AGENT-BASED MODELING AND SIMULATION OF COLLABORATIVE SOCIAL NETWORKS

Greg Madey
Yongqin Gao
Computer Science
University of Notre Dame
gmadey@nd.edu
ygao1@nd.edu

Vincent Freeh
Computer Science
North Carolina State University
vin@cs.ncsu.edu

Renee Tynan
Chris Hoffman
Department of Management
University of Notre Dame
rtynan@nd.edu
choffman@nd.edu

Abstract

We describe a research framework for studying social systems. The framework uses agent-based modeling and simulation as key components in the process of discovery and understanding. A collaborative social network composed of open source software (OSS) developers and projects is studied and used to demonstrate the research framework. By continuously collecting developer and project data for over two years from SourceForge, we are able to infer and model the structural and the dynamic mechanisms that govern the topology and evolution of this social network. We describe the use of these empirically derived agent-based models of the SourceForge OSS developer network to specify simulations implemented using Java/Swarm. Several network models and simulations of the evolution of SourceForge, and the verification and validation processes of the framework are described. The nature of social network processes hidden from view that could plausibly generate the observed system properties can be discover through an iterative modeling, simulation, and validation and verification process. Such a process, dynamic fitness based on project life cycle, was discovered.

Key words: Agent-based modeling, collaborative social networks, open source software, simulation, computational social science, social networks, Java/Swarm simulation

Introduction

In this paper we explore how agent-based modeling and simulation can be used as a research technique to study collaborative social networks. In particular the collaborative social network we studied is composed of global virtual self-organizing teams of software developers working on open source software projects. Open source software conforms to a two-part definition that stipulates: 1) the software is distributed openly and freely, and most importantly, 2) the software's source code is open to viewing and modification. Both characteristics of OSS run contrary to the traditions of proprietary software licensing. Vendors of proprietary closed source software typically issue end user licensing agreements (EULAs) that prohibit such free and open activities.

Free distribution of OSS means its developers typically receive no monetary compensation for their efforts. Yet, the OSS movement has produced widely recognized successes even in the absence of profit motive. Apache, Linux, GNU, MySQL and many other crucial Internet infrastructure software components are open source and often number one in market share. This is displayed in Apache's case with 62% market share according to a Netcraft.com survey (Netcraft.com 2003).

Some researchers tout the process of OSS development as intrinsically superior to closed proprietary development. They argue that OSS development generates more stable code, more quickly. These assertions, however, have yet to be proven.

Nonetheless, a process beneficial to so many economic agents, but devoid of a primary economic motivator deserves attention. Proprietary software developers may adapt and adopt all or some aspects of OSS development should the process emerge truly superior. Governments may discover sufficient interest to subsidize the OSS movement; for example, the German government is supporting development and acquisition of open source software as an alternative to closed source software (Shankland 2003). However, understanding the promise of OSS development requires better understanding of the OSS movement itself, which is precisely the aim of this study.

Social Network Theory

Social network theory is a conceptual framework through which we view the OSS developer movement. The theory, built on mathematical graph theory, depicts interrelated social agents as nodes or vertices of a graph and their relationships as links or edges drawn between the nodes (Wasserman 1994). Let the circles labeled 'A,' 'B,' and 'C' in Figure 1 represent three developers in a highly simplified collaborative social network. Each is a node or vertex of a graph. The edges connecting the vertices could represent the relationship of joint OSS project membership. Thus, Figure 1 displays a graph of a social network that describes a three-person project. In Figure 2, 'C' and 'D' form a second project and we may consider 'D' connected to 'A' and 'B' through their shared adjacent vertex 'C.' Such a vertex 'C' plays an important role in social networks and is sometimes called a hub or linch-pin node. The number of edges connected to a vertex is called its index and in Figure 2, vertices 'A' and 'B' have an index of 2, and 'D' and 'C' have index values of 1 and 3, respectively.

