Agent Based Modeling to Inform the Design of Multiuser Systems

Yuqing Ren and Robert E. Kraut

Why Agent-Based Modeling?

Decades have passed since the inception of the field of Human–Computer Interaction. The emergence of the Internet has shifted researchers' attention from understanding how individuals interact with computers to understanding how individuals interact with one another using computer technologies. A wide range of systems designed for multiple users have been labeled as groupware, collaborative computing, multiuser applications, and more recently social computing technologies. Designing these types of system is more challenging than designing single-user ones because other people and their behaviors are integral elements of the system as experienced by users (Grudin, 1994). The system itself, therefore, is non-deterministic and evolutionary because the experience of some users is partly the result of decisions that earlier users have made. Because the behavior of a multiuser system is not stable until a critical mass of users has developed a routine way of using it, it is difficult to predict how groups of users will respond to a particular design before the stable state is reached. As a result, interactive design and evaluation cycle, perhaps the most successful HCI technique for system design, is insufficient for the design of multiuser applications.

Consider the design of an online health support group like breastcancer.org and the decision about whether to employ moderators who ensure group members spend their time discussing cancer-related topics, channel off-topic content to sub-forums, or prevent users from posting advertisements. A member's decision to participate in

Y. Ren (✉)
Carlson School of Management, University of Minnesota,
321 19th Avenue South, Minneapolis, MN 55455, USA
e-mail: chingren@umn.edu

R.E. Kraut
Human–Computer Interaction Institute, Carnegie Mellon University,
5000 Forbes Ave., Pittsburgh, PA 15213, USA
e-mail: robert.kraut@cs.cmu.edu

J.S. Olson and W.A. Kellogg (eds.), *Ways of Knowing in HCI*,
DOI 10.1007/978-1-4939-0378-8_16,

the community depends in part upon the content that other members post and what moderators, if they are used, allow. But how should the designers go about deciding whether moderators will improve the site? Building different versions of the site with alternative design options would be impractical and costly. Another solution is to use computational modeling to simulate a system before building it. The simulation can be used to run virtual experiments to evaluate users' likely reaction to alternative design choices and to predict how it will actually be used under various scenarios. Assuming that the simulation can replicate known patterns of behavior in the phenomena it attempts to replicate, it also can be used to predict reactions to as yet undeveloped features.

A computer simulation is a program that embodies a partial theory of how some phenomenon operates. The method has been used for decades by social scientists to understand a wide variety of social dynamics and processes. For example, Schelling (1971) created a simple model to show how residential segregation can emerge even when most members of a community would tolerate living in an ethnically mixed environment. It is "runnable" in the sense that a scientist can turn on and off or vary input parameters (e.g., the initial sizes of the ethnic groups, the strength of members' preference for diversity or speed of housing turnover), and the simulation will generate output to predict e.g., the extent to which the society will become segregated.

A similar approach can be taken to study HCI phenomena characterized by bottom-up, self-organizing, and complex interactions among individual users. For example, the use of social media such as wikis, blogs, social networking, and social bookmarking has become very prevalent in many organizations (Treem & Leonardi, 2012) and has attracted great interest from HCI researchers (e.g., DiMicco et al., 2008; Shami, Ehrlich, Gay & Hancock, 2009; Thom-Santelli, Millen & Gergle, 2011; Wu, DiMicco & Millen, 2010). Simulation can answer questions such as the following: How does usage spread within an organization? What patterns will emerge in the use and adoption of these technologies by individual users? How will the adoption and use of such technologies change organizational hierarchy? How can an organization align system design, incentives, and its culture and policies to encourage effective use of the technologies?

Scientists and engineers have built several genres of simulation to simulate social systems, including statistical, causal models, mathematical models, system dynamics models, neural networks, cellular automata, multilevel simulations, evolutionary models, and agent-based models (Taber & Timpone, 1996). In this chapter we focus on agent-based modeling as a tool to inform the design of multiuser systems and to advance our knowledge of how these systems operate because of the isomorphism between the systems we are attempting to simulate and the simulation techniques. An agent-based model simulates a multiuser system by modeling the behaviors of and interactions among individual users who comprise the system. We start with a brief review of the method, followed by our key contribution, a seven-step roadmap that HCI researchers can follow to build or evaluate agent-based models. We then describe how we followed the steps and built an agent-based model that can inform the design of online communities. In the end, we share a personal account of how we encountered the method and include references for readers who would like to learn more about the method.

What is Agent-Based Modeling?

Agent-based modeling is a form of computational simulation that "enables a researcher to create, analyze, and experiment with models composed of agents that interact within an environment" (Gilbert, 2008). The agents can imitate a wide variety of physical and social entities such as human beings, animals, particles, or molecules. Agent-based modeling is similar to mathematical modeling in terms of rigor but better suited for situations when agents are autonomous and heterogeneous, when there are complex interactions between agents, and when lower-level actions and interactions can lead to the emergence of system-level structures. Compared with conventional methods of developing theories in social sciences, agent-based modeling is especially suitable for bottom-up theorizing (Kozlowski & Klein, 2000), and for understanding how individual agent behaviors interact over time and lead to emergent system-level patterns.

The system-level regularities are often the results of multiple forces working together. The tension among the forces may be temporal, structural, or spatial and often result in nonlinear relationships like tipping points (Davis, Eisenhardt & Bingham, 2007). A famous example is Reynolds' "boids" model (1987) that simulates the behaviors of flocks of birds. The agents in this model are birds with limited perception programmed with three simple rules as illustrated in Fig. 1: separation to avoid getting too close to other birds, velocity to travel at the speed of nearby

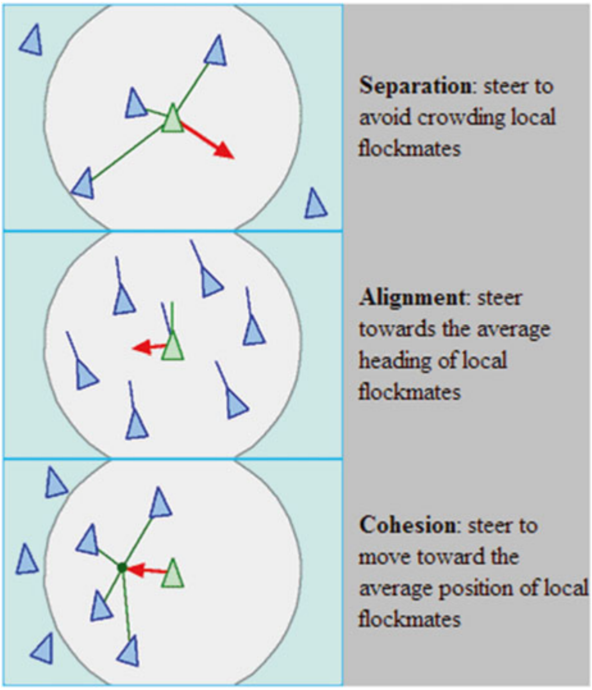

Fig. 1 Illustration of Reynold's boids model (Reprinted with permission from http://www.red3d.com/cwr/boids/)

flockmates, and cohesion to head for the perceived center of nearby flockmates. The model does a remarkable job of replicating how flocks of birds fly together without bumping into each other. This tension of wanting to follow the crowd but not get too close applies to many human groups, too.

