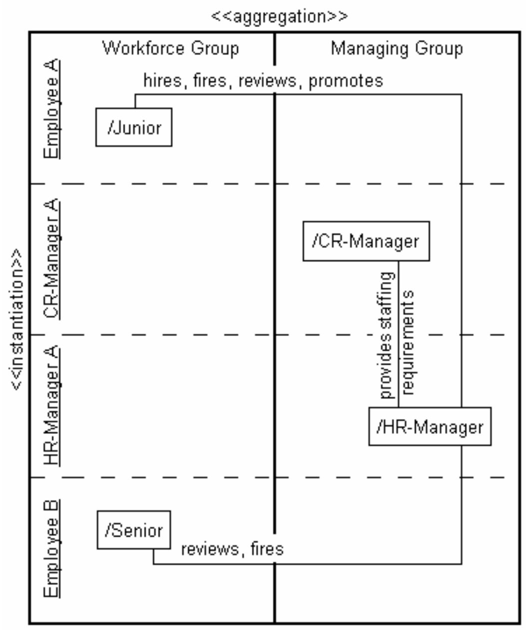

Figure 2: Simplified Class Diagram Capturing Agent-Based Workforce Dynamics

The diagram depicts a workforce structure emphasizing the bottom-up approach. On an individual level, the diagram shows four agent instantiations (Employee A, CR-Manager A, HR-Manager A, Employee B) stemming from three types of agents (Employee, CR-Manager, HR-Manager). Four roles are assigned to the agents (Junior, Senior, CR-Manager, HR-Manager). At the same time, relations between different roles are shown (e.g. the CR-Manager provides staffing requirements to the HR-Manager, an agent who hires, fires, reviews and promotes employees). According to their role agents are organized into two groups: junior and senior employees are members of the Workforce Group while the CR-Manager and the HR-Manager are members of the Managing Group.

Homogeneous vs. Heterogeneous View

Single objects flowing through a stock and flow network cannot be identified and traced in system dynamics. Stocks only represent the quantity of items contained so coflows are used to model the attributes of items in stocks. Coflow structures mirror the main stock and flow network (Sterman 2000). Attributes are modeled using corresponding stocks that keep the attribute values of the items represented in the main structure. Referring to the workforce model (Figure 1) a coflow is used to represent the knowledge of junior and senior staff (Figure 3).

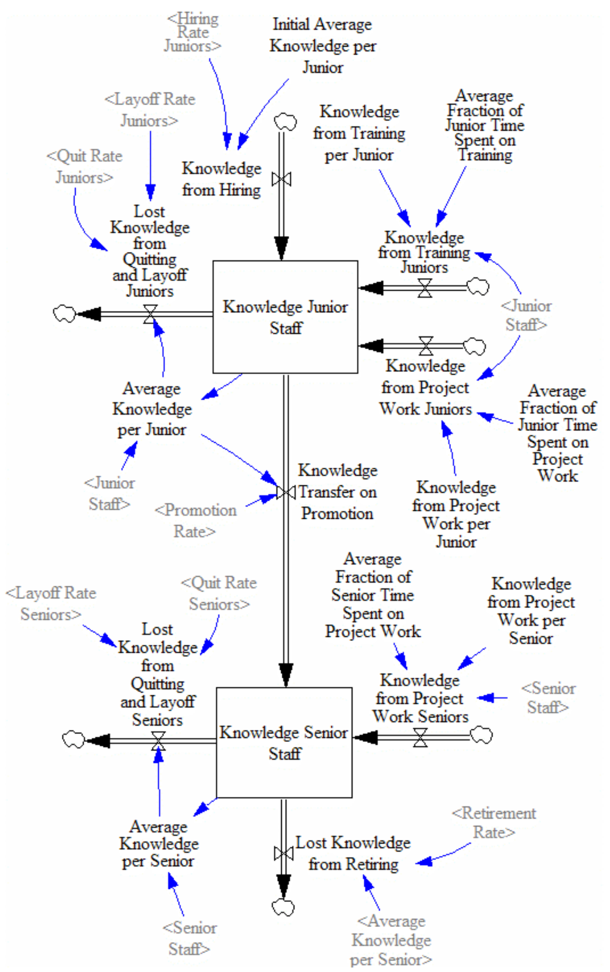

Figure 3: Modeling Workforce Knowledge Using a Coflow Structure

Two stocks are used to represent the knowledge levels of junior and senior members. Further, three inflows are modeled that contribute to the knowledge of junior staff. The first inflow is based on the hiring rate as each new employee brings in a certain amount of initial knowledge. The second inflow is due to project involvement and depends on the average time juniors spend on the project-related work. Further, training is assumed as the third source of knowledge acquisition since juniors not assigned to a project undergo training activities.