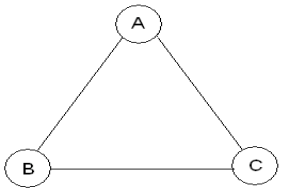

Figure 1. Simple Collaborative Network

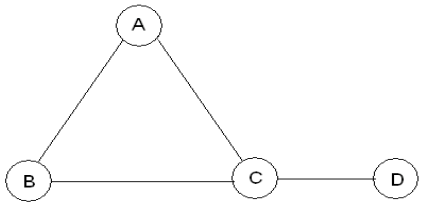

Figure 2. Collaborative Network Of Two Projects: {A, B, C} and {C, D}

Social networks of any serious interest are more complex, encompassing tens of thousands of agents (or more) with many possible topological and structural patterns. Some of these topological or structural characteristics include 1) the minimum, maximum, average, and distribution of index values over the nodes, 2) local clustering or connectedness of nodes, measured by a quantity called the clustering coefficient, and often described as the size and connectedness of a node's "circle of friends", 3) "weak ties" between local clusters created by linchpin nodes (such as 'C' in Figure 2), 4) the diameter of the network, and 5) the small world phenomenon. Even as network populations grow exceedingly large, the shortest path between any two agents remains relatively small. The literature has termed this property the Small World Phenomenon or the Kevin Bacon effect (Watts 1999; Watts 2003).

Of special interest are the evolutionary processes and associated topological formation in dynamic growing networks. Early work in this field by Erdos and Renyi focused on random graphs, i.e., those where edges between vertices were attached in a random process (called ER graphs here) (Bollobas 2001). In ER graphs, the range of minimum to maximum index values is small, and the statistical distribution of index values are Poisson for smaller graphs, and approximately Gaussian as the number of nodes increases. These distributions of index values for the random graphs do not agree with the observed power law distribution for many social networks, including the OSS developer network reported for SourceForge (Madey 2002a; Madey 2002b; Madey 2002c). The ER graphs also fail to fit observed characteristics of real social networks, including the local clustering, the existence of linchpin nodes, and the small world property.

Other evolutionary processes, other than random attachment of edges, must then be in play to produce the observed properties of many collaborative social networks. Some proposed mechanisms are characteristic of more recently described models including 1) the Watts-Stogatz (WS) model (Watts 1998), 2) the Barabasi-Albert (BA) model with preferential attachment (Albert 1999; Barabasi 1999; Barabasi 2000), 3) the modified BA model with fitness (Barabasi 2002; Barabasi 2001), and 4) an extension of the BA model (with fitness) to include dynamic fitness based on project life cycle discovered on this project and described in this paper. The WS model captures the local clustering property of social networks and was

extended to include some random reattachment to capture the small world property, but failed to display the power-law distribution of index values. The BA model added preferential attachment, preserving the realistic properties of the WS model, also displaying the power-law distribution. The BA model was extended with the addition of random fitness to capture the fact that sometimes newly added nodes grow edges faster than previously added nodes (the "young upstart" phenomenon).

Self-organization is another property of many social networks. A self-organizing network grows and develops structure without centralized decision making. Instead, macroscopic properties evolve from the many interactions of individual agents. From a vast number of local decisions, global properties emerge. For example, in a free market economy, aggregate demand and supply are macroscopic properties that we see emerge in a self-organized fashion from the independent purchasing decisions of consumers and producers. We also observe this property in the OSS collaborative social network.

Modeling and Simulation

Prior research suggests that the OSS phenomenon can be considered a complex, self-organizing system (Axelrod 1999; Barabasi 2002; Barabasi 2000; Faloutsos 1999; Holland 1998; Huberman 1999; Johnson 2001; Kuwabara 2000).These systems are typically comprised of large numbers of locally interacting elements. Although the rules describing those local interactions may be few and simple, often unexpected and difficult to predict global properties emerge. Many investigators of such systems have found that they can only be understood through modeling, and specifically through what some researchers call iconological modeling and structural modeling (Eve 1997; Harvey 1997; Kiel 1997; Smith 1997). The goals of these simulation approaches can be achieved using the agent-based approach pioneered by Schelling (1978), and advanced by Axelrod (1984), Epstein and Axtell (Epstein 1996), Resnick (1994), Cohen et al. (Cohen 1998), and many others. Using such models enables social science researchers to address the modeling difficulties of this field; social processes are complex, they have sensitive dependence on behaviors of individual heterogeneous agents (which are not omniscient rational utility optimizers), and they cannot always be modeled as systems assumed to be in static equilibrium (Axelrod 1997b; Byrne 1997; Epstein 1996; Gaylord 1998; Goldspink 2002). The goal of these models is not to predict, but to develop an understanding of how and why the elements of the system are able to produce emergent behavior, and what invariant properties such as boundedness, periodicity, or chaotic attractors are present (Axelrod 1997a; Axelrod 1997b; Harvey 1997; Holland 1998). This understanding can be obtained by discovering the rules and mechanisms that control agent interactions. In the simulations of OSS processes, the agents could be the developers, the projects, and clusters of projects.

