Project Number 001907

DELIS

Dynamically Evolving, Large-scale Information Systems

Integrated Project

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

Deliverable D4.3.2

Initial Tools for Network Evolution, Characterisation and Design

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 2005
Actual submission date:	January 2006
Dissemination level:	PU – public
Work Package 4.3:	Models and Methods for Dynamics, Evolution and Self-Organisation
Participants:	Universita di Bologna (UniBO), Italy Computer Science Institute (CTI), Patras, Greece University of Cyprus (UCY), Cyprus RWTH Aachen (RWTH), Germany

Authors of deliverable:	David Hales (hales@cs.unibo.it) Ozalp Babaoglu (babaoglu@cs.unibo.it) Stefano Arteconi (arteconi@cs.unibo.it) Edoardo Mollona (mollona@cs.unibo.it) Giovanni Rossi (giovanni.rossi6@unibo.it) Gian Paolo Jesi (jesi@cs.unibo.it) Spyros Kontogiannis (kontog@cti.gr) Paul Spirakis (spirakis@cti.gr) Berthold Vöcking (voecking@cs.rwth-aachen.de)

Abstract

This report comprises the complete D4.3.2 deliverable as specified for workpackage WP4.3 in Subproject SP4 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. One way of approaching this problem is to understand and harness self-organisation through evolutionary dynamics within networks. In order to begin this process new tools for both simulation and analysis are required. In this deliverable we report on the initial tools we have developed in the form of simulation models and analysis methods and some results obtained from their application. We also discuss on-going work in bringing together the evolutionary approach and the analytical approach applied to dynamic networks. We are particularly interested in the role that coalition models may offer for characterising and understanding novel network evolutionary algorithms.

Contents

1 Introduction

As discussed previously in DELIS deliverable D4.3.1, the aim of this workpackage is to model and understand dynamical evolving network systems, often out of equilibrium. We attempt to bring together two strands of enquiry: simulation based modelling and analytical game theoretic analysis. In order to progress this aim both simulation and analytical tools need to be developed and applied.

For the simulation side of the work we have further developed and applied a computational simulation tool called "Peersim" [31] applying it to a number of abstracted evolving network scenarios. Although Peersim was developed for peer-to-peer (P2P) simulation work, it is applicable to generic evolving network scenarios. Peersim is an open source publicly available tool orginally produced to support work within the BISON project [33]. We have implemented addtional protocols for the DELIS related work. Where appropriate these protocol implementations, covering the models described here, are available from the public Peersim webpage. By making these protocols available along with the existing Peersim documentation, we have built a tool set for experimentation with the evolving networks developed within DELIS. The protocols implementing the models we describe are parameterised and easily modifiable for experimentation.

We have also developed a stand alone simulation tool "FirmWorld" which has been used to capture and experiment with employee skill-set dynamics between firms in an artificial economy. Here we used and developed further ideas and theories from organisational theory and computational economics. Although this model follows an evolutionary approach, explicit networks were not modelled.

We developed a further model "SkillWorld", in which aspects of the FirmWorld model were incorporated within an evolving P2P network simulation - this has been implemented within the Peersim tool. We have demonstrated with SkillWorld that forms of proto-institutions can form within dynamically evolving networks - peers or nodes with complementary skills form functional groups that help each other to achieve their goals.

We developed further the SLAC copy-and-rewire algorithm (as discussed in D4.3.1 - and see section 3.1.2) modifying it in order to produce cooperative and fully connected networks. The networks produced form small-world topologies in which all pairs of nodes are linked by a short cooperative path. We call the new algorithm SLACER (see section 3.4).

In a recent line of work we have begun to consider the application of the existing analysis methods from "coalition theory". It appears that, in some sense, the existing tag based copy-and-rewire approach bears comparison to a kind of stochastic evolutionary coalition formation process. Traditional approach in existing coalition work tends to exclude randomised or network related effects. We have begun to tackle this by developing a new simulation model "CoalitionWorld" which captures tag-like process but can be formally specified. We then hope to add incrementally addtional aspects and through simulation and formal analysis incorporate them. The ultimate goal would be non-trivial proofs verified by simulation results of the valuable tag-like cooperation supporting processes evidenced in SLAC, SLACER, SkillWorld and TagWorld (see section 3.5).

