Project Number 001907

DELIS

Dynamically Evolving, Large-scale Information Systems

Integrated Project

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

Deliverable D5.4.2

Understanding and engineering multi-scale selection in evolving dynamic 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 5.4:	Multi-Scale Topology Evolution in Natural and Artificial Networks
Participants:	Universitat Pompeu Fabra (UPF), Barcelona, Spain Universita 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) David Hales (hales@cs.unibo.it) Ozalp Babaoglu (babaoglu@cs.unibo.it)

Abstract

This report comprises the complete D5.4.2 deliverable as specified for workpackage WP5.4 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. In this workpackage we consider a broad, though often overlooked, range of network evolution phenomena that evidence multi-scale and multi-level phenomena. We believe that understanding such phenomena is key to relating form to function in complex evolving networks that display emergent properties and rich self-organised structure.

In the first section we report on-going work on the analysis of Open Source software development networks at different scales. This allows for the separation of endogenous and exogenous factors that cause change over time. This method has application in determining, automatically, the software modules that comprise the system. This could be of particular use when refactoring of large systems is required.

In the second section we report on-going work on novel models of multi-level selection applicable to peer-to-peer (P2P) networks. We are interested in "group selection" models which emerge higher-level selection units from selection operating at the individual level. We have begun to compare models within a general framework. We identify a recent novel model (the group-splitting model) that appears to offer potentially powerful properties if it can be implemented within a distributed system. Group selection approaches offer the potential for a general design framework in which individual agents or peers, in distributed systems, self-organize producing higher-level functions. In addition it may also be the key to understanding the emergence of complex multi-cellular life and social complexity.[1]

[1]Most papers produced within DELIS are available from the DELIS website as DELIS Technical Reports. Where this is the case references are appended with the DELIS Tech Report number in square brackets. This indicates the paper was produced within the DELIS project, not some other project.

Contents

1 Open Source Software Evolution

In this deliverable, we present a new multi-scale analysis for the time series of open-source (OS) software development activity [30]. Previous studies have modeled the dynamics of software development by measuring the time-evolution of project size [17] and workforce size [1]. The final goal of these studies is to enable realistic predictions for the resources needed to complete any software project.

There is a consistent increase trend in the demand for software these years. Unfortunately, we pay a significant cost for the growth of the software industry. Development of software is typically marked by cost overruns, late deliveries and poor user satisfaction. Unfortunately, very little is known about the laws governing software development dynamics. Indices of global development activity have been devised to assess the stability of the software system. A popular measure, software volatility, measures the frequency or number of enhancements per unit of application over a specified time frame [3]. High volatility is often associated to high maintenance costs. When software volatility exceeds some threshold, it may be more economical to rewrite the entire system from scratch instead of maintaining an aged (and unstable) software system [5, 15].

There are different factors affecting software volatility. Here, we propose a classification of endogenous and exogenous software changes. Traditional studies of software volatility have focused in (so-called) exogenous forces, which are represented by the requests made by software users. Very often, programs have evolved as their application domains change continuously. Eventually, software becomes very complex due to extensive modifications. However, there is an important source of endogenous changes because every new change can trigger a cascade of additional (or secondary) changes [7]. Indeed, software developers devote huge efforts to repair these secondary changes without introducing any software enhancement or new functionalities. Then, it is important to understand the origin of such cascades and to compute the probability of triggering a cascade with every new software modification (see fig. 1A).

Is there is any balance between endogenous and exogenous software changes? An early explanation can be found in the analysis of software volatility at short (a few minutes or hours) and long (a few months) time scales. In order to make a proper interpretation of the process, we have compared the data with a reasonable neutral model, i. e., a Langevin process on a scale-free software architecture [30]. At large time scales, our analysis of empirical data reveals the exogenous forces driving software development (i.e., user requests for modifications and enhancements). At the short time scale, the Langevin model coincides with the empirical data. This indicates an endogenous source of software volatility generated by the propagation of changes [7].The above suggests that software development combines endogenous and exogenous mechanisms depending on the time scale addressed.