Data Collection

The empirically collected data examines over 24 months of snapshot data of developers and projects statistics from SourceForge between January 2001 and May 2003. SourceForge.net is an online OSS project support site sponsored by VA Software (SourceForge 2003). While not the exclusive repository for open source software, SourceForge does host projects for a large enough population of OSS developers to be considered representative of the community as a whole, with the possible exception of the very largest projects which maintain their own online support sites. At the time of this writing, SourceForge hosts more than 80,000 developers and their 50,000 projects (SourceForge 2003).

The empirical data that is collected from SourceForge consists of a two-field table. The first field contains a developer's identification number, unique to each developer; the second contains a similar number attributed to a single project. Data is stored and analyzed in a relational database system. The most recent data collection (May 2003) consisted of 120,000 records. Our entire research database consists of a combined total in excess of 2 million records.

A developer's degree is the number of projects to which he or she contributes. For a project, degree refers to the number of participating developers. A power law was discovered for each distribution (Madey 2002a; Madey 2002b). Figure 3 shows an example of such a power law for project degree distribution from the SourceForge data.

Apart from distributions, we also extracted time series data on the growth of projects and participating developers. Linear regression of each trend produced growth rate constants. These and other empirical data are used to parameterize the simulation and for the verification and validation of hypothesized models and their simulation. Figure 4 below shows an example of output from one time series trend analysis.

Simulation Design

Accurately simulating OSS development as a self organizing collaborative social network relies on agent-based modeling (Terna 1998). In an agent-based model, each individual data object, or agent, contains its decision logic. Agents behave according to this logic alone. No central authority commands or controls them. Thus, agents self-organize; agent-based modeling can simulate self-organizing systems.

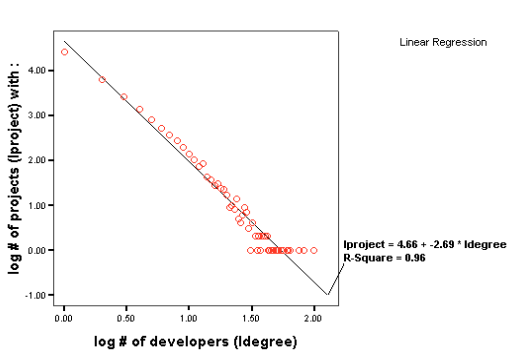

Figure 3. Project Degree Distribution

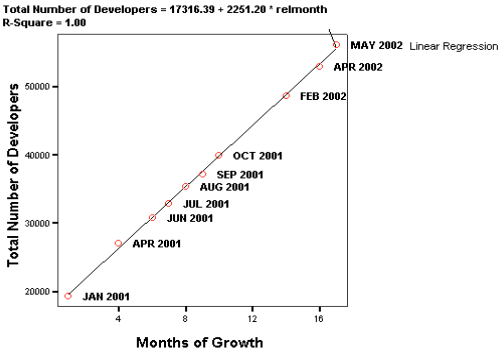

Figure 4. Sample Trend Analysis

In our model, open source software developers are the agents. Each is an instance of a Java class with methods that encapsulate a real developer's possible daily interactions with the development network. Developers can create, join, or abandon a project each day or continue their current collaborations. A separate Java method models each of the first three possibilities. A fourth method encapsulates a developer's selection of one of the three alternatives. Here, three model parameters appear. Each represents the probability of one of the three developer activities. Comparison of a randomly generated number to these probabilities determines which behavioral method the agent will enact. The trend analysis discussed in the previous section provides a benchmark by which to judge calibration of these parameters and determine future adjustments. Table 1 below describes parameters in the current program version.

