Project Number 001907

DELIS

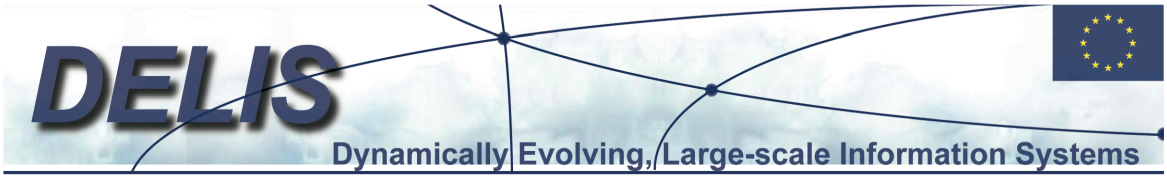
Dynamically Evolving, Large-scale Information Systems

Integrated Project

Member of the FET Proactive Initiative **Complex Systems**

Deliverable D5.2.4

Modeling open source development as evolving networks



Start date of the project:	January 2004
Duration:	48 months
Project Coordinator:	Prof. Dr. math. Friedhelm Meyer auf der Heide Heinz Nixdorf Institute, University of Paderborn, Germany
Due date of deliverable:	December 2006
Actual submission date:	December 2006
Dissemination level:	PU – public
Work Package 5.2:	Evolved Tinkering and Degeneracy as Engineering Concepts
Participants:	Universitat Pompeu Fabra (UPF), Barcelona, Spain Uiversita di Bologna (UniBO), Italy Telenor Communication AS (Telenor), Oslo, Norway
Authors of deliverable:	Sergi Valverde (svalverde@imim.es) Ricard V. Solé (ricard.sole@upf.edu)

Abstract

This report comprises the complete D5.2.1 deliverable as specified for workpackage WP5.2 in Subproject SP5 of the DELIS (Dynamically Evolving Large-scale Information Systems) Integrated Project.

The essential goal of the DELIS project is to understand, predict, engineer and control large evolving information systems. The main aim of this workpackage is to understand how evolved structures emerge in networks when there is no central design or control.

A major aspect of this work involves the design of measures and models that elaborate structure within networks. Here we consider the e-mail contact networks between programmers in Open Source projects. Interestingly, we show how a distributed process can lead to hierarchy (a "rich club" significant developers) and hence some level of centralization. We have formulated measures and applied them to empirical data from a number of OS projects. Also, we have developed a simple model that reproduces the observed structures.

Contents

1 Introduction

In both nature and engineering, complex designs often emerge from distributed collective processes. Open-source software (OSS) communities constitute remarkable examples of distributed intelligence: a social network of interacting agents that send, receive and process information at different timescales and levels of detail. By modeling these social networks (OSN) we can compare them to other complex networks and so build up evidence for basic principles of self-organization [1]. We think that social network analysis is an innovative approach to the OS phenomenon and will provide deep insights about how OS software systems develop. Open source communities are a nice illustration of how human interaction takes place these days. The electronic support tracks every social interaction and enables us to gather highly detailed registers of human activities.

On the other hand, the study of OS communities is different from other studies of online communities [2], which are apparently quite similar if we look at how communication is implemented (i.e., Internet-based communication). Instead, interaction in the OS community stems from the common goal of achieving a functional system, i.e., an OS software system, while interaction on community web sites considers a much more diverse range of interests and motivationS. The OS community is a team of people with a well-defined purpose while the community web site describes a real world social network running on the Internet. Other differences have a more quantitative basis. For example, a way to understand human behaviour is by analying the interevent time distribution $P(T)$ of elapsed time between any two consecutive events. Many human activities are characterized by the power-law tail $P(T) \approx T^{-\alpha}$ where $\alpha = 3/2$ or $\alpha = 1$ [3]. The α exponent describes the way individuals execute the different tasks ahead of them. For example, the $a = 1$ exponent has been observed in web browsing, e-mail and library datasets, signaling the existence of limitations on the queue length. Here, we have observed a different pattern of individual behaviour (see fig.1B).