In the next sections we summarise the main features and results from the models and techniques discussed, we do not go into detail here but rather overview the work, where appropriate we make reference to related papers [1].

2 The Peersim System

Evaluating the performance of P2P protocols is a complex task. One of the main problems for their evaluation, is the extremely large scale that they may reach. P2P networks involving hundred of

[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.

thousands of peers (or more) are not uncommon (e.g., about 5 millions machines are reported to be connected to the Kazaa/Fasttrack [34] network). In addition P2P systems are highly dynamic environments; they are in a continuous state of flux, with new nodes joining and leaving (or crashing). These properties are very challenging to deal with. Evaluating a new protocol in a real environment, especially in its early stages of development, is not feasible. Distributed planetary-scale open platforms (e.g., Planet-Lab [35]) to develop and deploy network services are available, but these solutions do not include more than about 500 (at the time of writing) nodes. Thus, for large-scale systems, a scalable simulation platform is mandatory.

The Peersim P2P simulator [31] has been developed with the aim to deal with the previously stated issues. Its first goals are: extreme scalability and support for dynamism. It is a GPL open-source Java based software project. Peersim has proved to be a valuable tool and it is used as the default experimentation platform in the BISON project [33]. In the following, we provide a brief description of its characteristics.

2.1 Peersim Design Goals

The Peersim simulator is inspired by mainly two objectives:

- High scalability: P2P networks may be composed by millions of nodes. This result can be achieved only with a careful design of the data structures involved, trying to avoid (when possible) any overhead. But the memory footprint is not the only problem: the simulator engine must be also efficient.
- Support for dynamism: the simulator must manage nodes joining and leaving the network at any time; this feature has tightly relations with the engine memory management sub-system.

Another important requirement is the *modular* or *component* inspired architecture. Every entity in the simulation (such as protocols and the environment related objects) must be easily replaceable with similar type entities.

The Peersim extreme performances can be reached only accepting some relaxing assumptions about the simulation details. For example, the overhead introduced by the low level communication protocol stack (e.g., TCP or UDP) is not taken into account because of the huge additional memory and CPU time requirements needed to accomplish this task. Peersim works exclusively at the network overlay level of abstraction.

2.2 Peersim Architecture

As previously stated, Peersim is inspired by a modular and configurable paradigm, trying to limit any unnecessary overhead. The simulators main component is the *Configurator* entity targeted to read configuration files. A configuration file is a plain ASCII text file, composed of key-value pairs. It can also contain variables and basic mathematical expression evaluated at run time. The Configurator is the only non-interchangeable simulation component. All the other entities can be easily customized or swapped with other implementations.

In general, a Peersim simulation has the following three distinct logic elements: protocols, dynamics and observers. Each of them is implemented by a Java class specified in the configuration file. Protocols are instancies of the `Protocol` interface, while the other two entities are instancies of the `Control` interface. The network in the simulation is represented by a collection of nodes, each node can hold one or more protocols. The communication between node protocols is based on object method calls.

The dynamism is provided by the `Control` interface implementing components. They are scheduled by the engine to run periodically (fine tunable by the configuration file). Controls have the global knowledge of the system and can change any protocol aspect. Apart from being used to apply dynamic changes to the environment (such as changing internal parameters or shutting-down nodes), control components can be used as observers to collect statistic data. They can also be used to initialize other components (scheduling the initializer component in order to run only at the beginning).

Peersim has a utility class package to perform statistical computations or to provide some starting topology configuration based on well know models (such as: random-graph, lattice, BA-Graph, etc.).

Peersim offers a graph-like access to the network topologies defined by the protocols. Standard graph algorithms can be adopted to check and explore the overlay properties.