Table 1. Parameters that influence agent behavior

Parameter Name	Description of Value
probCreateInitially	Probability that a developer creates a project during his first time slice
probCreateEachPeriod	Probability that developer creates a project during any time slice after his first
probJoinEachPeriod	Probability that developer joins a project during any time slice after his first
probAbandonEachPeriod	Probability that developer abandons a project during any time slice after his first
developersPerPeriod	The number of developers introduced to the network each period
endTime	The duration of the simulation in time slices (days)

The agents' virtual network of collaborations is stored in a relational database. The database is designed to catalogue all collaborations in the simulated network. It consists of three tables, "DEVELOPERS," "PROJECTS," and "LINKS." A record in each represents a developer, a project, or a developer's commitment to a project, respectively. Together, the primary keys from "DEVELOPERS" and "PROJECTS" form a composite primary key for entries in "LINKS." Thus, "LINKS," which archives all simulated collaborations, is identical to our table of SourceForge data. Consequently, we can analyze simulation data with the same scripts used on the empirically collected data from SourceForge. This analysis has aided with model parameter calibration and program revision as described in the next section.

Unlike developers, projects and links are not Java agents, since they are passive elements of the social network. Projects and links are not self-motivated. They change only as developer decisions so dictate. Objectification of projects and links,

however, is crucial to the flexibility of our design. As database objects, projects and links may gain attributes through table alterations adding fields. Based on planned surveys of actual OSS developers, we hope to discover very specific factors that affect their collaboration choices. Should we uncover characteristics like charismatic project initiator as attractors for developer contribution, we can update the database schema to include them.

The simulations are built using the Java programming language, the agent-based modeling library Swarm (Minar 1996; Swarm_Development_Group 2000; Terna 1998), and JDBC connections to the database. The Swarm library enables discrete event, multi-agent simulations. Both the Swarm library and the Java language are object-oriented, providing the object-oriented programming benefits of attribute and behavior encapsulation, information hiding, and inheritance (Epstein 1996; Minar 1996; Swarm_Development_Group 2000; Terna 1998). We instantiate developer objects (agents), and store them in a list. The Swarm activity scheduler then traverses that list, prompting each developer to choose a daily activity. Swarm automates model iteration, removing cumbersome looping structures from the program code.

Simulation Iterations

We have simulated random attachment of developers, and as expected in such ER graphs, no power law or small world characteristics were observed in the simulation of the OSS developer collaborative network. Developer agents join a randomly selected project without preference. However, power laws in the distribution of project degree suggest developers may employ preferential attachment: that developers identify projects with certain qualities as more attractive. In aggregate, their choices begin to favor, or prefer, projects with the given qualities, skewing project membership to fit a power law (Schroeder 1991). We next simulated preferential attachment, giving early arrivals a "first–mover" advantage. Projects that enter the simulation tend to have larger index values than new projects. This simulation implements the BA model, and as expected displayed a power-law distribution of developer indices and project indices. Also, as expected, new arrivals unrealistically rarely ever had index values greater than older arrivals. A random fitness parameter was added to the agents and projects thus implementing the BA model with fitness. We observed the independent bipartite nature of our modeled system by observing that fitness for developers and fitness for the projects could be added independently. The simulation thus more accurately described the empirical data, but not completely. In the next iteration, we added dynamic fitness, reflecting the life-cycle of projects: often rapid growth in their youth, stable properties in mid-life, and declining activity and participation as the project matures. This iterative process of using agent-based modeling and simulation is displayed in the research framework of Figure 5 below.

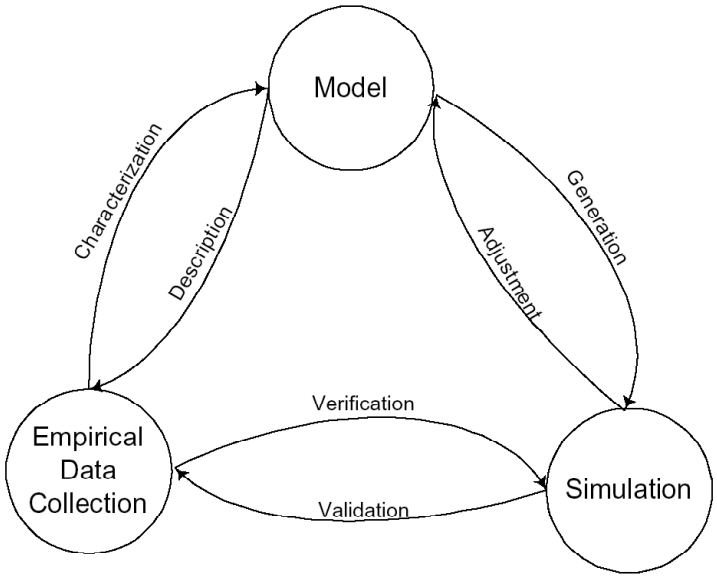

Figure 5. A research framework using data collection, modeling and agent-based simulation to gain understanding of a collaborative social network

