3-1-2008

A Tool for Identifying Swarm Intelligence on a Free/Open Source Software Mailing List

Lars P. Linden
University of Central Florida, llinden@bus.ucf.edu

A TOOL FOR IDENTIFYING SWARM INTELLIGENCE ON A FREE/OPEN SOURCE SOFTWARE MAILING LIST

Lars P. Linden
University of Central Florida
llinden@bus.ucf.edu

ABSTRACT

A software tool designed using the concepts of swarm intelligence and text mining is proposed as an aid in the analysis of free/open source software (FOSS) development communities. A prototype of the tool collects textual data from an electronic mailing list, a primary mode of FOSS developer communication. The tool enables a user to compare patterns of discussion topics found in the text with patterns of swarm intelligence. The research of this design is congruent with Madey et al.'s (2002) observation that the open source software development phenomenon shows an emergent behavior and can be modeled after agent-based, biologically-inspired swarms. The goal of a tool for identifying emergent intelligence on FOSS mailing lists is to increasing the user's understanding of a given FOSS development community.

Keywords

Free software, open source software, swarm intelligence, text mining, complexity theory

INTRODUCTION

Free software and open source software (FOSS) licenses and the Internet are two important artifacts that enable developers to work together and develop software (Kogut and Metiu, 2001). Understanding these FOSS development communities is important due to the increasingly mainstream adoption of FOSS (Fitzgerald, 2006) and the possibility of applying the communities' distributed and innovative methods to domains other than software development (Horwitch et al., 2000). However, as enterprises adopt or consider adopting open source software, concerns have been raised about the time spent to stay informed of developments (Byfield, 2007) and the need for people qualified to support it (Greenemeier, 2005).

Tools and metrics aimed at measuring and analyzing the characteristics of FOSS development communities are being developed (for example: Kumar and Wang, 2006, Capiluppi et al., 2003, German and Mockus, 2003). The variety and interest in FOSS development communities are such that calls have been made to organize the efforts of FOSS researchers with central repositories of data, models, and tools. This paper aims to contribute to the FOSS data analysis "toolbox" with a tool design that conceptually combines (1) the text mining of FOSS mailing lists and (2) Madey et al.'s (2002) observation that FOSS development processes are swarm-like and exhibit emergent properties. The proposed tool aims to contribute by allowing a more immediate comparison of FOSS data and patterns of swarm intelligence.

COMPLEXITY THEORY AND INFORMATION SYSTEMS DEVELOPMENT

The proposed tool design builds upon a trajectory of Information Systems (IS) research that references the science of complexity, which investigates naturally-occurring non-linear complex systems (Merali, 2004, p. 418). Aligned with this research stream is the observation that the processes of open source software development are self-organizing and exhibit emergent properties (Sowe et al., 2007, von Hippel and von Krogh, 2003, Truex et al., 1999, Iannacci and Mitleton–Kelly, 2005). Several researchers have drawn connections between the broad topic of complexity theory and various information systems phenomena, software development included.

Merali (2004) contrasts the emergent bottom-up behavior of the Internet-enabled networks with the top-down, hierarchically controlled paradigm. A network, such as the Internet, exhibits emergent phenomena whereby the "macroscopic 'whole' displays a set of properties that is distinct from those displayed by any subset of its individual constituents and their interactions" (Merali, 2004, p. 422).

Information systems development is also described as having emergent properties. Benbya and McKelvey (2006) compare top-down, traditionally planned development with bottom-up, self-organized emergent adaptations. Complexity theory is described as when "order emerges through the interaction of organisms or agents" (Benbya and McKelvey, 2006, p. 16). Complexity theory is used to create principles that are aimed at guiding information systems development as a continuously adaptive process where organization and IS co-evolve.

For example, Iannacci and Mitleton-Kelly's (2005) described the behavior of the Linux kernel community as a complex evolving system, in which the structure of the development community exhibits self-organizing characteristics. The emergent structure is a heterarchy, comprising of loosely-coupled hierarchies. Some hierarchical structure exists. For example, patches flow toward the project leader and a central release system. However, individual decision makers consider decisions previously made within the community and coordinate with lieutenants and others in smaller subgroups.