Typical applications of social network analysis include measures of member importance (i.e., node centrality) [14]. Here, nodes with highest centrality indices represent the main developers in the community [4]. In addition, investigating OS social structure is a useful way to understand how software teams develop complex systems. This approach enabled us to build quantitative reference models explaining human behavior in OS software development [8]. We believe that useful and realistic reference models will enable enhanced management of complex software processes. For example, careful comparison between real process measurements and model predictions highlights critical deviations from the original development plan.

2 Weighted Network Analysis

Social network analysis represents agent relationships with nodes and links [14]. Every node represents an agent within the social network; links (i, j) denote social ties between agents i and j. Here we have studied a dataset describing the e-mail activity of 120 different software communities [4]. This data comes from the SourceForge web site (http://sourceforge.net), which is a large and popular OS project repository. In the following, we will consider every software community in isolation. Previous studies on OS networks have studied the full network for the entire software development community at SourceForge, that is, they have aggregated all software communities into a single, huge community [5]. Our study is different and we have analysed 120 different social networks, instead.

To determine an individual's social relevance, we have analyzed the amount of submitted and received e-mails within the OS community. It is important to recognize that not all e-mails have the same influence in the development process. Then, we have discarded all e-mails not directly related to the software process (i.e., personal e-mails, spam, etc) and we have limited our consideration to email traffic associated to bug fixes and bug reporting (which is a crucial task for software development). From this listing of filtered e-mails we can reconstruct the OS social network as follows. In the social