In Figure 5, the three nodes represent three components of our research framework. The first component, Empirical Data Collection, consists of data collection and analysis activity. On this study, that included downloading developer and project

data from SourceForge, data cleansing and analysis, and discovering small world and power-law distribution properties in the data. The second component, labeled "Model" in Figure 5, includes the building of a conceptual model of the phenomenon under study. For this paper, that is a dynamic social network, with developer nodes attaching over time under random or preference attachment rules. The third component, the Simulation stage, includes the building of a computer simulation using the model as a specification. On this study, that was the Java/Swarm agent-based simulation of various social networks, calibrated by the empirical data, and implementing hypothesized attachment rules. The six links between the three nodes in the research framework displayed in Figure 5 represent activities of 1) characterization of the model using empirical data, 2) the use of the conceptual model to describe the phenomenon under study and help make sense of the collected data, 3) the use of the model as a specification to help generate simulations, 4) feedback from the simulation activity to help refine and adjust the model, 5) verification of the simulation using known empirical data, and 6) validation of the model by comparing its predictive accuracy against newly collected data.

Summary and Conclusions

We describe an iterative research framework consisting of three inter-linked components: empirical data collection, agent-based modeling, and agent-based simulation. This framework was utilized and evaluated in the study of a large global virtual collaborative social network, the Open Source Software (OSS) developer network. In particular, we collected data from the OSS web site, SourceForge, on over 50,000 projects and 80,000 developers as of May 2003. Analysis of the data, collected monthly over 2 years, suggested that the theory of evolving collaborative social networks could be used as a conceptual modeling framework. Several such evolving collaborative social networks were iteratively modeled and simulated using an agent-based simulation approach employing Java/Swarm. Topological and statistical properties of the simulated collaborated social networks were compared with those of the real-world OSS developer network. Differences and similarities were used to calibrate, refine, support, and revise our understanding of the real-world OSS phenomenon. The distribution of developer index played an important role in this process, in particular the presence or absence of a power law distribution of index values.

We modeled and simulated 1) the ER network, 2) the BA network, and 3) the BA network with fitness, supporting published results on how each would describe the real-work OSS developer network. Our iterative framework helped us discover a limitation of the state-of-the-art model, the "BA network with fitness", suggesting its extension to include dynamic fitness, caused in part by the life-cycle of the project.

This research was funded in part by the NSF Award-0222829, from the Digital Society & Technologies Program, CISE/IIS.

References

Albert, R., Jeong, H., Barabasi, A. L. "Diameter of the World Wide Web," *Nature* (401), September 9, 1999 1999, pp 130-131.

Axelrod, R. *The Evolution of Cooperation* Basic Books, New York, 1984.

Axelrod, R. "Advancing the Art of Simulation in the Social Sciences," *Complexity* (3:2) 1997a, pp 16-22.

Axelrod, R. *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration* Princeton University Press, Princeton, 1997b.

Axelrod, R., M. Cohen *Harnessing Complexity: Organizational Implications of a Scientific Frontier* The Free Press, New York, 1999.

Barabasi, A.-L. *Linked: The New Science of Networks* Perseus, Boston, 2002.

Barabasi, A.L., Albert, R. "Emergence of Scaling in Random Networks," *Science* (286), October 15, 1999 1999, pp 509-512.

Barabasi, A.L., Albert, R., Jeong, H "Scale-free Characteristics of Random Networks: The Topology of the World Wide Web," *Physica A*) 2000, pp 69-77.

Barabasi, A.L., Jeong, H., Neda, Z., Ravasz, E., Schubert, A., Viscek, T. "Evolution of the Social Network of Scientific Collaborations," 2001.

Bollobas, B. *Random Graphs*, (2nd ed.) Academic, New York, 2001.

Byrne, D. "Simulation - A Way Forward," *Sociological Research Online* (2:2) 1997.

Cohen, M.D., Riolo, R.L., Axelrod, R. "The Emergence of Social Organization in the Prisoner's Dilemma: How Context-Preservation and other Factors Promote Cooperation," University of Michigan, http://www.pscs.umich.edu/RESEARCH/pscs-tr.html, Ann Arbor, MI.