The history of agent-based models dates back to Von Neumann in the late 1940s, when he developed a machine that was capable of self-replicating (Gilbert, 2008). His creation of a self-replicating automaton without a computer eventually led to the creation of cellular automata, a popular technique for doing agent-based modeling by placing individual agents on a two-dimensional lattice or grid of cells and observing what patterns emerge as they interact with neighbors (Davis et al., 2007). The idea motivated the creation of Conway's Game of Life (Gardner, 1970) and gradually, the method morphed its way from mathematics into economics, social science, and other disciplines. The social science version of the game is called the Sugarscape model, created by Epstein & Axtell (1996) to simulate and study human societies.

In the past two to three decades, agent-based modeling has become much more widespread due to the exponential growth of computing power. A wide variety of models have been developed to simulate physical and social phenomena, such as flow in sand piles and the activities of animals such as birds and ants (Sawyer, 2003), social and organizational behaviors in cooperation and collective action (Macy, 1991), learning (March, 1991), social influence and norm formation (Axelrod, 1986; Axelrod, 1997a, 1997b), cultural dissemination (Harrison & Carroll, 1991), and innovation diffusion (Strang & Macy, 2001).

How Can Agent-Based Modeling Inform HCI Theory and Design?

Agent-based modeling can be used for a wide range of purposes such as *description* of behaviors and *training* managers to make better decisions (Burton & Obel, 1995), *development* of theories of the conditions or mechanisms that generate certain behaviors (Davis et al., 2007), *discovery* of unexpected consequences of local interactions, and *prescription* to suggest better modes of operation or organization (Harrison, Lin, Carroll & Carley, 2007). We believe there are at least two important ways in which agent-based modeling can be leveraged in HCI research: to advance theories related to multiuser systems and to inform the design of these systems as well as interventions, policies, and practices surrounding them. The former corresponds to the use of agent based modeling to *explain* mechanisms, processes, or conditions that lead to certain behaviors and the latter corresponds to the use of agent-based modeling to *prescribe* actions to obtain desired outcomes.

A good example to illustrate the use of agent-based modeling to *advance theory* is the Shape Factory model developed by Nan and colleagues (2005, 2008). Their research began with a laboratory experiment to investigate how geographic separation influences the performance of collocated and remote workers. Ten participants, five collocated and five remote, earn points by making and buying parts of different shapes to fill "customer" orders. A puzzling pattern emerged, in which collocated and remote players were equally successful even though collocated players had communication advantages. Two theoretically plausible mechanisms—in-group favoritism and communication delay—could have been at work, but they were confounded in the experiment, making it impossible to isolate their independent effects.

Agent-based modeling is well suited for addressing such challenges because it grants researchers the ability to computationally turn a mechanism on and off and observe how outcomes change as a result. Using behavioral patterns observed in the lab experiments as benchmarks, Nan and colleagues (2005, 2008) developed an agent-based model to separate the effects of in-group favoritism and communication delay. They implemented the two mechanisms as two behavioral rules: in-group favoritism meant collocated agents always transacted business with other collocated agents before contacting remote agents; communication delay meant a one-step time delay in communications with all remote agents. Their simulation results suggested in-group favoritism actually had a detrimental effect on the performance of collocated players (by limiting themselves to transact with only local agents) although (lack of) communication delay had a positive effect on their performance. The two effects cancelled out each other in the laboratory experiment and would be hard to disentangle without agent-based modeling.

The Nan study illustrates how agent-based modeling can be used to complement other empirical methods, in this case laboratory experiments, to enrich our theoretical understanding of the working of multiuser systems. As Davis et al. (2007) suggest, simulation occupies a "sweet spot" between theory-creating methods such as case studies and formal modeling, and theory-testing methods, such as survey and experiments. The model needs to be grounded in theoretical insights and empirical evidence, and in turn it can expand our understanding beyond the conditions that were observed in early research.

Because researchers in management, public policy, and sociology have already documented how to use simulation to develop and test theories (Axelrod, 2005; Davis et al., 2007; Harrison et al., 2007; Macy & Willer, 2002), in this chapter, we focus on the use of agent-based modeling in HCI research to inform the design of multiuser systems and policies and practices surrounding them. We use online communities as an example of multiuser systems. A key challenge in designing online communities is that designers must make numerous decisions about features, structures, and policies. Even experienced designers can be overwhelmed by the trade-offs involved in the decisions and fail to anticipate how users will respond. For instance, when launching an online community, if a community offers points for contributions and recognizes the most active contributors on a public "leader board," this feature may encourage the least active participants to increase their level of contribution, and heavy contributors to contribute less if the former perceive themselves as under-contributing and the latter perceive themselves as over-contributing; moreover, it could discourage most community members from contributing at all if they perceive that the leaders are providing sufficient content.

These contradictory predictions originate from two social science theories: social comparison theory (Festinger, 1954) and the Collective Effort Model (Karau & Williams, 1993). The former argues that people are motivated to match their performance to the performance of similar others, and thus increase their effort when being told that others have contributed more than they have (Harper et al., 2007). The latter argues that people exert less effort when working in groups than individually because they perceive their efforts are unnecessary to achieve group outcomes. Perhaps because of contradictory predictions like these, theories from social psychology, organizational behavior, sociology, and economics have been applied to describe behaviors in online communities, more than they have been applied prescriptively, to offer solutions for building successful communities (see Ling et al., 2005, for exception Kraut & Resnick 2012).

An important reason that these social science theories seem ill suited for design is that the logic of design, which manages trade-offs among tens or hundreds of parameters that can influence members' behaviors, is at odds with the logic of social science research, which examines the influence of a small set of variables while holding everything else equal. Agent-based models can bridge this gap, by synthesizing insights from multiple theories to identify the pathways through which particular design choices will affect the different outcomes that designers aim to achieve. In other words, agent-based models can be used to link and integrate what Davis et al. (2007) termed "simple theories" to infer "the combined implications of several theoretical assumptions or empirical results" (Taber & Timpone, 1996, p. 6).

What Constitutes Good Work: A Seven-Step Roadmap

In this section, we provide a roadmap with a set of guidelines HCI researchers can follow to build agent-based models, as shown in Table 1. To make the guidelines concrete and accessible, we use our personal experience to demonstrate how we followed these guidelines and built an agent-based model to inform the design of text-based online communities. We assume that you already have a research question and wonder if agent-based modeling is the appropriate method to study it.