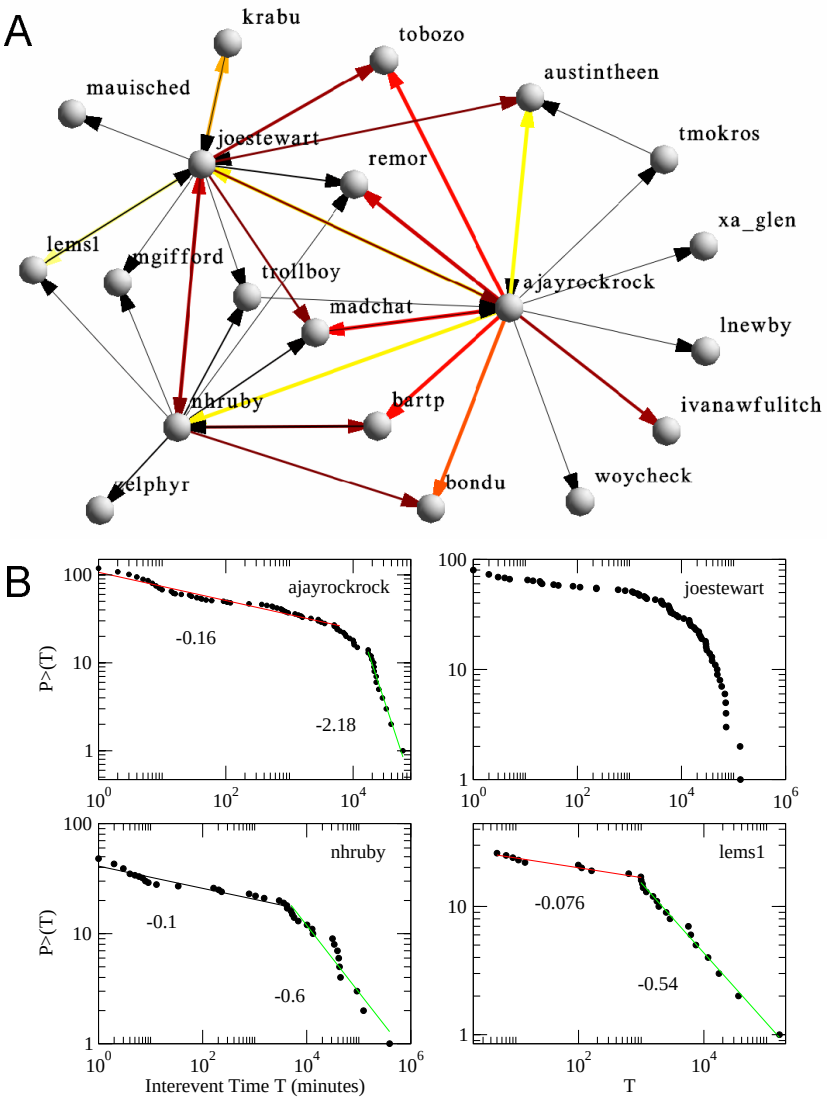


Figure 1: Study of e-mail exchanges while fixing bugs in the *phpsplash* OS community. (A) Heterogeneous features in the e-mail interaction network. (B) Cumulative distributions $P_>(T)$ of interevent times T (i.e., elapsed time between two consecutively submitted e-mails) in the e-mail activity of four different software developers. The second most active developer (joestewart) displays a rather exponential distribution $P_>(T)$.

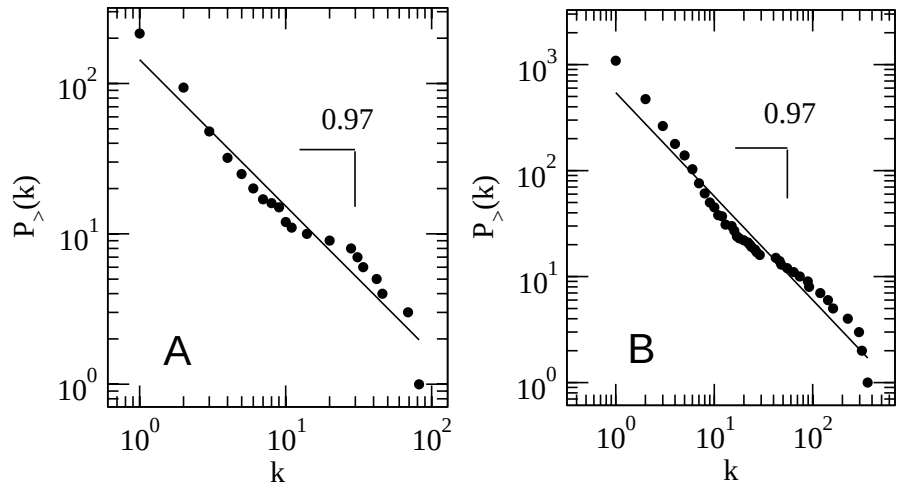


Figure 2: Cumulative Degree Distributions (A) TCL (B) Python.

network, every node i represents a member and a link (i, j) denotes non-zero flow of e-mails from member i to member j (see fig.1A).

Unlike previous studies of OS structure [4], here we will consider the intensity of social interaction by measuring the amount of information flowing through the tie, that is, the total number of e-mails exchanged between any pair of nodes[1]. We refer to this value as the link weight $w_{i,j}$. OS communities display a heterogenous interaction pattern because the probability of having a link with weight $w_{i,j}$ decays as a power law (see fig. 2),

$$P(w_{i,j}) \propto w_{i,j}^{-\gamma} \tag{1}$$

that is, there is a few pairs of members exchanging much more e-mails than with the rest of the community. Our analysis suggests these key members play the role of hubs in the social network, that is, they have the largest number of connections with the average community member.

What is the origin of this highly skewed distribution of e-mail traffic? Is this pattern a signature of (nearly) optimal social organization [7]? Supporters of OSS development argue that decentralization leads to a distinctive social organization that solves the communication bottleneck long associated to large software teams [6]. Interestingly, we have found at least two different models that reproduce such heterogenous link weight distribution [1]. Then, we require additional network measurements in order to characterize OS communities properly.