1.1 Multi-Scale Analysis of Fluctuations

We have analysed the aggregated activity of software developers at different timescales, as recorded in the CVS database [30]. CVS (concurrent versioning system) is a popular electronic database that keeps a register of every software change and its author. This feature makes the CVS an invaluable source of information about software evolution and development activity. In the CVS we have not only the information about development history but also the source code files required to build the software system (i.e., our object of study). At any given time t, the system consists of $N(t)$ different source code files written in a given programmming language (i.e., C, C++, Fortran). This set of files describe a software network $G = (V, E)$ [32] , where nodes are the source code files $|V| = N$ and links denote compile-time dependencies between files (see fig. 1A). As we will see below, links indicate how change propagation takes place in the system, that is, what files are susceptible to be altered as a consequence of an exogenous change.

For each file f_i we measure the time series of file change dynamics:

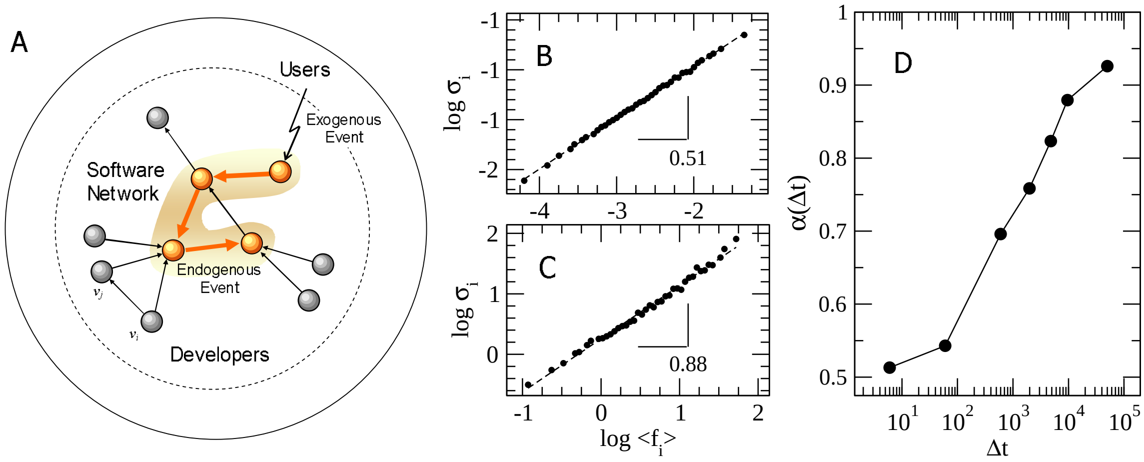

Figure 1: (A) Schematic representation of an OSS community. (B) Scaling of fluctuations with average change activity for the software project XFree86 at $\Delta t = 6$ hours (top) and $\Delta t = 9600$ hours (bottom). (D) Dependence of the observed α exponent with the measuremt window Δt.

$$f_i^{\Delta t}(t) = \sum_{\tau \in [t, t+\Delta t]} c_i(\tau) \tag{1}$$

where Δt is the measurement window and $c_i(t) = 1$ when file v_i has been changed at time t or $c_i(t) = 0$ otherwise. By variying Δt we can study the dynamics of file changes at different temporal scales. Recent measurements on the fluctuations of network nodes [19][20] indicate a power-law scaling between the average flow $\langle f_i \rangle$ and the standard deviation $\sigma_i = \sqrt{\left\langle (f_i - \langle f_i \rangle)^2 \right\rangle}$, that is,

$$\sigma_i \sim \langle f_i \rangle^{\alpha} \tag{2}$$

where α is an exponent which can take values betwen $1/2$ and 1 [19]. It seems that real systems can be classified in two different classes depending on the observed α exponent. Systems dominated by endogenous (internal) dynamics display the $\alpha = 1/2$ exponent (for example, the physical Internet or the switching activity of electronic circuits). On the other hand, systems involving human interactions or strongly influenced by exogenous forces have an $\alpha = 1$ exponent. Here, we have observed a transition in file change fluctuations from $\alpha = 1/2$ to $\alpha = 1$ as the measurement window zooms out from short time scales (hours and days) to large time scales (months and years) (see fig. 1B and 1C). Then, the multi-scale measurement window is a useful tool to detect endogenous or exogenous software development activities.