Step 1: Evaluate the Appropriateness of Agent-Based Modeling for the Research Question

Whether agent-based modeling is appropriate for your research depends on several factors: the phenomenon of interest, the level of analysis of your research questions,

Table 1 Roadmap for using agent-based modeling (ABM) to inform HCI design

Steps	Activities/questions
Evaluate the appropriateness of ABM for your research question	– Can the overall system behavior be decomposed into decisions and actions by autonomous interacting agents? – Are their decisions and actions influenced by multiple forces? – Is the system likely to be multilevel, nonlinear, and dynamic? – Are there simple theories or empirical evidence available to ground the model?
Define boundary conditions and build a conceptual model	– Decide the scope of the model (types of agents, types of objectives, types of agent behaviors, the larger environment) – Identify theories to help construct the conceptual map – Identify key variables in the conceptual map – Start with a simple model and gradually expand
Translate the conceptual model to computational representations	– Operationalize three key elements: agents, environment, and timescale – Translate theories to behavioral rules governing agents' motion, communication, and action – Simulate time as forced parallel
Implement the model	– Decide whether to use an existing platform or build from scratch – Compare and choose a platform if needed – Program, debug, test the program
Validate the model	– Check program to make sure that it is an accurate translation of the conceptual model and is bug free – Calibrate the model by modifying the model to match theory predictions, stylized facts, or empirical training data – Test the external validity of the model by comparing simulation results with theory or empirical testing data
Experiment with the model	– Design virtual experiments (determine key factors and their values or range and number of runs) – Set parameters with theoretical or real-life values – Run experiments and gather output data
Publish the model and results	– Provide sufficient detail for others to replicate the model – Arrange to share the source code – Discuss practical as well as statistical significance

and the working body of knowledge from which you can borrow insights to ground the model. Phenomena well suited for agent-based modeling typically have the following characteristics:

1. They involve the actions and interactions of individual agents.
2. Individual agents have heterogeneous motivations, interests, or behaviors.
3. Individual agents form a large social system whose structure is determined by individual actions and the size and structure of the social system, in turn, shape individual behaviors.

4. The system dynamically evolves over time as individual agents interact with one another, and as a result, it can be characterized as multilevel, nonlinear, and dynamic.

Here are some sample HCI problems that agent-based modeling may help tackle:

- *Attention management in communication exchanges.* Spontaneous, informal communication at work is important, yet, it often helps the initiator of the communication at the expense of the person being interrupted (Perlow, 1999). Interventions designed to balance the benefits and costs of spontaneous communication have often had unforeseen consequences. For example, pricing systems that increase the cost of interruptions can reduce the volume of communication below optimal levels (e.g., Kraut et al., 2002) or awareness displays that show when someone is interruptible can increase instead of reducing interruptions (e.g., Fogarty, Lai & Christensen, 2004). Agent-based models can help predict the long term impact of alternative interventions.
- *Feedback mechanisms in online contribution.* Online production communities like Wikipedia need high quality contributions. Interventions that aim to increase quality often have unintended consequences on the contributors. For example, making new members pass a quality test can increase their quality and motivation but reduce the number of members who join (Drenner, Sen & Terveen, 2008), and giving contributors corrective feedback may direct their attention away from the task and towards themselves and harm their performance (Kluger & DeNisi, 1996). Agent-based models can help community leaders manage these trade-offs.

To reiterate, several common themes run through these examples that make them appropriate for agent-based modeling. First, the phenomena are generated *bottom-up,* in the sense that individuals make autonomous decisions and the outcome—whether it is the success or failure of a system or communication patterns—are jointly determined by individual actions and interactions. Second, *multiple forces* drive individual behaviors, implying the model needs to combine multiple theories to be a valid representation of reality. Finally, the system-level regularities cannot be intuitively predicted based on rules for individual actions because the multiple forces affecting behaviors may work in opposite directions or cancel each other out.

These examples draw upon relatively mature theoretical and empirical understanding of the phenomena being studied. Such understandings should be past the exploratory stage, with sufficient literature available to ground the model. It is ideal to have multiple theoretical propositions or empirical results, none of which seems capable of explaining the observation alone but collectively have the potential to do so (Taber & Timpone, 1996). For example, an agent-based model to simulate how starting conditions influence a community's success can rely upon a rich literature on critical mass (Markus, 1987), network externalities (e.g., Shapiro & Varian, 1999), organizational ecology (Hannan & Freeman, 1989), and group commitment (e.g., Mathieu & Zajac, 1990). It is also desirable to have ways to gather new empirical data to fill in detail of the model where theories are silent or fail to provide detail to specify functions or parameters.

Step 2: Define Boundary Conditions and Build a Conceptual Model

Agent-based models, like mathematical and statistical models, are a simplified representation of reality. It is crucial to clearly define the boundaries of a model to capture the essence of the phenomenon being studied. Many multiuser systems are complex by nature, involving agents in different roles, artifacts of different types, and complicated connections between agents, artifacts, and their environment. For example, work in Wikipedia occurs in 270 different languages and depends upon the contribution of tens of thousands of volunteer editors who take on a variety of tasks from creating new articles to writing policies (Bao et al., 2012; Welser et al., 2011). The editors are organized into hundreds of subgroups known as WikiProjects, and they collaborate on a technical infrastructure run by a non-profit organization. The content ultimately becomes viewable to tens of millions of Internet users. If you were to build an agent-based model to understand the working of Wikipedia collaboration, where should you draw the boundaries? Besides editors and articles, should Wikipedia readers or other agents like the bots (automated programs) that repair vandalism be explicitly modeled? What about higher-level social entities like WikiProjects or the Wikimedia Foundation, which supports Wikipedia's infrastructure?

These are nontrivial decisions, and the answers are not straightforward. Trade-offs between simplicity and reality or between parsimony and accuracy plague agent-based modeling. An agent-based model, while needing to be sufficiently comprehensive and complete to be accurate, also needs to be a simplified representation of reality to be useful (Gilbert, 2008). Complex models can be more accurate in their predictions but are more difficult to debug and may become so incomprehensible that readers or even its developers have difficulty deciphering how variations in the model's features lead to its results. The right decision, therefore, requires a balance of capturing the central phenomenon while stripping away the nonessentials. To a large extent, this balancing act is a judgment call (Davis et al., 2007) or "the art of simulation" (Harrison et al., 2007). There are no universally correct answers; settling on one depends on individual researchers' preferences and style of research.

Some modelers lean toward simplicity. Simple models are especially good for theory development, exemplified by Schelling's (1971) racial segregation model. Simple models can be quite powerful if unexpected system-level patterns can be generated with simple rules at the agent level. On the other hand, simple models often fall short of making accurate predictions to guide practice because they fail to incorporate all the important forces or mechanisms driving a phenomenon. Agent-based modeling needs a reasonable level of complexity to provide useful guidance to design. Even when building complex models, a good practice is to start with a simple model and gradually expand to add more fidelity.

Once the boundaries are established, researchers can identify the important concepts they want to capture in the model and their relationships. Taber and

Timpone (1996) suggest the practice of creating "an inventory of concepts on paper when dealing with a complex model" (p. 15). This concept inventory should define the concepts in qualitative terms and propose a loose notion of how they might be operationalized. Researchers should consider theories from multiple disciplines to ground their models, because social behaviors and processes cannot be decomposed into separate subprocesses that neatly match the artificial divisions of different disciplines (Epstein, 1999). Individual behaviors can be driven by economic, psychological, political, and technological factors, and researchers should not let disciplinary boundaries prevent them from identifying important aspects of a phenomenon.