3 Rich-Club Phenomenon

In order to better understand the role played by hubs in OS networks, we have analyzed the subgraph composed by the hubs and the links connecting them [8]. This subgraph constitutes the so-called rich-club [12] or an elite of highly connected and mutually communicating members that control the flow of information generated by the OS community (see fig.3A). This is consistent with empirical observations of existing OS communities, where typically a small number of core developers contribute nearly 90 percent of changes [13]. We can detect this rich-club by means of the rich-club coefficient,

$$\Phi(k) = \frac{2E_{>k}}{N_{>k}(N_{>k} - 1)} \tag{2}$$

where $E_{>k}$ represents the number of links between the hub nodes $N_{>k}$ having more than k links. $\Phi(k)$ indicates the ratio of observed number of links out of all possible links between $N_{>k}$ nodes.

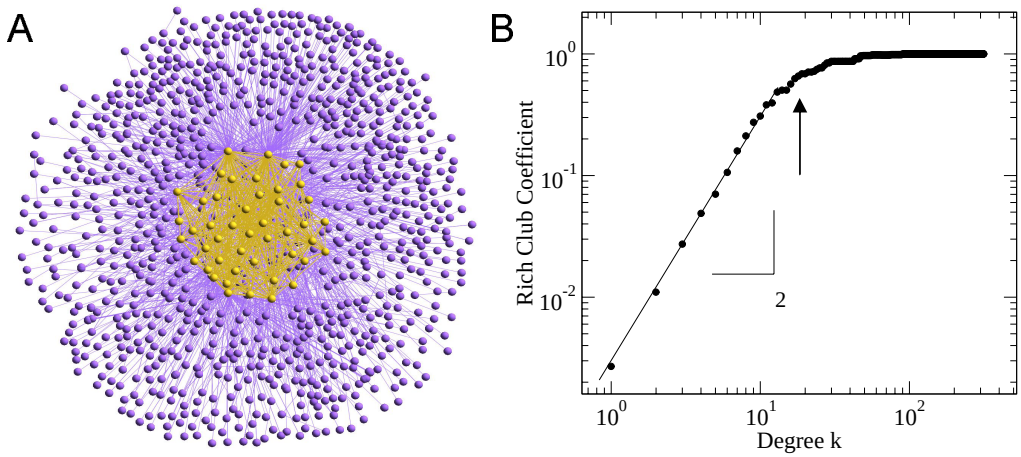


Figure 3: Correlations and rich-club phenomenon in the *Python* OS community. (A) Visualization of the rich-club where yellow balls depict hubs having $k > k_c$. (B) The rich-club coefficient (see text) scales with degree k and saturates once $k > k_c$. The pointing arrow indicates the crossover $k_c \approx 10$.

This coefficient is a correlation measure which is non-trivially related to the average nearest-neighbors degree. For OS networks, $\Phi(k)$ increases for $k < k_c$ and saturates for $k > k_c$ (see fig. 3B). Such monotonic increase is often interpreted as a signature of rich-club phenomenon. Here, we will interpret the crossover k_c as the characteristic size of the OS rich-club. Moreover, we suggest that $N_{>k_c}$ gives a better indicator of the number of active developers in any community than the total number of members in the OS community or N, the number of nodes in the full OS network.

4 Non-local Evolution

Interestingly, a very simple model predicts the evolution and dynamics of OS networks, including the heterogenous distribution of connectivities and local measurements of correlations[8]. An important assumption made by our model is that agents exploit social cues to evaluate one another's social status. Members earning high social status are the most visible [9] and thus, they will be contacted much more frequently. These key members have mean global picture of the whole community, instead of being aware of some specific parts of it. Members having a deeper knowledge of the overall system are likely to manipulate high amounts of information and we should expect them to canalize information flowing from many different parts of the social network [10].

Our algorithm (see [8] for details) produces a synthetic OS network that has the same number of nodes N and links L of the real OS network. The model is a great simplification of the real process and combines two basic mechanisms: network growth and a preferential attachment rule. At every step, a new member joins the community. This member will report a small number of m new e-mails to existing community members. However, e-mail destinataries are not chosen at random. Indeed, it is very likely that core developers will be chosen more frequently as e-mail targets from newcomers. Taking into account the above, we propose the amount of processed traffic (or node load) is a good surrogate of social status (and thus, of member visibility).