1.2 Evolution-based Software Clustering

In addition, the above separation of endogenous and exogenous changes enables us to determine subsets of files that change together (so-called "modules") [30]. Software engineering practices explicitly encourage programmers to subdivide their systems into different modules. Modularization eases software comprehension and allows software extensibility. Unfortunately, software systems loose their initial modular structure as the program evolves. Every software development reaches some critical moment when the program is so desorganized that specific re-structuring actions (i.e., refactoring) must be taken. Such refactorings allow for future changes but they are costly and require experienced software engineers. We propose a new technique that may help software engineers to identify modules easily and without previous knowledge of the system. By using the methods described here, we can analyze the software change history stored in the CVS database. Then, we can determine what files are more likely to change together and thus, what subsets of files must be grouped within the same module. Such a tool might significantly reduce the required refactoring effort.

2 Emergent Multi-Level Selection

We have previously proposed network re-wire models based on "tag" evolution to support cooperation within P2P networks [12]. More recently, we applied the same approach to structure networks into clusters of specialist nodes holding different "skills" [11]. This process could be seen as a form of higher-level selection (or group selection) which operates via emergent structures produced by individual selection. That is, the individual nodes have no notion of group utility or social benefit yet through pursuing their individual selfish behavior, imitating other nodes that get higher utility, group beneficial behavior emerges. Recently novel variants of this approach have been proposed.

2.1 Towards a general framework

We have come to view these models as particular implementations within a more general framework of group selection which can operate in networks, randomly interacting populations or other structures. We have identified four key aspects for relating these models:

- Group boundary - a mechanism which restricts interactions between agents such that the population is partitioned into groups.
- Group formation - a process which forms groups dynamically in the population.
- Migration - a process by which agents may move between different groups.
- Conditions - cost / benefit ratio of individual interactions and other conditions which are sufficient for producing group-level selection.

In on-going work we are refining and developing the framework. Ultimately this could be used as a basis or template for a set of "design patterns" for distributed systems which describe the context and conditions of application for such models [4]. In the next sections we overview a number of models of group selection within this framework.

2.2 Comparing recent models

In almost all proposed models, in order to test if group selection is stronger than individual selection, populations are composed of individuals that can take one of two kinds of social behaviour (or strategy). They can either act pro-socially, for the good of their group or they can act selfishly for their own individual benefit at the expense of the group. This captures a form of commons tragedy.

Often this is formalised as a Prisoners Dilemma (PD) or a donation game in which individuals receive fitness payoffs based on the composition of their group. In either case there is fitness cost c that a pro-social individual incurs and an associated fitness benefit b that individuals within a group gain. A group containing only pro-social individuals will lead each to gain a fitness of $b-c$. However, a group containing only selfish individuals will lead each to obtain a fitness of zero. But a selfish individual within a group of pro-socials will gain highest fitness. In this case the selfish individual will gain b but the rest will gain less than $b-c$. Given that c and b are positive then it is always in the an individuals interests (to maximise fitness) to behave selfishly. In an evolutionary scenario

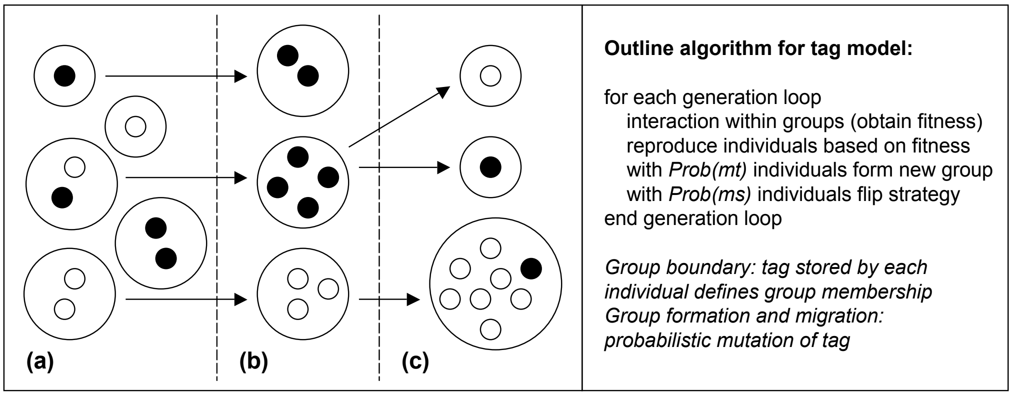

Figure 2: Schematic of the evolution of groups in the tag model. Three generations (a-c) are shown. White individuals are pro-social, black are selfish. Individuals sharing the same tag are shown clustered and bounded by large circles. Arrows indicate group linage. Migration between groups is not shown. This can occur but with very low probability.