Step 3: Translate the Conceptual Model into Computational Representations

The next step is to operationalize the conceptual model by translating theoretical relationships to assumptions, agent attributes, and behavioral rules. Gilbert (2008) identified three key elements to specify an agent-based model: agents, environments, and timescales. An agent can be a person, animal, or object. Agents imitating humans can engage in the following activities:

- Perceive the environment including the presence of other agents or objects in their neighborhood.
- Perform a set of behaviors, such as moving within a space, communicating (sending messages to and receiving messages from other agents), acting or interacting with the environment (such as joining a group or contributing information to a corporate wiki).
- Remember their previous states, actions, or consequences (e.g., for learning purpose).
- Follow policies or adopt strategies that determine what actions to take next.

In the Shape Factory simulation, agents' perceptions of the environment include the awareness of collocated and remote players and the shapes they produce. Agents could not move but could communicate by sending and fulfilling shape requests. Agents did not have memories and could not learn from past behaviors. They did not engage in sophisticated strategies although they had a goal of maximizing the number of orders they filled.

In more complex models, agents could engage in more sophisticated behaviors. For instance, in an model developed to study transactive memory, agents possessed knowledge about both their own and other agents' areas of expertise (Ren, Carley & Argote, 2006). The transactive memory enabled agents to efficiently search for information and assign tasks to those with specialized knowledge.

Timescale is another key element in translating theories to agent behaviors. The order in which agent behaviors occur can significantly influence simulation results because output from early agents changes the environment that other agents experience later. For example, in simulating an online discussion group, researchers must consider whether a post will be broadcast to all other agents immediately after it is posted or kept in a repository until all agents finish the current round of activities. The choice arises because parallel processing is computationally costly, but less costly modeling procedures can approximate it. For example, researchers can buffer all interactions in the environment (e.g., recent messages posted in a community) and wait until all actions are completed before displaying the new interactions (i.e., new messages) to all agents. In the jargon of agent-based modeling, actions can be organized in staged episodes, with time simulated as "forced parallel." This technique is also called "simulated synchronous execution" (Gilbert, 2008). A useful tool in model design is a flowchart, which shows the sequence of actions together with the conditions under which a rule applies or an action is taken.

Step 4: Implement the Model

The implementation step is often mistaken as the core of agent-based modeling (all you need to do is to write computer code which will generate tons of numbers, right?). Implementation is important in the sense that the modeler must accurately translate the conceptual model to computer code so that the program runs efficiently and is bug free. Compared to the conceptual model, however, implementation is less important. Without a valid conceptual model, whatever data you get from the simulation will be worthless, i.e., "garbage in, garbage out."

There are typically two options for implementing an agent-based model: build on an existing simulation platform like NetLogo (Wilensky, 1999) or code it from scratch using a general-purpose computer language like Java, Python, or C++. This choice determines the user interface that the modeler will use to interact with the model. Myers and colleagues (2000) have identified criteria for evaluating user interface software. Criteria that are especially important in choosing tools for agent-based modeling include the software's threshold (i.e., the difficulty in learning the system and building initial software), ceiling (how much can be accomplished with the software), path of least resistance (i.e., whether the software helps the user produce appropriate models), and stability (i.e., whether the software is changing too rapidly for its users to gain significant experience with it).

Each approach has its pros and cons. Which to choose depends on the complexity of the model, the researchers' timeline and programming skills, and the extent to which a decent user interface and add-on features such as network analysis and visualization are needed. For beginners, we recommend building on a platform such as NetLogo (http://ccl.northwestern.edu/netlogo/), Repast (http://repast.sourceforge.net/), or Mason (http://cs.gmu.edu/~eclab/projects/mason/), especially if the model is simple, can be built using standard modules or if researchers have limited programming skills. Even experienced programmers can save a great deal of time and effort by building on a platform. On the other hand, if the model is complex and requires functions that are unavailable in established platforms, it can be difficult to coerce the platform to fit your purpose. If so, building from scratch may be the only option. The first author's experience of developing a model to simulate transactive memory systems fits this category. A key element of the model was the concept of transactive memory that stores information about other agents' areas of expertise (Ren et al., 2006). Existing platforms did not have a built-in module that can be modified to simulate the working of a transactive memory system. Therefore, we built our own model using Java. It was a multiyear effort (close to 3 years including model validation) but it allowed us to capture the core concept we wanted to study.

Detailed reviews and comparisons of agent-based modeling platforms are available from several sources. Gilbert (2008) compares four platforms—Swarm, Repast, Mason, and NetLogo—on user base, speed of execution, support for graphical interface and systematic experimentation, and ease of learning. He reports that "NetLogo stands out as the quickest to learn and the easiest to use, but may not be the most suitable for large and complex models. […] Repast has the advantage of being the newest … but also has a significantly smaller user base, meaning that there is less of a community that can provide advice and support" (p. 49). Our own experiences with the two platforms are consistent with Gilbert's assessment. In addition, we recommend Repast for building large, complex models that are computationally demanding.

Step 5: Demonstrate the Internal and External Validity of the Model

The next step is to ensure that the model is a valid representation of reality. Doing so consists of three processes: verification, calibration, and validation. Model *verification* involves checking that the agent-based model satisfies its specification, is correctly implemented and bug free (Gilbert, 2008). Model *calibration* involves tuning a model's rules or parameters to produce results that match real data or stylized facts (i.e., simplified representations of empirical findings) with reasonable accuracy (Carley, 1996). Model *validation* involves comparing model predictions to a holdout sample of data that was not used in the calibration process to see how well the two match (Gilbert, 2008). Verification ensures internal validity or the degree to which the implemented model corresponds with the conceptual model and calibration and validation ensure external validity or the degree to which the model corresponds with the real world (Taber & Timpone, 1996). All three processes can be time consuming so researchers must budget sufficient time when planning the project. No matter how carefully one has worked to develop a model, it will always contain bugs. Some bugs are obvious and easy to find because they prevent the model from running or they generate anomalous results. Other bugs are harder to find, because the model runs and produces results that appear superficially plausible. These bugs require more careful scrutiny and rigorous testing. Occasionally they survive the verification process and get caught in calibration or validation when researchers have difficulty producing results that match theoretical predictions or empirical data (although sometimes it is the theory that needs to be modified).

Once you are confident about a model's internal validity, you can move on to assess its external validity with calibration and validation techniques. The two processes are often confused as one. While both processes examine whether model output matches real world data, calibration involves the "tweaking" of the model iteratively so that its output matches (some of the) data. Validation involves running the model to assess its match to a new sample of data. To avoid overfitting, a good practice is to split data into two sets: one set used to calibrate the model and the other used to validate the model (similar to training and testing sets in machine learning).