Swarm Intelligence

One particular model of self-organization is swarm intelligence. Swarm intelligence is "the emergent collective intelligence of groups of simple agents" (Bonabeau et al., 1999, p. xi). Biologically-inspired concepts are drawn from the problem-solving behaviors of flocks of birds and colonies of ants. Behaviors seen in nature are modeled by algorithms and then applied to human problem domains.

Madey et al. (2002) applied swarm intelligence models to IS research using simulation methodology. The simulation runs with software agents modeled after OSS developers. The agents are grouped into project swarms and even clusters of project swarms to examine the simulated effect of an entire "OSS development swarm" (p. 1473). Rather than considering this large scope, where individual developers are able to participate in several projects, the goal of the proposed tool design is to focus on observing behavior of subgroups within one particular community.

Particle Swarm Algorithm

One algorithm from swarm intelligence research is particle swarm optimization (PSO). PSO is a technique used to model social networks that have problem-solving capabilities built in (Kennedy, 2005, Kennedy and Eberhart, 1995). PSO provides algorithms that specify how individual agents traverse a problem space, influencing each other as they go. The objective here, with respect to the proposed tool, is to draw parallels between the details of a selected particle swarm algorithm and the FOSS development community.

The social element is that each particle (an individual) moves in a way that is influenced by its neighbors. A particle changes positions in the multi-dimensional parameter space based on (1) its memory of previous states and (2) its polling of surrounding particles.

This polling of neighbors represents the community interaction, and particles are able to improve their assessment of the optimal position in the social topology by taking their neighbor's position into consideration. The swarm of particles moves within the parameter space, problem solving.

The proposed tool design aims to enable a researcher to compare social patterns found in mailing list communications with patterns of swarm intelligence. Because a mailing list represents only a partial representation of a given FOSS group's social interactions, the tool is considered an exploratory tool that is helpful to the degree that it allows the discovery and cataloguing of self-organizing behavior patterns of community subgroups. This, then, leads to the central question of the research: Is it possible to develop an information system tool that, given e-mail messages, allows a user to identify the swarm behavior of a subgroup of individuals within a given FOSS project?

DATA SOURCES AND DATA ANALYSIS METHODS FOUND IN FOSS RESEARCH

Researchers use a variety of data sources and data analysis methods when investigating FOSS artifacts and communities. Selected examples of FOSS research are highlighted below to show the variety and to place the proposed tool in context.

Madey et al. (2002), for example, studied the social network of developers working on open source projects using metadata found on SourceForge.net and concluded that the network displayed a power-law relationship, characterized by a small percentage of developers with many relationships and a large percentage of developers with few relationships.

Sowe, Stamelos, and Angelis (2007) examined Debian GNU/Linux project mailing lists and found particular knowledge-sharing patterns, for example, the emergence of hierarchy that included people who dominated question posting and replying.

Capiluppi, Lago, and Marisio (2003) studied a random sample of projects listed in the OS project portal "freshmeat" and reported a variety of project attributes, including a characterization of an "active" project as being a project that, over a period of time, has increases across an entire a set of measurements (vitality, popularity, and subscriber and developers).

Yu and Ramaswamy (2007) mined the data of CVS repositories, analyzing various measures, such as the frequency of co-editing and the frequency of task, with a clustering method in order to identify core members and associate members.

Jensen and Scacchi (2004) used data mining techniques to automatically examine various artifacts of development (e.g. source code, communications transcripts) and to produce a model of software development processes.

Stewart, Darcy, and Daniel (2006) obtained the source code from selected projects hosted by SourceForge.net and applied function data analysis to study the patterns of evolution in the complexity of open source software.

The growth in studies that obtain and analyze publicly-available data prompted Gasser et al. (2005) to propose a global, self-managed research infrastructure for aiding efforts of gathering and analysis of FOSS data. In another global effort, Hahsler and Koch (2005) proposed a methodology for the automated collection of FOSS data into a repository.

The tool design proposed in this paper aims to contribute to this research stream by combining the concepts of swarm intelligence and text mining to facilitate the discovery of emergent behavior in FOSS mailing lists.

TEXT MINING

In addition to swarm intelligence, text mining is another important theoretical foundation for the design of the tool. The data that is analyzed by the tool is structured HTML that includes unstructured text, and so the matching of patterns in textual data is a research subfield that can greatly inform the design of the tool.