in which the entire population interacts within a single group, which can be thought of as a mean field mixing model, then selfish behaviour will tend to be selected because this increases fitness. This ultimately leads to an entire population of selfish individuals and a suboptimal average population level fitness of zero. This is the Nash Equilibrium and an Evolutionary Stable Strategy for such a system [34].

There have been various models of cooperation and pro-social behaviour based on reciprocity using iterated strategies within the PD [2], however, we are interested in models which do not require reciprocity since these are more generally applicable. In many situations, such as large-scale human systems or distributed computer systems, such as P2P systems, repeated interactions may be rare or hard to implement due to large population sizes (of the order of millions) or cheating behaviour that allow individuals (or computer nodes) to fake new identities.

2.2.1 Tag model

In [14] a tag model of cooperation was proposed which selected for pro-social groups. It models populations of evolving agents that form groups with other agents who share an initially arbitrary tag or social marker. The tag approach was originally proposed by Holland [16] and developed by Riolo [24, 23]. The tag is often interpreted as an observable social label (e.g. style of dress, accent etc.) and can be seen as a group membership marker. It can take any mutable form in a model (e.g. integer or bitstring). The strategies of the agents evolve, as do the tags themselves. Interestingly this very simple scheme structures the population into a dynamic set of tag-groups and selects for pro-social behaviour over a wide range of conditions. Figure 2 shows a schematic diagram of tag-group evolution and an outline algorithm that generates it.

In general it was found that pro-social behaviour was selected when $b > c$ and $mt >> ms$, where mt is the mutation rate applied to the tag and ms is the mutation rate applied to the strategy. In this model groups emerge from the evolution of the tags. Group splitting is a side effect of mutation applied to a tag during reproduction. A subsequent tag model [23] produced similar results although it cannot be applied to pro-sociality in general because it does not allow for fully selfish behaviour of individuals [25, 9].

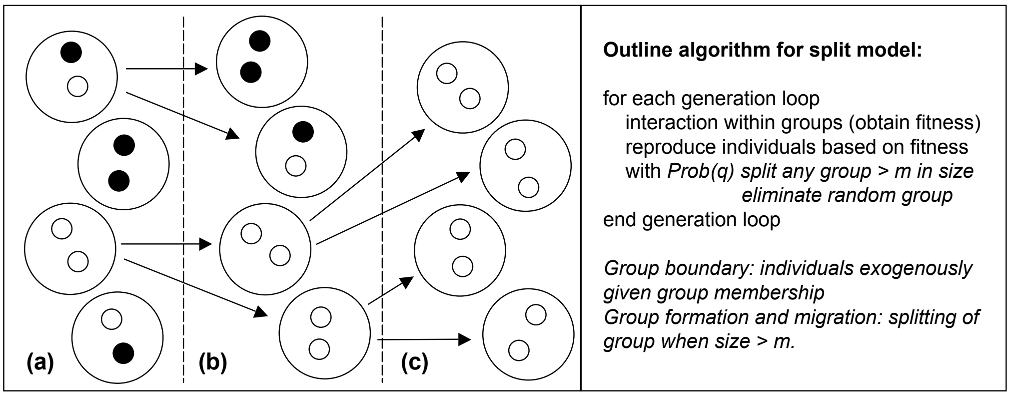

Figure 3: Schematic of the evolution of groups in the group-splitting model. Three generations (a-c) are shown. White individuals are pro-social, black are selfish. Individuals sharing the same group are shown clustered and bounded by large circles. Arrows indicate group linage. Migration between groups is not shown.