For calibration and validation, researchers often focus on assessing outcome validity by comparing model predictions with real-world data or with the predictions of other competing models[1] (Taber & Timpone, 1996). The primary criterion is to show that the model can replicate the system-level regularities that the research seeks to explain (Gilbert, 2008). The replication can be assessed using multiple criteria such as correlations, analysis of variance, linear or nonlinear regression, or tests for comparison of means (Taber & Timpone, 1996). Carley (1996) describes four levels of assessing outcome validity: *pattern validity* requires the pattern of simulation results matches patterns of real data, *point validity* requires the output variables of the model, taken one at a time have the same mean as the real data, *distributional validity* requires the distribution of model output has the same distributional characteristics as the real data, and *value validity* requires the highest level of precision in matching, that is, the model output matches the real data on a point-by-point basis. Which level to choose is at the discretion of researchers depending on research purpose.

Step 6: Experiment with the Model

Once a model is validated, it can be used to run virtual experiments to generate simulated data for which no real data yet exist. This is the step where HCI researchers and practitioners will experience the value of agent-based modeling. They can vary parameters across a wide range and at great granularity—much beyond the level of control typical of field studies or laboratory experiments. Once the model is built, the costs of running a virtual experiment are minimal. More importantly, researchers can open the proverbial "black box" by observing and analyzing intermediate variables to reveal the mechanisms or processes that cause the resulting patterns.

[1]Another rare form of model validation is called model alignment or "docking" in short, under which researchers compare two or more models to see if they can produce the same results. A good example is Axtell and colleagues' work (1996) to align the cultural transmission model and the Sugarscape model. They call for wider practice of docking among modelers.

Similar to laboratory experiments, a virtual experiment generates data for each cell in an experimental design by running the model with a combination of parameters. Meaningful results require careful setting of parameters to match reality and to determine how many data points to simulate for each condition. A good practice is to use theory or empirical evidence to restrict the range of parameters, either qualitatively or quantitatively. Another approach is to sample the parameter space to cover a reasonable range (Gilbert, 2008). For instance, Nan (2011) built an agent-based model to simulate IT use in organizations and used data from a case study by Orlikowski (1996) to set initial conditions. It is also important to include counterfactual analysis or what is often referred to as "what-if" experiments to explore what might happen if parameters are to set to values different from existing empirical observations.

After the experiments, researchers should run sensitivity analysis or robustness checks (Davis et al., 2007) to assess how sensitive simulation results are to key assumptions and parameters built in the model. Sensitivity analysis is the process of relaxing assumptions or systematically changing functions and parameters to see how robust simulation results are or to understand the conditions under which the model yields the results (Gilbert, 2008). Researchers can be more confident about the results if they remain stable when key constraints are relaxed or key parameters are varied. Sensitivity analysis can also be used to facilitate model validation. This practice is especially valuable when little theory or empirical evidence is available to inform the specification of experimental parameters. One recommendation is to expand the parameter space to identify and report "boundary conditions" when simulation results no longer hold.

Step 7: Publish the Model and Results

Analogously to conducting a usability study in industry, if your only purpose of building an agent-based model to inform system design, you are done. However, if you are an academic working on peer-reviewed publications, more work still remains. Because many reviewers are unfamiliar with the method and because the details of a model are harder to describe than the details of an empirical study, it can be difficult to publish research using agent-based models. In this section, we share some of our experiences of reviewing and publishing simulation work.

Lesson 1—Write in plain English and provide enough detail about the model. This advice is easier said than done. Good writing is important for publishing all papers but especially crucial for simulation work because you must appeal to both domain and methodology experts and readers vary greatly in their familiarity with the method. Even a moderately complex model, like the one we describe below, might include dozens of rules, close to 100 variables and 1,300 lines of code built on a platform. You need to provide enough detail about how the model works, without making every reader read the original program. Some common mistakes are failing to include all rules that determine agent behaviors (e.g., saying an agent's opinion is influenced by close neighbors without specifying how the influence occurs), failing to specify the order in which behaviors occur (e.g., do agents express their opinions first and then get influenced or decide to switch groups or vice versa), or failing to clearly describe the initial conditions of virtual experiments (e.g., how many agents to begin with, the rate at which new agents enter, the number of runs for each condition). A rule of thumb to assess whether sufficient detail has been provided is that experienced modelers should be able to draft the pseudo code of the program based on the model description. If space permits, it is also a good idea to include pseudo-code, key functional forms, and a flow chart showing the sequence of actions.

Lesson 2—Prepare to share your code. Whether you are building on a platform or programming from scratch, write clear code with good documentation so that it can be easily read and understood by an average programmer. Some reviewers may request to see your code and other researchers may be interested in confirming or extending your model. There are multiple ways of sharing one's model, privately or publicly. One advantage of building on a platform is the ease of sharing models. For example, NetLogo hosts a Modeling Commons for its users to share models and search for others' models (http://modelingcommons.org/account/login). You can access our online community model at http://dl.dropbox.com/u/11116596/OnlineCommDesign.nlogo.

Lesson 3—Be mindful of sample size when reporting simulation results. Sample size is determined by the number of runs for each experimental condition. Because it is so easy to replicate an experiment once a model is developed, reporting statistical significance is insufficient. A reviewer comment we once received vividly illustrates the concern: "Could you have simulated 1,000 groups and got everything to be significant? How did you choose [the number we had chosen in the paper]?" Our advice is to report effect sizes (e.g., % increase in adoption rates or number of visitors for a day) in addition to statistical significance.

Following the Roadmap: Using ABM to Inform the Design of Online Communities

In this section, we show how we have followed the seven steps and built a model to inform the design of online communities. We began with the research question of how design choices such as topical breadth, message volume, and discussion moderation interact to influence the success of an online community. We believed agent-based modeling was appropriate to address the question. Online communities are bottom-up social structures whose success depends on the active participation and interaction of individual members. Members are heterogeneous in their attributes (e.g., interests, knowledge, experiences with the community) and motivations (e.g., seeking information, emotional support, reputation, entertainment, a sense of belonging) (Ridings & Gefen, 2004; Wasko & Faraj, 2005). When we were starting the project in 2006, there was a good body of knowledge from survey and interviews about what motivates users to participate in online communities (Bryant, Forte & Bruckman, 2005; Ridings & Gefen, 2004). We were also able to ground the model on well-established theories from economics and social psychology around individual decisions to join groups, participate in collective actions, and the influence of perceived benefits and costs. Agent-based modeling was a useful tool to integrate these theories to understand challenges of building successful online communities.

Based on our own empirical research and the literature, we decided the core concept to simulate was individuals' motivation to participate. We chose the Expectancy Theory of motivation (Vroom, Porter & Lawler, 2005) and one of its extensions, the Collective Effort Model (Karau & Williams, 1993) as our basis. These theories assume people contribute to a group to the extent they believe their efforts will lead to outcomes for themselves that they value. Neither theory, however, is specific about the types of benefits that motivate people. Research on online communities had identified six benefits that consistently drove participation: (1) information, (2) fulfillment of altruistic or expressive needs produced by helping others, (3) identification with the group, (4) relationships formed with group members, (5) entertainment, fun, and other forms of intrinsic motivation, and (6) reputation and other forms of extrinsic motivation (e.g., Ren, Kraut & Kiesler, 2007; Ridings & Gefen, 2004; Roberts, Hann & Slaughter, 2006; Wasko & Faraj, 2005).

