Conceptual Framework for Agent-based Modeling and Simulation

Yongqin Gao

Department of Computer Science and Engineering

University of Notre Dame

ygao1@nd.edu

Vince Freeh

Department of Computer Science

North Carolina State University

vin@cse.ncsu.edu

Greg Madey

Department of Computer Science and Engineering

University of Notre Dame

gmadey@nd.edu

Abstract

The open Source Software [6] (OSS) development movement is a classic example of a collaborative social network [4, 5, 6]; it is also a prototype of a complex evolving network [2, 19, 20]. By collecting developer and project information monthly from SourceForge for over two years, we have sufficient data to infer the structural and the dynamic mechanisms that govern the topology and evolution of this complex social network system, using agent based modeling and simulation techniques [9, 10]. In this paper, we present the process of building empirically derived agent-based models of the SourceForge OSS developer network and simulation of this collaborative social network. We accomplish our goal by extracting statistics of the OSS SourceForge network, including snapshots and longitudinal data. Several network models of the evolution of SourceForge, the simulation library used [8], and the verification and validation process are given in this paper. The hidden nature of social network processes that could plausibly generate the observed system properties are discovered from an iterative modeling, simulation, validation and verification process.

We proposed a conceptual framework for agent-based modeling and simulation, as shown in Figure 1. We have three entities in the framework: empirical data collection, model and simulation. The model is the process description that is implemented in simulation, and by which we can reproduce the evolution of empirical data. The simulation is a tool to verify and validate the model. We also have six edges in the framework, each represents one kind of process in the framework. Characterization is a process to abstract the characteristics of the empirical data and to generate the rules and attributes in the model. Description is a process to manifest the underlying mechanisms of the empirical data evolution. Generation is a process to build a simulation based on the given model. Adjustment is a process to modify the model according to the feedback from verification process. Verification is a process testing the simulation's behaviors by comparing the simulation output with the empirical data and the designed model behaviors. Validation is a process of interpolating or extrapolating the simulation output and comparing with the empirical data and its attributes not used to define the model. Validation may add more rules or attributes into the model for prospective improvement. The main goal of this kind of study is to get a "fit" model to describe the evolution of a collaborative network by simulation iterations.

We explain this conceptual framework using a real example – a study of the SourceForge collaboration network [1]. The

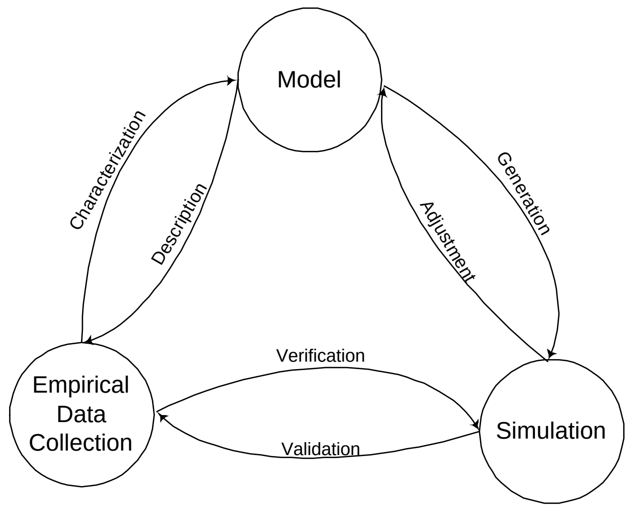

Figure 1: Conceptual framework for agent-based modeling and simulation

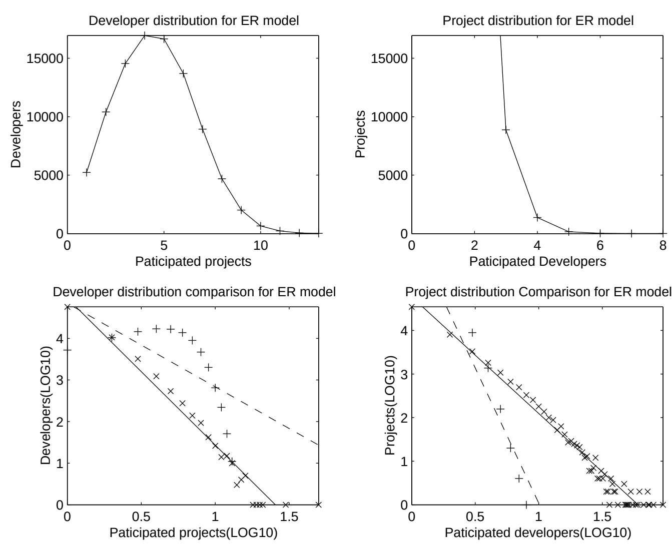

Figure 2: Degree distribution for ER model

empirical data is the collected data we obtained from SourceForge for over two years. We get the simulation parameters from the collection of empirical data, which will be used to characterize the model. So far we generated all three popular complex network models for the SourceForge network using 1) Random Network theory implemented as a ER(Erdös and Rényi) model [12], 2) scale free network implemented as a BA (Barabási and Albert) model [7], and 3) scale free network with fitness using the same simulation parameters. By comparing the simulation results with the real data, we get not only the clues on how to adjust the evolution parameters, but also a validation of the models. In our project, we have simulated random attachment of developers (ER model [12, 19]). We collected all the frequently used statistics [13, 14] from the simulation results. For the clustering coefficient, we get

