An Analysis of Open Source Software Development Using Social Network Theory and Agent-Based Modeling

Greg Madey
Vincent Freeh
Renee Tynan
Chris Hoffman
University of Notre Dame
March 2003
The Second Lake Arrowhead Conference on HUMAN COMPLEX SYSTEMS
Hosted by the UCLA Center for Computational Social Sciences
Collaborators

Greg Madey*  
Yongqin Gao  
Nadir Kiyamclar  
Carlos Siu  
Computer Science  
University of Notre Dame

Vincent Freeh*  
Computer Science  
North Carolina State University

Renee Tynan*  
Chris Hoffman  
Department of Management  
University of Notre Dame

*co-PIs – Operation Research, Computer Science, Social Psychology
Goals and Objectives

- **Understanding** Open Source Software Development
  - NSF Grant, 2002-2005
  - CISE/IIS/Digital Society Technology
- Conceptual Explanatory Model of OSS Phenomenon
- Agent-Based Simulation of OSS
- Large Scale Data Mining: SourceForge and other Archives
- Social Network Analysis with Dynamic Attachment
- **Understanding** Social & Task Dynamics
- **Understand** Role of Linchpin Developers
- Frameworks, Data, Models to Support Future Studies
Understanding the **Social and Task Dynamics** that predict Developer Behaviors

**Social Network Analysis**: Longitudinal Study of Preferential Attachment and Dynamic Attachment

**Cross Validation**

**Combined Data Mining**

Parameter Values

Structural Features

Parameter Values

Structural Features

Parameter Values

Conceptual Explanatory Model of OSS: Agent-Based Modeling and Simulation
Overview

- Characteristics of Open Source Software (OSS) Development phenomenon
  - Self-organized
  - Decentralized
  - Emergent properties
  - Complex Adaptive Process
- Research
  - Data collection
  - Social Network Models
  - Agent-Based Models
  - Agent-Based Simulation
Open Source Software (OSS)

- Free ...
  - to view source
  - to modify
  - to share
  - of cost

- Examples
  - Apache
  - Perl
  - GNU
  - Linux
  - Sendmail
  - Python
  - KDE
  - GNOME
  - Mozilla
  - Thousands more
Leaders

Linus Torvalds
Linux

Larry Wall
Perl

Eric Raymond
Cathedral and Bazaar

Richard Stallman
GNU Manifesto
Open Source Software (OSS)

- Development
  - Mostly volunteer
  - Global teams
  - Virtual teams
  - Self-organized - often peer-based meritocracy
  - Self-managed - but often a “charismatic” leader
  - Often large numbers of developers, testers, support help, end user participation
  - Rapid, frequent releases
  - Mostly unpaid
Open Source Software (OSS): Significance

- Contradicts traditional wisdom:
  - Software engineering
  - Coordination, large numbers
  - Motivation of developers
  - Quality
  - Security
  - Business strategy
- Significant component of e-Commerce infrastructure
- Almost everything is done electronically and available in digital form
- Opportunity for Social Science Research -- large amounts of data available

Research issues:
- Understanding motives
- Understanding processes
- Intellectual property
- Digital divide
- Self-organization
- Government policy
- Impact on innovation
- Ethics
- Economic models
- Cultural issues
- International factors
Open Source Software (OSS)

Major Component of e-Technology Infrastructure with major presence in
  e-Commerce
  e-Science
  e-Government
  e-Learning
Apache has over 62% market share of Internet Web servers
Linux on over 7 million computers
Most Internet e-mail runs on Sendmail
Tens of thousands of quality products
Part of product offerings of companies like IBM, Apple
  Apache in WebSphere, Linux on mainframe, FreeBSD in OSX
Corporate employees participating on OSS projects
Open Source Software (OSS)

- Seems to challenge traditional economic assumptions
- Model for software engineering
- New business strategies
  - Cooperation with competitors
  - Beyond trade associations, shared industry research, and standards processes — shared product development!
- Virtual, self-organizing and self-managing teams
- Social issues, e.g., digital divide, international participation
- Government policy issues, e.g., US software industry, impact on innovation, security, intellectual property
Related Research

- Feller and Fitzgerald (ICIS, 2000)
  - Research framework and analysis of the OSS phenomenon
- Hars and Ou (HICSS 2001)
  - Survey of OSS developers
  - Reported on motivations of developers
- Scacchi (IEE Proceedings - Software, 2002)
  - Study of socio-technical processes associated with OSS development practices
- Wolf, Lakhani, and Bates (BCG/MIT Sloan, 2002)
  - Survey of Source Forge Developers
- Hann, Roberts, Slaughter, and Fielding (ICSE, 2002)
  - Survey of Apache developers - economic incentives
Collaborative Social Networks

- Small World Phenomenon
- Research papers on joint authorship
  - Newman, 2001
  - Barabasi et al., 2001
  - Erdos number
- Kevin Bacon Game
- Open Source Software development
  - Link detachment
Related Research: Methodology