We supplemented the Expectancy Theory with other theories to operationalize the six benefits. We drew insights from theories of group identity theory (Hogg, 1996) and interpersonal bonds (Berscheid, 1994) to calculate social benefits, and we drew insights from resource-based theory (Butler, 2001) and information overload theory (Jones, Ravid & Rafaeli, 2004) to calculate informational benefits. Theories of group identity and interpersonal bonds propose that members commit and contribute to a group if they feel psychologically attached to the group or its members (Prentice, Miller & Lightdale, 1994). Information overload theory proposes that human beings' information processing capacity is limited and too much information or irrelevant information is aversive (Rogers & Agarwala-Rogers, 1975).

This is where the value of agent-based modeling's ability to combine multiple theories becomes apparent. First, motivation has multiple causes, and each cause is typically treated by a separate social science theory. For example, information overload theory focuses on how informational benefits affect motivation while group identity theory focuses on the motivational influence of psychological attachment to the community. Therefore, multiple theories are needed to model motivation. Second, a single design choice, when routed through different theoretical lenses, can have divergent effects on motivation. One example is the effect of group size. When examined through resource-based theory of online social groups, large group size is a measure of resource availability and thus provides informational benefits. When examined through the Collective Effort Model (Karau & Williams, 1993), however, members of large groups tend to contribute less time and resources because of dilution of responsibility. When examined through the lens of interpersonal bonds (Frank & Anderson, 1971), large group size reduces motivation because it makes it difficult to form relationships with other members. Combining these effects

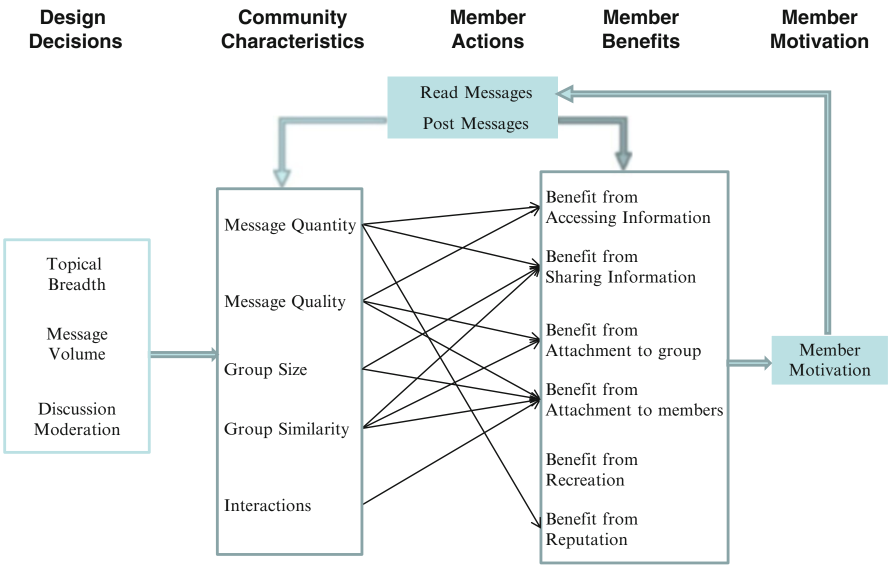

Fig. 2 The conceptual model

in an agent-based model allows us to better understand how potential design decisions affect member motivation and contribution through multiple routes.

In terms of boundary conditions, we decided to focus on members, their interactions, and how various design choices affect their experiences. Although theories from organizational ecology (Hannan & Freeman, 1989), which focus on intercommunity competition, could have been relevant, we ignored these to limit the model's scope and to make model development tractable. We also excluded, for example, the cost of implementing the design choices primarily because our goal is to assess the effectiveness of various designs and partially because it is complicated to model costs (e.g., due to different design contexts).

Figure 2 depicts our conceptual model. Member actions such as reading and posting messages are determined by benefits and costs associated with participation. Reading and posting behaviors change community dynamics such as the number and quality of messages, as well as the number of members and their relationships with one another; these, in turn, influence experienced benefits and motivation. Design interventions, such as the cost of posting messages, diversity of nominal topics, and moderation also influence community dynamics.

We then translated the conceptual model into agents' attributes and behavior rules. The two behaviors that agents engage in are reading and posting messages. Following the utility-like logic underlying the expectancy-value theories, we assumed an agent (1) logs in to read messages when expected benefit from participation exceeds expected cost, and (2) posts messages when expected benefit from contributing exceeds expected cost. Details about how we calculated member benefits can be found in Ren and Kraut (In press). For example, the model assumed that social, identity-based benefit is a function of the extent to which agents' interests are similar to the group's interests, and social, bond-based benefit is a function of the number of other agents with whom the agent has had repeated interactions. In this model, agents take actions during a simulated day, and we simulated time as forced parallel. All active agents in the simulated community are given the opportunity to make a reading and posting decision before anyone moves to the next day. Messages posted the previous day are distributed to all agents the next day and used to update their expectations of benefits.

For a simulated day, agents could make up to three decisions. They first decide how many messages to read. We calculated messages an agent viewed on a specific day as proportional to the amount of benefit he received in the past from reading messages minus the cost of reading, capped by the total number of messages available to read. They next decide whether to post messages, which incurs greater costs than reading messages. If an agent decides to post a message, he makes three additional decisions: (1) whether to start a new thread or reply to an existing post; (2) the topic of the message and (3) which message to reply to. Based on empirical evidence from Usenet groups, we assumed an agent is equally likely to start a new thread or to reply to one. The topic of the message is a joint function of the agent's interests, topics of the messages the agent has recently viewed, and the topic of the replied-to message if it is a reply. Theory and empirical evidence (Fisher, Smith & Welser, 2006; Faraj & Johnson, 2011) suggest three common patterns of interaction among community members: (1) preferential attachment, in which members respond to popular messages or posters; (2) reciprocity, in which members respond to others who have written to them in the past; and (3) interest matching, in which members respond to messages that match their interests. We thus assumed that agents in the model choose to reply to a message based on the average of (1) the number of replies the message has received; (2) the number of times the poster of the message has responded to the agent; and (3) the match between message topic and the agent's interests.

We first built our model in NetLogo, a cross-platform multi-agent modeling environment (Wilensky, 1999). Figure 3 shows a snapshot of the user interface. The buttons in the upper-left corner allow researchers to specify the initial members, messages, type of the community and run time. The window on the right shows members in the community. The plots track statistics such as member entry, exit, and the number of participants and contributors. It took us a year and half to design, build, and validate the model. The online communities we simulated grew to have thousands of members including both lurkers and active contributors and thousands of messages. We later re-implemented it using Repast to achieve greater speed. A virtual experiment with 540 runs that used to take three days to run in NetLogo took several hours in Repast.