$$CC \sim N^{-0.14} \tag{1}$$

where CC is clustering coefficient and N is the network size. This value is significantly smaller than we observe in the SourceForge network. And the diameter of the simulated network, which is around 3, is smaller than we observed in the SourceForge network, which is around 6. But the diameter is increasing with the network size. We also get the degree distribution result as shown in Figure 2. As expected in such ER graphs, no power law was observed in the simulation of the OSS developer collaborative network. Developer agents join a randomly selected project. However, power laws in the distribution of project degree suggest that developers may employ preferential attachment, i.e., that developers may 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.

Next we simulated preferential attachment [15], giving early arrivals a "first-mover" advantage. Projects that enter the simulation earlier 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. And the clustering coefficient is around 0.7, higher than the ER model and comparable to the empirical data. The diameter is around 7 and decreasing, which also fits the empirical data. Also, as expected, rarely (and inconsistent with our empirical data) new arrivals ever had index values greater than older arrivals.

A random fitness parameter was added to the agents system by observing that fitness for developers and fitness for the projects could be added independently. The model thus described the empirical data more accurately using statistical attributes of the whole network, but not completely according to the individual development patterns (e.g., life cycle of developer and project). Finally, we implemented our proposal for dynamic fitness. Dynamic fitness, implemented as an individual developer and project life cycles with declining fitness over time, kept all the existing statistical attributes of BA model, and modeled the individual developer and project life-cycle patterns in the empirical data.

References

[1] Y. Gao, V. Freeh and G. Madey, *Analysis and modeling of open source software community*, submitted to NASOS 2003.

[2] L. A. Adamic and B.A. Huberman, *Scaling behavior of the world wide web*, Science, 287(2115), 2000.

[3] Z. Neda, E. Ravasz, A. Schubert and A.L. Barabási, *Evolution of the social network of scientific collaborations*, PhysicA, 311(590), 2002.

[4] R. Albert and A.L. Barabási, *Dynamics of complex systems: scaling laws for the period of boolean networks*, Physics Reviews.

[5] R. Albert and A.L. Barabási, *Emergence of scaling in random networks*, Science, 286:509-512, 1999.

[6] G. Drummond, *Open source software and documents: a literature and online resource review*, 1999.

[7] P. Erdös and A. Rényi, *On random graphs*, Publications mathematicae, 6:290-297, 1959.

[8] D. Hiebeler, *The swarm simulation system and individual-based modeling*, Advanced technology for natural resource management, 1994.

[9] G. Madey, V. Freeh and R. Tynan, *Agent-based modeling of open source using swarm*, 8^{th} Americas Conf. of Information Systems, 2002.

[10] G. Madey, V. Freeh and R. Tynan, *The Open Source Software development phenomenon: an analysis based on social network theory*, 8^{th} Americas Conf. of Information Systems, 2002.

[11] P. J. Kiviat, *Simulation, technology, and the decision process*, ACM Tran. on modeling and computer simulation, 1(2):89, 1991.

[12] D. J. Watts, M.E.J. Newman, *Random graph models of social networks*, Physics reviews, 64(026118), 2001.

[13] M.E.J. Newman, *Scientific collaboration networks: I. network construction and fundamental resules*, Physics reviews, 64(016131), 2001.

[14] M.E.J. Newman, *Scientific collaboration networks: II. shortest paths, weighted networks, and centrality*, Physics reviews, 64(016131), 2001.

[15] M.E.J. Newman, *Clustering and preferential attachment in growing networks*, Physics reviews, 64(025102), 2001.

[16] D. J. Watts and S. H. Strogatz, *Collective dynamics of small-world networks*, Nature 393(440), 1998.

[17] Liuyuan Lu, W. Aiello and F. Chung, *Random evolution in massive graphs*, IEEE symposium of foundations of computer science, 2001.

[18] B. Bollob$\acute{a}$s, *Random graphs*, London:academic, 1985.

[19] R. Kumar, F. Maghoul, P. Raghavan and *et. al.*, *Graph structure in the web: experiments and models*, Computer networks, 33(309), 2000.

[20] M. Barthelemy, L.A.N. Amaral and *et. al.*, *Classes of small-world networks*, Proceedings of the national academy of sciences, 97(21), 2000.