The outflow of junior staff knowledge depends on the corresponding layoff, quit and promotion rates. In all cases, the departing juniors take the average knowledge with them. In case of quitting and layoff, the knowledge is lost while in case of promotion, knowledge is transferred to the stock of senior knowledge. It is assumed that seniors are not subject to training activities so knowledge is solely acquired through project work. Similar to the juniors stock, seniors take an average amount of knowledge with them when leaving.

Modeling knowledge dynamics depicts the homogeneous perspective as an inherent aspect of the system dynamics approach. Although the coflow structure captures the overall knowledge on each job level, it is not possible to keep track of individual's knowledge attribute. Hence, the average knowledge per employee is calculated dividing the overall knowledge by the according number of staff members.

On the other hand, agent-based models provide a way of representing heterogeneity. Due to the bottom-up approach agent-based models preserve the individuality observed in real-world systems. Individual agents can have different attributes and rules of behavior depending on their internal states and the surrounding environment. Applied to knowledge modeling, this heterogeneous individual-based perspective provides the opportunity to model individual knowledge sets for each agent. For example, instead of assuming an average knowledge level for every hired junior employee, it is possible to assign individual knowledge sets thereby creating a heterogeneous agent population. Furthermore, the development of individual knowledge sets can depend on agent interaction with other knowledge repositories in the environment. Using unique sets of natural numbers to represent distinct knowledge items, comparison and modification of individual items within knowledge repositories is feasible.

Figure 4 depicts various agent interactions with knowledge repositories during project and knowledge-management activities. The figure shows how two agents A and B mutually interact or manipulate project and organizational knowledge repositories by transferring or creating knowledge items. A dashed arrow symbolizes a flow of knowledge from the source to the destination. A solid arrow represents knowledge creation where the tail depicts the origin of the newly created idea while the head identifies the location where the created knowledge is temporally or permanently stored.

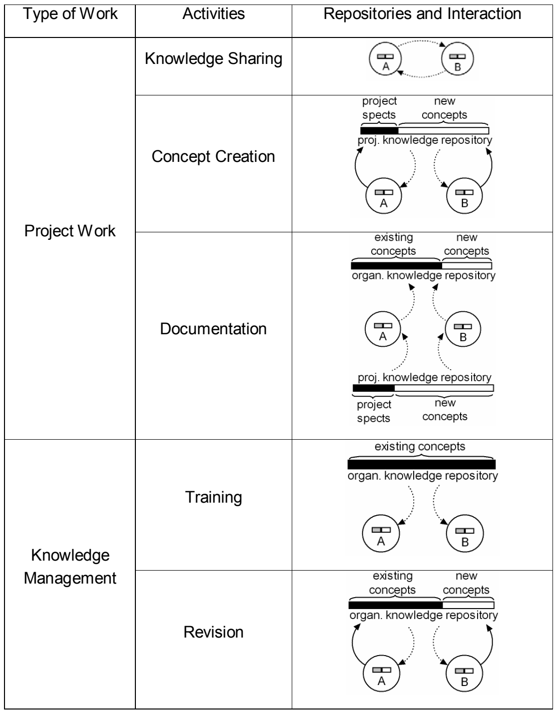

Figure 4: Modeling Knowledge Repositories and Agent Interaction

During project work three activities contribute to the knowledge creation and transfer. First, agents being assigned to a project share some of their individual knowledge while collaborating. Second, the project itself is regarded as a source of knowledge. Agents are confronted with new topics and problems, two additional sources that enlarge their knowledge repositories. While solving problems at hand, agents create new concepts contributing to the project knowledge repository. Third, it is assumed that agents spent some time documenting project findings while transferring the new knowledge items to the organizational knowledge repository.

Agents not involved in project work spent their time on knowledge management activities. Training time is allocated to allow juniors to read and accept approved concepts that are documented as part of organizational knowledge repository. Seniors spend time reviewing already documented concepts while creating new ideas – concepts are merged and extended into new knowledge items. However, all knowledge transfer and creation rates have an upper limit as a way of representing an agent's cognitive limits and the resource use restrictions. The individual's learning capabilities, group contributions to ongoing projects, project contributions to organizational knowledge, as well as knowledge management efforts such as training and revision are limited.