Knowledge discovery in databases (KDD) is a field of endeavor that studies how to identify useful patterns in large collections of data (Fayyad et al., 1996). Text mining is an extension of KDD and data mining, one that focuses on textual data. Standard text mining makes use of statistical and natural language processing methods. Intelligent text mining combines standard text mining methods with artificial intelligence methods to enable a person, such as a decision maker, to interact during the analysis (Durfee, 2006).

Text Mining FOSS Mailing Lists

Social science research has made use of electronic mailing lists. Kuk (2006) regards discussion threads of a mailing list as a basic unit of analysis because "the threads provide a virtual place for epistemic interactions and serve as conduits of knowledge sharing to a wider epistemic network embedded within and beyond the borders of discussion threads" (p. 1033). Text mining processes applied to the examination of mailing lists increase the efficiency of an analysis.

The prototype of the proposed design currently uses only a rudimentary text mining approach that includes acquisition, pre-processing, and regular expression pattern matching. The first tests of the prototype targeted the threaded discussion of the Linux Kernel Mailing List (LKML) and uses postings that span from January 1, 2007 to February 28, 2007. Using a series of Perl scripts, the HTML pages of the archived list were non-interactively downloaded. The harvesting of the posts was performed in many stages with a wait loop, as suggested by Howison and Crowston (2004), so as not to strain the servers. Perl scripts pre-processed the data, cleaning the HTML and extracting particular strings of interest, such as the date, subject line, and the body of the message.

TOOL DESIGN AND PROTOTYPE DEVELOPMENT

The design of the tool is guided by text mining and swarm intelligence. A prototype based upon the proposed design is under development as a proof-of-concept (Gregg et al., 2001). The prototype enables a user to input keywords. Visualizations are produced and indicate interconnected relationships of the community subgroup participating in the mailing list, as shown in Figure 1 (next page). The horizontal axis represents time and the vertical axis represents individual posters. The points in the visualization represent one email. The points are color-coded to indicate the presence of the keywords. Lines are used to indicate possible social influences found amongst the posts, and are drawn when recent posts include the same search term. A visualization of a particle swarm algorithm can also be produced, as shown in Figure 2 (next page). The goal of the proof-of-concept is to demonstrate how a user could compare the visualizations of mailing list data and particle swarm algorithm.

For the user of the tool, the objective is exploration. The user explores by entering keywords and examining the resulting visualizations for interesting patterns that appear to match the baseline PSO patterns. The prototype should be tested to determine if the tool allows a user to detect whether or not email messages of subgroups of a FOSS development community exhibit swarm behavior.

The prototype was written as a set of Perl scripts and modules. The visualizations were created using OpenGL. Development was performed using a GUN/Linux distribution with the KDE windowing environment. Tcl/Tk was used as the windowing mechanism. The pre-processed text was stored in a MySQL database. The PSO algorithm was modeled after Kennedy's (2005) particle swarm pseudo-code and was coded as a set of object-oriented Perl modules that included the following classes: Topology, Swarm, Particle, CognitiveElements, CognitiveElement, and Neighborhood.

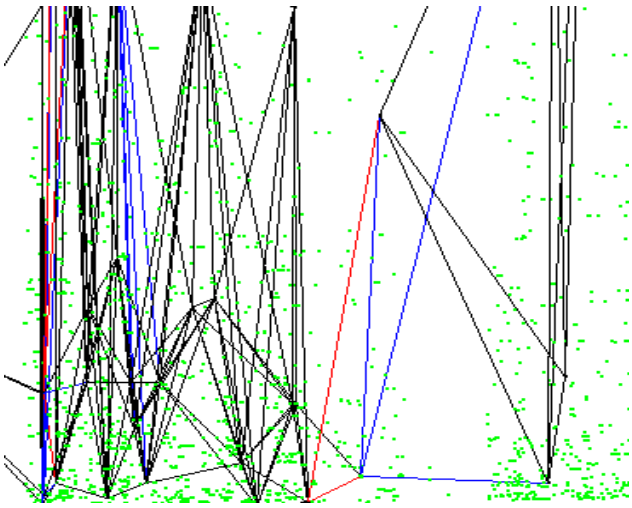

Figure 1. Close-up of the prototype tool after searching for the term "readahead" (points are emails, lines are matched emails)