2.2.2 Group-splitting model

In [29] a group selection model is given that sustains pro-social behaviour if the population is partitioned into m groups of maximum size n so long as $b/c > 1 + n/m$. In this model group structure and group splitting is set *a priori* as exogenous parameter. Splitting is accomplished by explicitly limiting group size to n, when a group grows through reproduction beyond n it is split with (high) probability q into two groups by probabilistically reallocating each individual to one of the new groups. By endogenously controlling n and m a detailed analysis of the model was derived such that the cost / benefit condition is shown to be *necessary* rather than just sufficient. The model also allows for some migration of individuals between groups outside of the splitting process. Significantly, the group splitting model can potentially be applied recursively to give multilevel selection – groups of groups etc. However, this requires explicit splitting and reallocation mechanisms at each higher level. Figure 3 shows a schematic of group-splitting evolution and an outline algorithm that implements it.

2.2.3 Network-rewiring model

Network rewiring models for group selection have been proposed with direct application to P2P protocol design [12, 13]. In these models, which were adapted from the tag model described above, individuals are represented as nodes on a graph. Group membership is defined by the topology of the graph. Nodes directly connected are considered to be within the same group. Each node stores the links that define its neighbours. Nodes evolve by copying both the strategies and links (with probability t) of other nodes in the population with higher utility than themselves. Using this simple learning rule the topology and strategies evolve promoting pro-social behaviour and structuring the population into dynamic arrangements of disconnected clusters (where $t = 1$) or small-world topologies (where $0.5 < t < 1$). Group splitting involved nodes disconnecting from all their current neighbours and reconnecting to a single randomly chosen neighbour with low probability mt. As with the tag model we found that pro-social behaviour was selected when $b > c$ and $mt >> ms$, where ms is the probability of nodes spontaneously changing strategies. Figure 4 shows a schematic of network evolution (groups emerge as cliques within the network) and an outline algorithm that implements it.

In this model we have translated dynamics and properties similar to the tag model into a graph. This is important because P2P networks can be viewed as graphs. Hence with relatively modest

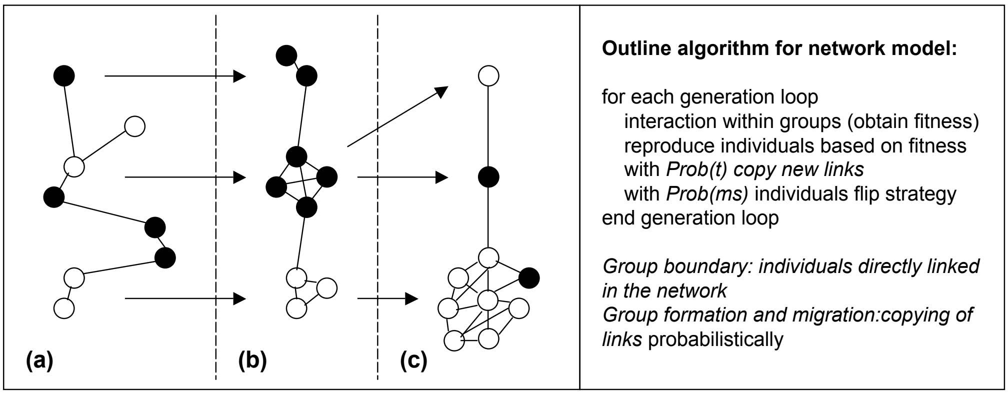

Figure 4: Schematic of the evolution of groups (cliques) in the network-rewiring model. Three generations (a-c) are shown. White individuals are pro-social, black are selfish. Arrows indicate group linage. Notice the similarity to the tag model in figure 2.

translation effort we were able to formulate a general P2P protocol that would promote pro-social behaviour over a wide rage of potential application domains. In [13] we applied the protocol to a simulated file-sharing scenario. In [12] we proposed a protocol that could be applied to collective spam filtering. In [11] the same fundamental rewiring protocol was applied to a scenario requiring nodes to adopt specialised roles or skills within their groups, not just pro-social behaviour alone, to maximise social benefit.

Interestingly it has also been shown recently [22] in a similar graph model tested over fixed topologies (e.g. small-world, random, lattice, scale-free) that under a simple evolutionary learning rule pro-social behaviour can be sustained in some limited situations if $b/c > k$, where k is the average number neighbours over all nodes (the average degree of the graph). This implies that if certain topologies can be imposed then pro-social behaviour can be sustained without rewiring of the topology dynamically. Although analysis of this model is at an early stage it would appear that groups form via clusters of pro-social strategies forming and migrating over the graph via nodes learning from neighbours.