Model Accuracy

To elaborate on model accuracy, the simulation results of the simple workforce and knowledge models, which have been introduced above, are presented. Continuous workforce growth is assumed for the entire simulation time of ten years. In order to ensure equivalent initial conditions, average initial knowledge levels of the instantiated junior and senior agents are used for parameterization of the system dynamics model. The quit rates and the retirement age are set the same. Figure 5 shows the simulation results of both models for the average junior knowledge.

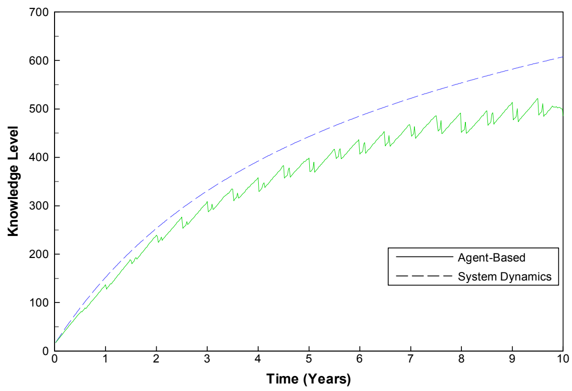

Figure 5: Average Junior Knowledge

The results reveal that compared to the agent-based simulation, the system dynamics model overestimates junior knowledge values. Due to the aggregate and homogeneous modeling perspective of the approach, an average knowledge value is removed from the junior knowledge stock in cases of quitting, layoff and promotion. However, the use of the average knowledge is clearly an oversimplification of the amount of knowledge that is lost or transferred to the senior level.

Although the system dynamics approach cannot handle individual items, the use of an estimated attribute distribution over the total amount of items leads to more sophisticated results. Instead of assuming that juniors possess the same average amount of knowledge, a uniform distribution is implied. Minimum knowledge is calculated using a moving average of the initial average knowledge per hired junior. The time span taken for the moving average equals the average tenure of juniors. Maximum knowledge is given by doubling the difference between the average knowledge of all juniors reduced by the moving average of the initial knowledge per junior. Figure 6 shows the simulation results of the improved model versus the agent-based simulation.

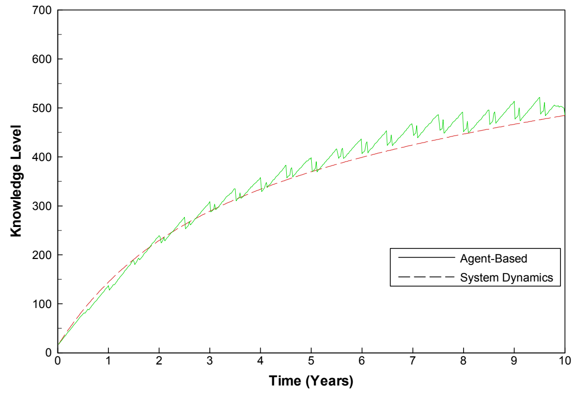

Figure 6: Average Junior Knowledge Using Uniform Attribute Distribution in the System Dynamics Model

Uniform attribute distribution results are in better agreement with the agent-based simulation results. Nevertheless, in a case of heterogeneous attribute values, the agent-based model provides a more accurate result than the system dynamics model. The aggregate and homogeneous view inherent to the system dynamics approach reduces accuracy compared to the individual and heterogeneous view of agent-based simulation.

Model Development

Using simulation models to support managerial decision making, two development aspects are important. First, seriously impinging the applicability of models as decision-making tools, efforts necessary to develop appropriate models are considered complex and time-consuming. Second aspect not to be ignored is the client participation. An effective simulation-based tool requires from the decision makers to accept the simulation model as an adequate representation of the problem. Fostering management participation in model development promotes the acceptance of resulting simulation models (Lane 1994). However, modeling only becomes an active part of decision support if the selected approach is simple enough to allow decision makers to participate in the model formulation and implementation without technical savvy.

Compared to the agent-based models, the development of system dynamics models takes significantly less time. In addition, modeling elements are easy to understand on a qualitative level. Causal loop and stock and flow diagrams allow a graphical model development that reveals the overall model structure. Easy-to-use workbench tools are available for model specification, implementation, execution and documentation as well as visualization and analysis of the simulation results. Nevertheless, modeling and especially the development of mathematical models are tasks left to modeling experts since comprehensive experience is necessary to develop valid models in a reasonable amount of time.

The development of agent-based models proves to be more involved and more time consuming as it is far more concerned with programming details. Standard modeling approaches such as UML, which are increasingly being used for business modeling and modeling of other non-software systems, contributes to the facilitation of agent-based modeling. Advances in graphical representation of agent-based concepts based on UML (e.g. AUML) facilitate the collaboration between developers and end-users. Following well-established standards, models will be easier to use, communicate and understand.