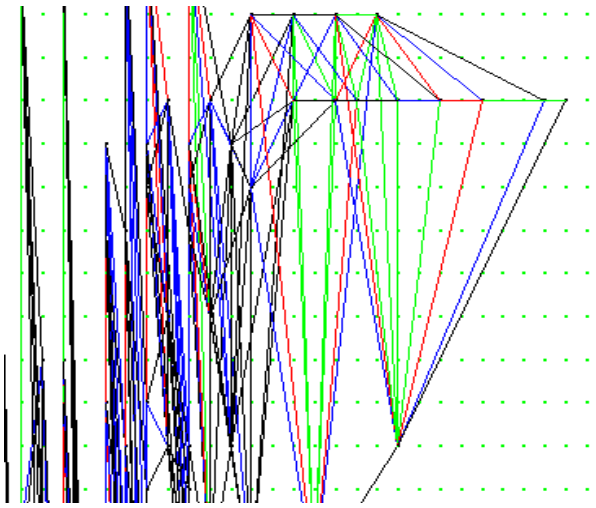

Figure 2. Close-up of the prototype tool showing the output of a particle swarm optimization algorithm.

Limitations

Several limitations are evident in this research. Currently, the research relies upon an assumption that social influence is exhibited via emails found in threaded discussion. Second, the size of swarms is usually 20-30 individuals (Kennedy, 2005), a range that might not include the participant tally of a given FOSS projects. Third, a FOSS developer community may be subject to anti-social behavior. This issue is raised by Iannaci and Mitleton-Kelly (2005), who discuss how developers within the community may compete with each other. Consideration of this issue may require a change of algorithm used. Fourth, Merkle and Middendorf (2005) describe a "queen particle" as a particle that is located at the center of the swarm and has a dominant role, and this concept might be useful in modeling project leadership. Fifth, the posts and replies are texts, artifacts produced by people, and not the interacting beings themselves. Finally, on the text mining aspects of the design, Conklin (2006) provides guidance as to many of the limitations of data mining FOSS projects. For example, the source of published data may be taken offline and, thus, no longer be available for replication studies.

CONCLUSION

While project leaders and frequent developers are familiar with the history and current workings of the development community with which they are involved, people who have only recently become interested in adopting a particular project may find it difficult to assess the workings of particular community. In addition to practitioners, researchers also are interested in analyzing FOSS communities and are working to improve the collection and analysis FOSS artifacts. The proposed tool design aims to aid in the understanding of FOSS development communities by facilitating the exploration of mailing list data for evidence of emergent behavior.

REFERENCES

1. Benbya, H., and McKelvey, B. "Toward a complexity theory of information systems development," *Information Technology & People* (19:1) 2006, pp 12-34.
2. Bonabeau, E., Dorigo, M., and Theraulaz, G. *Swarm Intelligence from Natural to Artificial Systems* Oxford University Press, New York, 1999.
3. Byfield, B. "FOSS consulting offers special advantages and challenges," 2007. http://www.itmanagersjournal.com /feature/21889. Accessed on January 19, 2008.
4. Capiluppi, A., Lago, P., and Morisio, M. "Characteristics of open source projects," Seventh European Conference on Software Maintenance and Reengineering, 2003, pp. 317-327.
5. Conklin, M.S. "Beyond Low-Hanging Fruit: Seeking the Next Generation in FLOSS Data Mining," in: *IFIP International Federation for Information Processing*, Springer, Boston, 2006, pp. 47-56.
6. Durfee, A. "Text Mining Promise and Reality," Americas Conference on Information Systems, Acapulco, Mexico, 2006.