Also a recent network model shows similar properties [26].

2.2.4 SkillWorld and ResourceWorld

We adapted the network rewiring model to P2P-like scenarios in the SkillWorld (SW) model [11] and variant ResourceWorld (RW) [18]. In these models we did not *only* consider pro-social and selfish behavioral variants but allocated "skills" to nodes such that the clusters or groups that gave best performance required both pro-social behavior and a range of skills - i.e. the clusters of nodes needed to be internally specialized. Nodes received jobs requiring a certain skill to be completed. If they did not posses the required skill they needed to rely on neighbor nodes to complete the job for them. In SW jobs completed by neighbors produced a cost for them but benefitted the node that originally received the job. In RW this was reversed, the node that completed the job always received the benefit with the receiving node paying the cost. For RW we found specialized and pro-social groups formed when $b \geq c$. Yet in SW we found $b/c \geq 2/3$. In both cases we only considered conditions where $t = 1$ and $mt >> ms$. The SW result is an interesting departure from the previous cost / benefit ratios. Essentially, in SW, pro-social behavior (altruism) can be selected even if the actual benefit to the recipient is less than the cost - up to a third less. We derived these results empirically via simulation and it is currently still unclear to us why this latter condition holds. More analysis is required, but this appears an intriguing result because it implies a kind of "super altruism" which does not maximize collective utility, but rather, may increase collective equality. Equality, even at a collective cost, could be selected if this improves group stability over time. However, as stated, this needs to be investigated further.

2.3 Summary

By relating group selection models within a general framework we can compare their relative conditions and mechanisms of application. We are beginning to formulate such a framework, currently at a modest stage. For the purposes of evolving P2P we have developed a number of network rewiring models which were derived from, and hence can be closely compared with, previous tag models. We have shown how the novel group-splitting model, relates within this framework. It would be of interest to explore the possibility of developing a distributed network implementation of the group-splitting model and test it in P2P like scenarios. One of the benefits of this model is a proven *necessary condition* for pro-social behavior selection. However, a possible disadvantage is the requirement for exogenously defined group boundaries and the requirement for whole group extinction. Implementing these in a distributed network would require novel approaches.

3 Conclusion

The analysis and engineering of multi-scale phenomena in networks appears fundamental to relating forms and functions in all but the simplest of networks. Much previous work on networks has focused on either individual node properties and behaviors (micro structure) or global properties of the entire network (macro structure). Yet, to understand the relationship between form and function in complex evolving networks, with emergent function, requires theories that relate micro and macro levels. In this sense we need to understand meso-scale processes - those that lay between, and produce, levels of organization. The on-going work presented in this deliverable follows this approach.

References

[1] Abdel-Hamid, T. K. and Madnick, S. E., (1989) Lessons Learned from Modeling the Dynamics of Software Development, *Comm. of the ACM*, 32(12), pp. 1426-1455.

[2] Axelrod, R. (1984). *The evolution of cooperation.* Basic Books, NY.

[3] Banker, R. D., and Slaughter, S. A. (2000) The moderating effects of structure on volatility and complexity in software enhancement, *Inf. Syst. Res.*, 11(3), 219-240.

[4] Babaoglu, O., Canright, G., Deutsch, A., Di Caro, G., Ducatelle, F., Gambardella, L., Ganguly, N., Jelasity, M., Montemanni, R., Montresor, A., and Urnes, T. (2006) Design Patterns from Biology for Distributed Computing. In *ACM Transactions on Autonomous and Adaptive Systems, vol. 1, no. 1, 26–66.*

[5] Chan, T., Chung S. L., and Ho, T. H. (1996) An Economic Model to Estimate Software Rewriting and Replacement Times, *IEEE Trans. Soft. Eng.*, 22(8), 580-598.

[6] Challet, D. and Le Du, Y. (2003) Closed source versus open source in a model of software bug dynamics, preprint cond-mat/0306511.

[7] Challet, D. and Lombardoni, A. (2004) Bug propagation and debugging in asymmetric software structures, *Physical Review E* 70, 046109.

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

[9] Edmonds, B. and Hales, D. (2003) Replication, Replication and Replication - Some Hard Lessons from Model Alignment. Special Issue on Model-2-Model Comparison, *Journal of Artificial Societies and Social Simulation* vol. 6, no. 4.