We went through all three steps to ensure the validity of our model. Previous studies show that three statistics describing online communities—posts per member, replies per post, and communication partners (out-degrees) per member—demonstrate a power-law distribution (Fisher et al., 2006; Smith, 1999). We used these

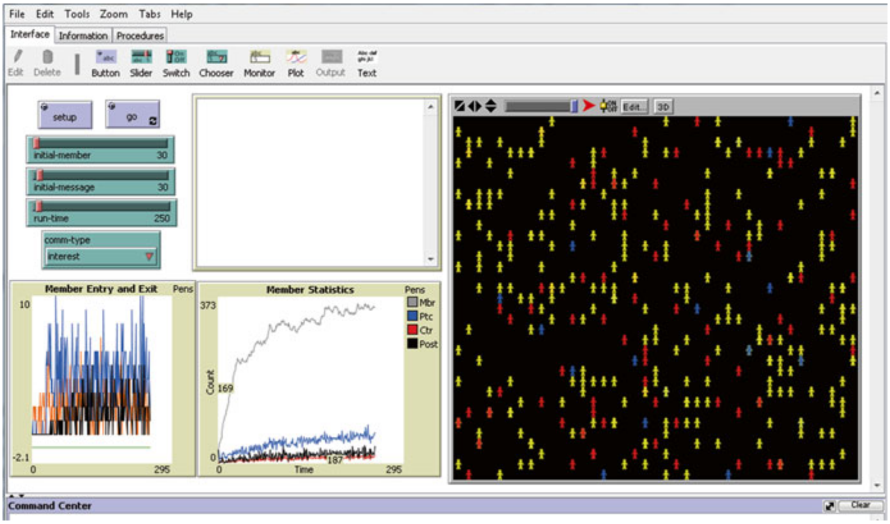

Fig. 3 Interface of the online community model in NetLogo

three stylized facts to calibrate the model. We constructed two data sets of *Usenet* groups and used a set of 12 groups to calibrate the model and a new set of 25 groups to validate it.

Calibration was an iterative process. After each run, we examined mismatches between the simulated and the real data, reexamined model assumptions, and made adjustments to the model in light of theoretical reasoning, empirical evidence, or knowledge about how the processes in the model operate. After ten iterations of tweaking, the model replicated the power-law distribution for all three statistics. We then simulated a new set of 25 Usenet groups. We used pattern validation and compared the pattern of three statistics from the model—posts per agent, replies per post, and out-degree ties per agent—with the pattern generated from real data. We also calculated Pearson correlations between the empirical data series and the simulated ones and the coefficients ranged between 0.90 and 0.96, confirming a good match between the two.

We used the model to explore three design decisions: How broad a set of discussion topics should the community encourage? What is an optimal level of message volume? What type of discussion moderation if any should the community adopt? We designed a full-factorial experiment to simulate three levels of topical breadth with one, five or nine topics, three levels of message volume, with an average of 10, 15, or 20 messages per day, and three types of moderation: no moderation, community-level moderation (under which off-topic messages are removed), and personalized moderation (under which a personalization algorithm presents a subset of messages that match a member's interests). We ran a 365-day simulation for each experimental condition on five randomly constructed groups. All groups began with 30 agents and 30 messages and evolved over time as newcomers joined and old-timers left.

We examined the effects of topical breadth, message volume, and moderation on two outcomes easily visible to a community manager: the number of new posts per day, an indicator of community activity, and the average number of login sessions per member, an indicator of member commitment. We ran analysis of variance (ANOVA) to examine the effects of topical breadth, message volume, and moderation. We also examined the benefits members received on the 100, 150, 200, 250, and 300th day of the experiment.

The model led to several plausible yet non-obvious findings: (1) members of topically broad communities were more committed or visited more frequently than members of topically narrow communities, although they did not post more messages, (2) community-level moderation led to greater commitment but not contribution, and (3) personalized moderation outperformed community-level moderation in communities with broad topical focus and high message volume. These results can be partially explained as a critical trade-off between informational and relational benefits, which the simulation revealed. For example, having more topics to discuss, on the one hand, increases informational benefits because it increases the number of messages likely to match one's interest; on the other hand, it reduces relational benefit because it reduces the chance of two members sharing a common interest.

To assure the robustness of our results, we ran a series of sensitivity analyses by relaxing key assumptions and varying key parameters. Results did not differ substantially. Some of the key parameters we varied were: the likelihood of posting a new message in a day (from 30 to 70 %), the criterion to be recognized as an active contributor (from top being in the top 5 % to the top 20 %), and the accuracy of personalization (from 60 to 100 %).

In terms of design implications, the simulation results call for reconsideration of well-established beliefs in the effectiveness of a narrow focus (Maloney-Krichmar & Preece, 2005) and community-level moderation (Preece, 2000). While these practices remain useful for some communities, our research suggests a contingency view of online community design. There is no universally optimal design for all online communities. The optimal choice depends on community characteristics (topical breadth and message volume) and the specific goals designers wish to accomplish (to make members loyal or to increase their contribution).

How We Discovered Agent-Based Modeling and Useful References

We were asked to also talk about our personal stories with agent-based modeling. For the first author, it could be traced to a belief she had since childhood that we could "simulate" (although at that time she did not know the word) and predict human society as accurately as we could predict the physical world. Serendipity also played a role when she started graduate school at Carnegie Mellon University working as a research assistant for Kathleen Carley, who is an expert in computational social and organizational theory. Later, she began conducting field studies and experiments because it was considered risky to do just simulation research (which may still be true in some disciplines). Years later, however, she enjoys and benefits greatly from being able to study a phenomenon using multiple methods including agent-based modeling.

The second author had been intrigued by the methodology for a long time and offered the first author a postdoctoral position, which started our years of collaboration to study online communities using various methods including agent-based modeling. So if you are foreign to the method, teaming up with someone who has done it can help you climb the learning curve. We should note that like other research skills, agent-based modeling is easy to learn but hard to do well. Experiences help and familiarity with the domain which you study helps as well. In addition, Axelrod (2005) has a book chapter in which he shares his experience of building the model to study the Prisoner's Dilemma game and his success as well as struggles of working with researchers from other disciplines and publishing his interdisciplinary work. It is a fun read.

Here is a list of papers that we have found useful and recommend as additional references:

- Schelling, T. C. (1969). Models of segregation. *American Economic Review*, 59(2), 488–493.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4), 25–34.
- Harrison, J. R., & Carrol, G. R. (1991). Keeping the faith: A model of cultural transmission in formal organizations. *Administrative Science Quarterly*, 36(4), 552–582.
- Carley, K. M. (1991). A theory of group stability. *American Sociological Review*, 56(3), 331–354.
- March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science*, 2(1), 71–87.
- Carley, K. M. (1992). Organizational learning and personnel turnover. *Organizational Science*, 3(1), 20–46.
- Epstein, J. M., & Axtell, R. L. (1996). *Growing Artificial Societies: Social Science from the Bottom Up*. Boston, MA: MIT Press.
- Axelrod, R. (1997). *The complexity of cooperation: agent based models of competition and collaboration*. Princeton, NJ: Princeton University Press.
- Macy, M. W., & Willer, R. (2002). From factors to actors: computational sociology and agent-based modeling, Annual Review of Sociology, 28:143–166.