The *Simulator* engine is the component that performs the computation; it schedules the component execution according to the configuration file instructions. Currently Peersim can perform simulation according to the following execution models:

- Cycle based: at each step, all nodes are selected in a random fashion and each node protocol is invoked in turn;
- Event based: a support for concurrency is provided. A set of events (messages) are scheduled in time and node protocols are run according to the time message delivery order.

In the work presented in this deliverable the first execution model is used.

3 Simulation model overviews and linkages

A number of simulation models have been produced - all of them follow evolutionary approaches in which individual agents (or nodes in a network) try to improve their performance by copying the characteristics of others they observe in the population who appear to be performing better. Figure 1 gives an overview of the models produced and their relationships. An arrow indicates a model was used to derive (via specialisation or generalisation) another model. Top to bottom roughly traces historical development with the higher models proceeding the lower ones. In this section we will quickly overview TagWorld, NetWorld and FileWorld (since these have been previously discussed in deliverable D4.3.1) and then give brief details of SkillWorld, FirmWorld and FriendWorld. Finally we discuss on-going work with CoalitionWorld (CoalWorld) and CastWorld.

3.1 TagWorld, NetWorld and FileWorld

Previously, we developed a tag based [27] sociologically inspired model called TagWorld into a network rewiring algorithm which we applied to a peer-to-peer (P2P) file-sharing scenario called FileWorld [10].

In order to test and develop the algorithm we finally used in FileWorld we first used the Prisoner's Dilemma (PD) game as a minimal test application that captures the potential for individuals to free-ride. We found that the results we obtained from this carried over into the more specific file-sharing scenario implemented in FileWorld. Since we have discussed these three models previously in deliverable D4.3.1, we do not go into details here, however, we briefly introduce the PD game, the basic algorithm used in NetWorld and FileWorld (called SLAC) and the essential findings from the work. We do this in order that the new work discussed can be put into some context. For full details of these models see the cited papers.

3.1.1 The Prisoner's Dilemma

The single-round two player Prisoner's Dilemma (PD) captures a form of commons dilemma where cooperation would benefit both participants but there is always an incentive for for each to not cooperative (to free-ride) and get a higher individual score. Figure 2 shows the pay-off matrix for the PD and explains the dilemma.

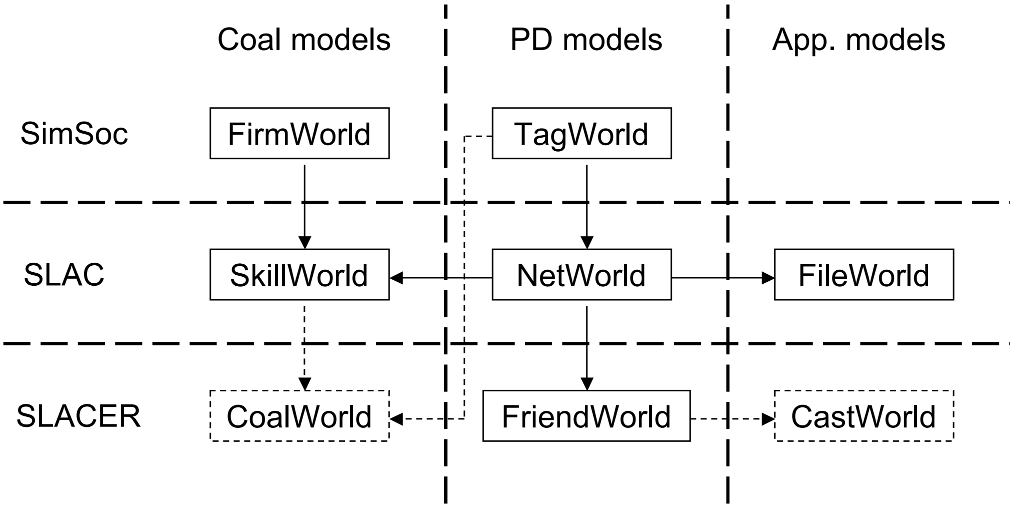


Figure 1: Evolutionary simulation models and their relationships. The models are placed in a grid indicating their task / scenario category in columns (Coalitions, Prisoner's Dilemma, Application) and the underlying algorithms in rows (Social Simulation inspired, SLAC (see section 3.1.2) and SLACER (see section 3.4). Arrows show linkages (that one model was derived from another) dotted lines on arrows and model boxes indicate work in progress, solid lines represent models covered in published work at the time of writing.