Model Analysis

To gain valuable insights from model analysis, it is important to have an understanding of the model structure and its behavior. Model transparency and the ability to trace the causes of a given behavior are indispensable. The graphical representation of system dynamics models supports this comprehension. The perception and communication of such graphical models is easy and identification of dominant and critical loops is possible. Due to the top-down approach, system dynamics already assures a basic understanding of the model structure as model development starts with an aggregate, hence less complex, view of the system while details are added step by step.

Work with agent-based models is somewhat more difficult. These difficulties are partially due to the bottom-up approach that requires more modeling details but such approach sometimes provides a more intuitive way of system representation. Agents are usually related to corresponding real-world objects or concepts, e. g. persons, machines, orders, etc. While being true regarding the analysis of individual agent interaction, this is questionable for the analysis of emergent behavior on the systems level. Following the bottom-up approach, the number of agents could easily reach into several hundreds, leaving the number of interactions as a multiple. Wooldridge and Jennings conclude that "[…] the dynamics of multi-agent systems are complex, and can be chaotic. It is often difficult to predict and explain the behavior of even a small number of agents; with larger numbers of agents, attempting to predict and explain the behavior of a system is futile" (Woolridge and Jennings 1998, pg. 5).

CONCLUSIONS

System dynamics and agent-based simulation offer two different modeling perspectives that carry a different burden of accuracy and model development time. System dynamics aids the understanding of complex system structures so that effective policies can be designed and targeted toward the most rewarding goals. However, regarding the model accuracy, the agent-based models outperform the system dynamics approach. Based on an aggregate view, system dynamics captures only homogeneous groups of objects whose members are not distinguishable. Agent-based models in contrast maintain individual heterogeneity implementing different agents. System dynamics models run the risk of oversimplification while agent-based models in contrast have to cope with complexity. Complexity reduces the ability to quickly identify relevant behaviors and the corresponding factors of influence. Understanding where the agent-based approach yields additional insight and where details have no importance is crucial in selection of the appropriate method (Rahmandad and Sterman 2004).

REFERENCES

Bakken, B.; Gould, J.; Kim, D.: Experimentation in Learning Organizations: A Flight Simulator Approach. Morecroft, J.; Sterman, J. (Eds.): Modeling for Learning Organizations. Portland: Productivity Press, 1994, pp. 243-266.

Coyle, R. G.: System Dynamics Modelling: A practical approach. London: Chapman & Hall, 1996.

Drogoul, A.; Vanbergue, D.; Meurisse, T.: Multi-Agent Based Simulation: Where are the Agents? Sichman, J. S.; Bousquet, F.; Davidsson, P. (Eds): Multi-Agent-Based Simulation II: Third International Workshop (MABS 2002), Lecture Notes in Computer Science 2581 (2003), pp. 1-15.

Forrester, J. W.: Urban Dynamics. Cambridge: MIT Press, 1976.

Gotts, N. M.; Polhill, J. G.; Law, A. N. R.: Agent-Based Simulation in the Study of Social Dilemmas. Artificial Intelligence Review 19, No. 1 (2003), pp. 3-92.

Lane, D. C.: Modeling as Learning: A Consultancy Methodology for Enhancing Learning in Management Teams. Morecroft, J.; Sterman, J. (Eds.): Modeling for Learning Organizations. Portland: Productivity Press, 1994, pp. 85-177.

Parunak, H. V. D.; Odell, J.: Representing Social Structures in UML. Wooldridge, M.; Ciancarini, P.; Weiss, G. (Eds): Agent-Oriented Software Engineering Workshop II. Berlin: Springer, 2002, pp. 1-16.

Rahmandad, H.; Sterman, J.: Heterogeneity and Network Structure in the Dynamics of Diffusion: Comparing Agent-Based and Differential Equation Models. MIT Sloan Working Paper 4512-04, 2004.

Schieritz, N.; Milling, P. P.: Modeling the Forest or Modeling the Trees – A Comparison of System Dynamics and Agent-Based Simulation. Proceedings of the 21th International Conference of the System Dynamics Society (2003).

Sterman, J. D.: Business Dynamics, Systems Thinking and Modeling for a Complex World. McGraw-Hill/Irwin, 2000.

Wooldridge, M.; Jennings, N. R.: Pitfalls of Agent-Oriented Development. Proceedings of the Second International Conference on Autonomous Agents (Agents 98) (1998), pp. 385-391.