  - Collective Dynamics of Small-World Networks
- Watts (Small Worlds, 1999)
- Pumain and Moriconi-Ebrard, (GeoJournal, 1997)
  - Zipf distribution of city sizes
  - Growth Dynamics of the World Wide Web, Power Laws
- Axtell, (Science, 2001)
  - Zipf Distribution of U.S. Firm Sizes,“
  - Evolution of the Social Network of Scientific Collaborations, Power Law distribution
OSS as a Social Network

  - Agents are nodes on a graph (developers)
  - Edges are relationships (joint project participation)
  - Growth of network: random or types of preferential attachment, formation of clusters
  - Network attributes: diameter, average degree, power law, clusters
SourceForge

- VA Software
- Part of OSDN
- Started 12/1999
- Collaboration tools
- 58,685 Projects
- 80,000 Developers
- 590,005 Registered Users
Savannah

- Uses SourceForge Software
- Free Software Foundation
- 1,508 Projects
- 15,265 Registered
Data Collection — Monthly

- Web crawler (scripts)
  - Python
  - Perl
  - AWK
  - Sed
- Monthly
- Since Jan 2001
- ProjectID
- DeveloperID
- Almost 2 million records
- Relational database

<table>
<thead>
<tr>
<th>PROJ</th>
<th>DEVELOPER</th>
</tr>
</thead>
<tbody>
<tr>
<td>8001</td>
<td>dev378</td>
</tr>
<tr>
<td>8001</td>
<td>dev8975</td>
</tr>
<tr>
<td>8001</td>
<td>dev9972</td>
</tr>
<tr>
<td>8002</td>
<td>dev27650</td>
</tr>
<tr>
<td>8005</td>
<td>dev31351</td>
</tr>
<tr>
<td>8006</td>
<td>dev12509</td>
</tr>
<tr>
<td>8007</td>
<td>dev19395</td>
</tr>
<tr>
<td>8007</td>
<td>dev4622</td>
</tr>
<tr>
<td>8007</td>
<td>dev35611</td>
</tr>
<tr>
<td>8008</td>
<td>dev7698</td>
</tr>
</tbody>
</table>
**OSS Developer - Social Network**

Developers are nodes / Projects are links

- 24 Developers
- 5 Projects
- 2 Linchpin Developers
- 1 Cluster

- Project 6882
- Project 7028
- Project 15850
- Project 7597
- Project 9859
Regression: Number of projects that developers are on

<table>
<thead>
<tr>
<th># of projects</th>
<th># of developers on that many projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21488</td>
</tr>
<tr>
<td>2</td>
<td>3688</td>
</tr>
<tr>
<td>3</td>
<td>1086</td>
</tr>
<tr>
<td>4</td>
<td>413</td>
</tr>
<tr>
<td>5</td>
<td>177</td>
</tr>
<tr>
<td>6</td>
<td>76</td>
</tr>
<tr>
<td>7</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
</tr>
</tbody>
</table>

Log( # of Developers) vs. Log( # of Projects)

\[ y = 10.6905 - 3.70892 \times \]

\[ R^2 = 0.979906 \]

Scale Free – Power Law (developers)
Empirical Data

Scale Free – Power Law (projects)

Growth Rates
Agent Based Model

- Grow an Artificial SourceForge
- Model as a Collaborative Social Network
- Parameterize model with empirical data
- Guess “hidden” processes, mechanisms and developer behaviors
- Agent Based Simulation
- Verify, validate, and iterate
- Discover how OSS works => Understanding
Agent Based Simulation

- Java Swarm / JDBC / Oracle Relational DB
  - Same database table design, same analysis tools
- Developer class
- Each simulated developer is an instance of “Developer” with random attributes and behaviors
  - Local decision logic
  - Simulates self-organization
  - Create new projects
  - Join existing projects
  - Abandon a project
Results: New Understanding?

- Random Attachment does not generate Power Law
- Preferential Attachment does, but …
- Bipartite nature of the social network
  - Preferential and random attachment can be implemented independently on developers and projects
  - Both display power laws
Results: New Understanding?

- “Young upstart” phenomenon not captured with preferential attachment
- Added fitness
  - Random fitness parameter for projects and developers
  - Power law still observed in both projects and developers and “young upstart” feature modeled
- Still missing a real feature observed in the real SourceForge …
Results: New Understanding

- Problem
  - Static fitness insufficient
  - Dynamic fitness needed
  - Discovered a life cycle property of the projects
    - Reflects both dynamic attachment and detachment
- Thus, dynamic fitness, with a life cycle
  - Power law preserved!
Results/Limitations/Future Work

- Using social network theory, and agent-based modeling and simulation to gain **understanding**
- Role of social network theory and agent based modeling:
  - A framework
  - An investigation tool
  - What does it mean if the simulated SourceForge fits the real SourceForge really well?
- Survey instruments to collect additional data on individual project and developer behavior
- More types of data
- More data sources
  - Savannah
  - Large OSS projects
Thank You

Questions?