5 Summary

Social network analysis has shown that open source communities are closer to the Internet and communication networks than to other social networks (e.g., the network of scientific collaborations). A distinguished feature of OS networks is the presence of the rich-club phenomenon. We have shown that OS communities are elitarian clubs where strong hubs control the global flow of information generated by many peripherical individuals. Our conclusions are consistent with qualitative observations done by researchers of the open-source phenomenon [11]. This is, as far as we know, the first time that quantitative evidence of elitism in technological communities has been provided.

The rich-club phenomenon in OS networks seems to be related to a pattern of non-local evolution. We have presented a model that predicts many global and local social network measurements of software communities. Our model assumes that reinforcement is nonlocal, that is, future e-mails are not independent of past communications. Fixing a software bug is a global task which requires the coordination of several members in the community. Any e-mail response requires to consider all the previous communications regarding the specific subject under discussion.

On a more general view, our study is the first quantitative evidence for the emergence of hierarchy in distributed networks of interacting agents. Different outcomes of the OS evolution process would have been expected, including the formation of a purely hierarchical tree of relations among developers or a purely SF system. The observed community organization indicates that even distributed systems develop internal hierarchies, thus suggesting that some amount of centralized, global knowledge might be inevitable.

References

[1] Sergi Valverde, Guy Theraulaz, Jacques Gautrais, Vincent Fourcassié, and Ricard V. Solé, "Self-Organization Patterns in Wasp and Open-Source Communities", IEEE Intelligent Systems, vol. 21, no. 2, pp. 36-40, Mar/Apr, 2006. [DELIS-TR-0434]

[2] Lada A. Adamic, Orkut Buyukkokten, and Eytan Adar, "A social network caught in the Web", First Monday, 8(6), June (2003).

[3] A.-L. Barabási, "The origin of bursts and heavy tails in human dynamics", Nature 435, 207211 (2005).

[4] Kevin Crowston and James Howison, "The Social Structure of Free and Open Source Software Development", First Monday, February (2005).

[5] Jin Xu, Yongqin Gao, Scott Christley, and Gregory Madey, "A Topological Analysis of the Open Source Software Development Community", Proc. IEEE 38th Hawaii Int. Conf. Syst. Sci, (2005).

[6] M. E. Conway, "How Committees Invent?", Datamation, 14 (4), pp. 28-31, (1968).

[7] M. Anghel, Zoltán Toroczkai, Kevin E. Bassler, and G. Korniss, "Competition-Driven Network Dynamics: Emergence of a Scale-Free Leadership Structure and Collective Efficiency", Phys. Rev. Lett. 92, 058701 (2004)

[8] Sergi Valverde and Ricard V. Solé, "Self-organization and Hierarchy in Open-Source Social Networks", Submitted to Physical Review E (2006). Previous preprint: http://arxiv.org/abs/physics/0602005. [DELIS-TR-0433]

[9] Duncan J. Watts, Peter Sheridan Dodds, and M. E. J. Newman, "Identity and Search in Social Networks", Science 296 (5571), 1302 (2002).

[10] Peter Sheridan Dodds, Duncan J. Watts , and Charles F. Sabel, "Information exchange and the robustness of organizational networks", PNAS, 100(21), pp. 12516-12521, (2001).

[11] Lerner, J., and Tirole, J., *Journal of Industrial Economics*, **52**, pp. 197-234, (2002).

[12] Zhou, S. and Mondragon, R.J., IEEE Comm. Lett., 8, pp. 180182, (2004).

[13] Mockus, A., Fielding, A., Herbsled, J., in Proc. Int. Conf. Soft. Eng., Limerick, Ireland, pp. 263-272, (2002).

[14] S. Wasserman, and K. Faust, "Social network Analysis: Methods and Applications", Cambridge University Press, Cambridge (1994)