Concluding Remarks

To summarize, in this chapter, we presented a roadmap of how to use agent-based modeling to synthesize multiple social science theories to inform the design of multiuser systems, using our model on online community design as an example.

We encourage HCI researchers to consider either building your own agent-based models or adapting existing models as a new way of understanding and addressing challenges in designing multiuser systems. Researchers and designers can collaborate to perform "full-cycle research" (Chatman & Flynn, 2005) by alternating between agent-based modeling and field experiments and using the two to complement one another—the former to combine theories and generate new predictions and the latter to test the redesigns informed by simulation results. Once developed and validated, the agent-based model can be continuously extended to incorporate new theories or study new design choices. It can also serve a test bed to help designers navigate design spaces and choose features that fit their design goals. We also foresee the possibility of building agent-based models as collaboration platforms to allow researchers from different disciplines to collectively tackle formidable design challenges.

Exercises

1. Name some social behavior that might be amenable to agent based modeling, outside of the ones listed in this chapter.
2. Where do the rules come from that determine the agents' behaviors?

References

Axelrod, R. (1986). An evolutionary approach to norms. *American Political Science Review, 80*, 1095–1111.

Axelrod, R. (1997). *The complexity of cooperation: Agent based models of competition and collaboration*. Princeton, NJ: Princeton University Press.

Axelrod, R. (1997a). The dissemination of culture: A model with local convergence and global polarization. *Journal of Conflict Resolution, 41*, 203–226.

Axelrod, R. (1997b). *The complexity of cooperation: Agent-based models of competition and collaboration*. Princeton, NJ: Princeton University Press.

Axelrod, R. (2005). Agent-based modeling as a bridge between disciplines. In K. L. Judd, & L. Tesfatsion (Eds.), *Handbook of Computational Economics, Vol. 2: Agent-Based Computational Economics*. Amsterdam: Elsevier.

Bao, P., Hecht, B., Carton, S., Quaderi, M., Horn, M., & Gergle, D. (2012). Omnipedia: Bridging the Wikipedia language gap. *Proceedings of the ACM Conference on Human-Factors in Computing Systems* (pp. 1075–1084). New York: ACM Press.

Berscheid, E. (1994). Interpersonal relationships. *Annual Review of Psychology, 45*, 79–129.

Bryant, S. L., Forte, A., & Bruckman, A. (2005). Becoming Wikipedian: Transformation of participation in a collaborative online encyclopedia. *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work* (pp. 1–10). New York: ACM.

Burton, R. M., & Obel, B. (1995). The validity of computational models in organization science: From model realism to purpose of the model. *Computational and Mathematical Organization Theory, 1*(1), 57–71.

Butler, B. S. (2001). Membership size, communication activity, and sustainability: A resource-based model of online social structures. *Information Systems Research, 12*(4), 346–362.

Carley, K. M. (1991). A theory of group stability. *American Sociological Review, 56*(3), 331–354.

Carley, K. M. (1992). Organizational learning and personnel turnover. *Organizational Science, 3*(1), 20–46.

Carley, K.M. (1996). *Validating computational models*. Working paper, Pittsburgh, PA.

Chatman, J. A., & Flynn, F. J. (2005). Full-cycle micro-organizational behavior research. *Organization Science, 16*(4), 434–447.

Davis, J., Eisenhardt, K. M., & Bingham, C. B. (2007). Developing theory through simulation methods. *Academy of Management Review, 32*(2), 480–499.

DiMicco, J., Millen, D. R., Geyer, W., Dugan, C., Brownholtz, B., & Muller, M. (2008). Motivations for social networking at work. *Proceedings of the ACM Conference on Human-Factors in Computing Systems* (pp. 711–720). New York: ACM Press.

Drenner, S., Sen, S., & Terveen, L. (2008). Crafting the initial user experience to achieve community goals. *Proceedings of the 2008 ACM Conference on Recommender Systems. (*pp. 187–194). New York, NY: ACM

Epstein, J. M., & Axtell, R. L. (1996). *Growing artificial societies: Social science from the bottom up*. Boston, MA: MIT Press.

Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity, 4*(5), 41–60.

Faraj, S., & Johnson, S. L. (2011). Network exchange patterns in online communities. *Organization Science, 22*(6), 1464–1480.

Festinger, L. (1954). *A theory of social comparison processes (Vol. 7)*. Indianapolis, IN: Bobbs-Merrill.

Fisher, D., Smith, M., & Welser, H.T. (2006). *You are who you talk to: Detecting roles in Usenet newsgroups*: *Proceedings of the 39th Hawaii International Conference on System Sciences in Waikoloa, Big Island, Hawaii*.

Fogarty, J., Lai, J., & Christensen, J. (2004). Presence versus availability: The design and evaluation of a context-aware communication client. *International Journal Human-Computer Studies, 61*(3), 299–317.

Frank, F., & Anderson, L. R. (1971). Effects of task and group size upon group productivity and member satisfaction. *Sociometry, 34*(1), 135–149.

Gardner, M. (1970). Mathematical games: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American, 223*, 120–123.

Gilbert, N. (2008). Agent-based models. In T. F. Liao (Ed.), *Quantitative applications in the social sciences*. Los Angeles, CA: Sage.

Grudin, J. (1994). Groupware and social dynamics: Eight challenges for developers. *Communications of ACM, 37*(1), 92–105.

Hannan, M. T., & Freeman, J. (1989). *Organizational ecology*. Cambridge, MA: Harvard University Press.

Harper, F.M., Frankowski, D., Drenner, S., Ren, Y., Kiesler, S., Terveen, L., Kraut, R. E., & Riedl, J. T. (2007). Talk amongst yourselves: Inviting users to participate in online conversations. *Proceedings of the 12th International Conference on Intelligent User Interfaces* (pp. 62–71). Honolulu, Hawaii.

Harrison, J. R., & Carroll, G. R. (1991). Keeping the faith: A model of cultural transmission in formal organizations. *Administrative Science Quarterly, 36*(4), 552–582.

Harrison, J. R., Lin, Z., Carroll, G. R., & Carley, K. M. (2007). Simulation modeling in organizational and management research. *Academy of Management Review, 32*(4), 1229–1245.

Hogg, M. A. (1996). Social identity, self-categorization, and the small group. In E. H. Witte & J. H. Davis (Eds.), *Small group processes and interpersonal relations* (2nd ed., pp. 227–253). Hillsdale, NJ: Lawrence Erlbaum Associates.

Jones, Q., Ravid, G., & Rafaeli, S. (2004). Information overload and the message dynamics of online interaction spaces: A theoretical model and empirical exploration. *Information Systems Research, 15*(2), 194–210.