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

Eve, R., S. Horsfall, M. Lee (ed.) *Chaos, Complexity and Sociology*. Sage Publications, Thousand Oaks, 1997.

Faloutsos, M., Faloutsos, P., Faloutsos, C. "On Power-Law Relationships of the Internet Topology," SIGCOMM'99, ACM, Cambridge, MA, 1999, pp. 251-262.

Gaylord, R.J., L.J. L'Andria *Simulating Society: A Mathematica Toolkit for Modeling Socioeconomic Behavior* Springer-Verlag (TELOS), New York, 1998.

Goldspink, C. "Methodological Implications of Complex Systems Approaches to Sociality: Simulation as a Foundation for Knowledge," *Journal of Artificial Societies and Social Simulation* (5:1) 2002, pp 1-19.

Harvey, D., M. Reed "Social Science as the Study of Complex Systems," in: *Chaos theory in the Social Sciences: Foundations and Applications,* L.D. Kiel, E. Elliot (ed.), The University of Michigan Press, An Arbor, 1997, pp. 295-323.

Holland, J. *Emergence: From Chaos to Order* Addison-Wesley Publishing Company, Reading, MA, 1998.

Huberman, B.A., Adamic, L. A "Growth Dynamics of the World Wide Web," *Nature* (401), September 9, 1999 1999, p 131.

Johnson, S. *Emergence: The Connected Lives of Ants, Brains, Cities, and Software* Scribner, New York, 2001.

Kiel, L.D., E. Elliot *Chaos Theory in the Social Sciences: Foundations and Applications* The University of Michigan Press, Ann Arbor, 1997.

Kuwabara, K. "Linux: A Bazaar at the Edge of Chaos," *First Monday* (5:3), March 2000 2000, pp 1-68.

Madey, G., Freeh, V., and Tynan, R. "Agent-Based Modeling of Open Source using Swarm," Americas Conference on Information Systems (AMCIS2002), Dallas, TX, 2002a, pp. 1472-1475.

Madey, G., Freeh, V., and Tynan, R. "The Open Source Software Development Phenomenon: An Analysis Based on Social Network Theory," Americas Conference on Information Systems (AMCIS2002), Dallas, TX, 2002b, pp. 1806-1813.

Madey, G., Freeh, V., and Tynan, R. "Understanding OSS as a Self-Organizing Process," The 2nd Workshop on Open Source Software Engineering at the 24th International Conference on Software Engineering (ICSE2002), Orlando, FL, 2002c.

Minar, N., R. Burkhart, C. Langton, M. Askenzi "The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations," http://www.santafe.edu/sfi/publications/96wplist.html, 1996, pp. 1-11.

Netcraft.com "Netcraft Web Server Survey," http://www.netcraft.com/survey/, 2003.

Resnick, M. *Turtles, Termites, and Traffic Jams* The MIT Press, Cambridge, MA, 1994.

Schelling, T. *Micromotives and Macrobehavior* W. W. Norton, New York, 1978.

Schroeder, M.R. *Fractals, Chaos, Power Laws* W. H. Freeman and Company, New York, 1991, p. 429.

Shankland, S. "Munich Breaks with Windows for Linux," c|net News.Com < http://news.com.com/2100-1016-1010740.html?tag=nl>, 2003.

Smith, T. "Nonlinear Dynamics and the Micro-Macro Bridge," in: *Chaos, Complexity and Sociology,* R. Eve, S. Horsfall, M. Lee (ed.), Sage Publications, Thousand Oaks, 1997, pp. 52-78.

SourceForge "http://sourceforge.net/," 2003.

Swarm_Development_Group "Brief Overview of Swarm," http://www.santafe.edu/projects/swarm/swarmdocs/set/book149.html, 2000.

Terna, P. "Simulation Tools for Social Scientists: Building Agent Based Models with SWARM," *Journal of Artificial Societies and Social Simulation* (1:2) 1998, pp 1-12.

Wasserman, S., K. Faust *Social Network Analysis: Methods and Applications* Cambridge University Press, Cambridge, UK, 1994.

Watts, D. *Small Worlds* Princeton University Press, Princeton, 1999, p. 262.

Watts, D. *Six Degrees: The Science of a Connected Age* w. w. Norton & Company, New York, 2003.

Watts, D., Strogatz, S. H. "Collective Dynamics of Small-World Networks," *Nature* (393) 1998, pp 440-442.