7. Fayyad, U.M., Piatetsky-Shapiro, G., and Smyth, P. "From Data Mining to Knowledge Discovery," in: *Advances in knowledge discovery and data mining*, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (eds.), MIT Press, Menlo Park, CA, 1996.
8. Fitzgerald, B. "The Transformation of Open Source Software," *MIS Quarterly* (30:3) 2006, pp 587-598.
9. Gasser, L., Ripoche, G., Sandusky, B., and Scacchi, W. "A Global Research Infrastructure for Multidisciplinary Empirical Science of Free/Open Source Software: A Position Paper," 2005.
10. German, D., and Mockus, A. "Automating the measurement of open source projects," 3rd Workshop on Open Source Software Engineering, 2003.
11. Greenemeier, L. "Open Source Goes Corporate," in: InformationWeek, 2005. http://www.informationweek.com/story/showArticle.jhtml?articleID=171200352. Accessed on Sept. 26, 2005.
12. Gregg, D., Kulkarni, U., and Vinzé, A. "Understanding the philosophical underpinnings of software engineering research in information systems," *Special Issue of Information Systems Frontiers* (3:2) 2001, pp 169-183.
13. Hahsler, M., and Koch, S. "Discussion of a Large-Scale Open Source Data Collection Methodology," 38th Annual Hawaii International Conference on System Sciences, Hawaii, 2005.
14. Horwitch, M., Parikh, M., and Nina, Z. "Open innovation: Transferring lessons from software for modern value creation," CISEP Workshop on Innovation and Diffusion in the Economy, Lisbon, Portugal, 2000.
15. Howison, J., and Crowston, K. "The perils and pitfalls of mining SourceForge," Proc. of Workshop on Mining Software Repositories at the International Conference on Software Engineering ICSE, 2004.
16. Iannacci, F., and Mitleton–Kelly, E. "Beyond markets and firms: The emergence of Open Source networks," in: *First Monday*, 2005. http://www.firstmonday.org/issues/issue10_5/iannacci/. Accessed on September 1, 2005.
17. Jensen, C., and Scacchi, W. "Data mining for software process discovery in open source software development communities," International Workshop on Mining Software Repositories (MSR 2004) W17S Workshop - 26th International Conference on Software Engineering, Edinburgh, Scotland, UK, 2004, pp. 96 -100.
18. Kennedy, J. "Particle Swarms: optimization based on sociocognition," in: *Recent developments in biologically inspired computing*, L.N. DeCastro and F.J. Von Zuben (eds.), Idea Group, Hershey, PA, 2005.
19. Kennedy, J., and Eberhart, R. "Particle swarm optimization," IEEE International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948.
20. Kogut, B., and Metiu, A. "Open-source software development and distributed innovation," *Oxford Review of Economic Policy* (17:2), Summer 2001, pp 248-264.
21. Kuk, G. "Strategic Interaction and Knowledge Sharing in the KDE Developer Mailing List," *Management Science* (52:7) 2006, pp 1031-1042.
22. Kumar, S., and Wang, L. "Metrics to Support Open Source Software Adoption Decisions," Americas Conference on Information Systems, Acapulco, México, 2006.
23. Madey, G., Freeh, V., and Tynan, R. "Agent-based modeling of open source using swarm," Americas Conference on Information Systems, Dallas, Texas, 2002.
24. Merali, Y. "Complexity and information systems," in: *Social Theory and Philosophy of Information Systems*, J. Mingers and L. Willcocks (eds.), Wiley, 2004, pp. 407-446.
25. Merkle, D., and Middendorf, M. "Swarm intelligence," in: *Search methodologies: introductory tutorials in optimization and decision support techniques*, E.K. Burke and G. Kendall (eds.), Springer, New York, 2005, pp. 402-435.
26. Sowe, S., Stamelos, I., and Angeli, L. "Understanding knowledge sharing activities in free/open source software projects: An empirical study," *Journal of Systems and Software* (in press), 2007.
27. Stewart, K.J., Darcy, D.P., and Daniel, S.L. "Opportunities and Challenges Applying Functional Data Analysis to the Study of Open Source Software Evolution," *Statistical Science* (21:2) 2006, pp 167-178.
28. Truex, D.P., Baskerville, R., and Klein, H. "Growing systems in emergent organizations," *Communications of the ACM* (42:8) 1999, pp 117-123.
29. von Hippel, E., and von Krogh, G. "Open source software and the 'private-collective' innovation model: issues for organization science.," *Organization Science* (14:2), Mar/Apr 2003, pp 209-223.
30. Yu, L., and Ramaswamy, S. "Mining CVS Repositories to Understand Open-Source Project Developer Roles," International Conference on Software Engineering, Proceedings of the Fourth International Workshop on Mining Software Repositories, 2007, p. 8.