[10] Freeman, P. and Hart, D. (2004) A science of design for software-intensive systems, *Communications of the ACM (CACM)*, 47, 8, pp. 19-2.

[11] Hales, D. (2006) Emergent Group-Level Selection in a Peer-to-Peer Network. *Complexus 2006;3.: 108-118.* [DELIS-TR-0200]

[12] Hales, D. and Arteconi, S. (2006) SLACER: A Self-Organizing Protocol for Coordination in P2P Networks. *IEEE Intelligent Systems* 21(2):29-35. [DELIS-TR-0368]

[13] Hales, D. and Edmonds, B. (2005) Applying a socially-inspired technique (tags) to improve cooperation in P2P Networks. *IEEE Transactions in Systems, Man and Cybernetics - Part A: Systems and Humans, 35(3):385-395.* [DELIS-TR-0111]

[14] Hales, D. (2000) Cooperation without Space or Memory: Tags, Groups and the Prisoner's Dilemma. In *Moss, S., Davidsson, P. (Eds.) Multi-Agent-Based Simulation. Lecture Notes in Artificial Intelligence 1979.* Berlin: Springer-Verlag.

[15] Heales, J. (2002) A model of factors affecting an information system's change in state, *J. Softw. Maint. Evol.: Res. Pract.*, 14, (2002), 409-427.

[16] Holland, J. (1993) The Effect of Lables (Tags) on Social Interactions. *SFI Working Paper 93-10-064. Santa Fe Institute.*

[17] Kemerer, C. F., and Slaughter, S. (1999) An Empirical Approach to Studying Software Evolution, *IEEE Trans. Software Eng.* 25, 4, pp. 493-509 (1999).

[18] Marcozzi, A., Hales, D. (2006) Emergent Social Rationality in a Peer-to-Peer System. *Technical Report UBLCS-2006-23, University of Bologna, Dept. of Computer Science.* [DELIS-TR-0372]

[19] de Menezes, M. A. Barabási, A.-L., *Phys. Rev. Lett.* 92, (2004), 28701.

[20] de Menezes, M. A. Barabási, A.-L., *Phys. Rev. Lett.* 93, (2004), 68701.

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

[22] Ohtsuki, H. et al. (2006). A simple rule for the evolution of cooperation on graphs and social networks. *Nature* 441(25):502-505.

[23] Riolo, R.; Cohen, M.; Axelrod, R. (2001) Evolution of cooperation without reciprocity. *Nature 414, pp. 441-443.*

[24] Riolo, R. (1997) The Effects of Tag-Mediated Selection of Partners in Evolving Populations Playing the Iterated Prisoner's Dilemma. *Santa Fe Institute Working Paper 97-02-016. Santa Fe, NM.*

[25] Roberts, G.; Sherratt, T. N. (2002) Commentary on Riolo et al. *Nature* 418, 449-500.

[26] Santos F. C., Pacheco J. M., Lenaerts T. (2006) Cooperation prevails when individuals adjust their social ties. *PLoS Comput Biol 2(10): e140. DOI: 10.1371/journal.pcbi.0020140*

[27] Solé, R., Ferrer-Cancho, R., Montoya, J. M., and Valverde, S. (2002) Selection, Tinkering and Emergence in Complex Networks, *Complexity*, 8(1).

[28] Solé, R. and Valverde, S, (2004) Information Theory of Complex Networks, in *Networks: Structure, Dynamics and Function, Lecture Notes in Physics*, Springer-Verlag (2004)

[29] Traulsen, A.; Nowak, M. A. (2006). Evolution of cooperation by multilevel selection. *Proceedings of the National Academy of Sciences* 130(29):10952-10955.

[30] Valverde, S. (2006) Crossover from Endogenous to Exogenous Activity in Open-Source Software Development, accepted for publication in Europhysics Letters. [DELIS-TR-0432]

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

[32] Valverde, S., Ferrer-Cancho, R. and Solé, R. (2002) Scale-Free Networks from Optimal Design, *Europhysics Letters* 60, pp. 512-517.

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

[34] Wilson, E. O. (1975) *Sociobiology: The New Synthesis*. Harvard University Press, Cambridge Mass.

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