	Cooperate	Defect
Cooperate	C, C	S, T
Defect	T, S	P, P

Figure 2: A payoff matrix for the two-player single round Prisoner's Dilemma (PD) game. Given $T > C > P > S \wedge 2C > T + S$ the Nash equilibrium is for both players to select Defect but both selecting Cooperate would produce higher social and individual returns. However, if either player selects Cooperate they are exposed to Defection by their opponent - hence the dilemma

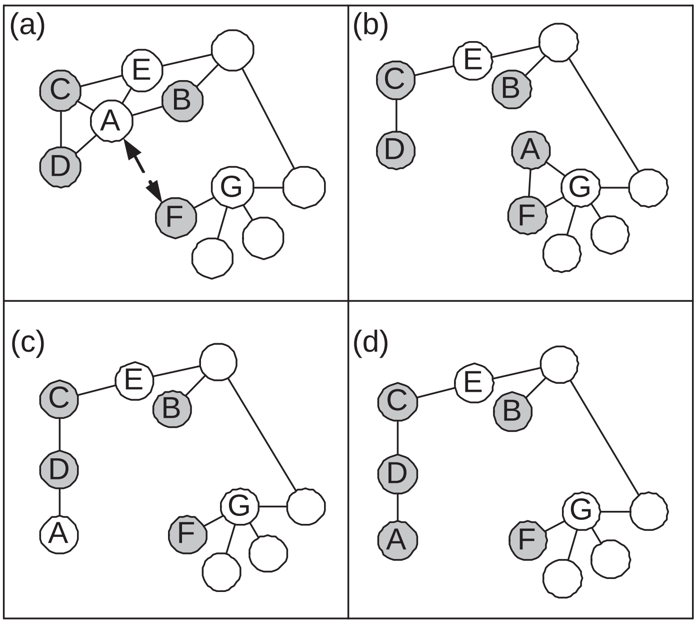


Figure 3: An illustration of "replication" and "mutation" as applied to the network (the SLAC algorithm). Dark nodes play D and light nodes play C. In (a) the arrowed link represents a comparison of utility between A and F. Assuming F has higher utility then (b) shows the state of the network after A copies Fs links and strategy and links to F. A possible result of applying mutation to As links is shown in (c) and the strategy is mutated in (d).

3.1.2 The SLAC algorithm

We found that we could produce high levels of cooperation when nodes played the PD within networks (modelled as unstructured, undirected graphs in the NetWorld model) by translating the TagWorld non-network evolutionary algorithm into a network rewiring algorithm which we call SLAC - Selfish Link Adaptation for Cooperation (this can be contrasted with more complex SLIC [28]. We do not go into details here, however, figure 3 gives an overview of the way that reproduction and mutation is applied to a network structure as a form of copy and rewire. Given the success of NetWorld in producing cooperation playing the PD we developed a further model (FileWorld) in order to test if the results from the PD carried over into a more realistic scenario. FileWorld implements a more realistic P2P query-answering scenario similar to that advanced by [28]. We found that SLAC suppressed the tendency of individual nodes to flood the system with queries [16]. In a further model we discuss later (section 3.4) we modified SLAC using probabilistic link dropping when rewiring (SLACER).

3.2 SkillWorld

In SkillWorld [13, 12] we apply the same SLAC algorithm to something different from the previous scenarios. We define an abstract scenario in which nodes, that have complementary skills may help each other to attain a better outcome. We wish to test if SLAC can produce specialisation within the formed clusters that we observed within the previous models. In those models clusters formed in which all nodes shared the same (cooperative) behaviour. In this sense all nodes within the clusters tended to behave identically - and this was sufficient to produce high levels of cooperation (both in the PD and the FileWorld file-sharing scenario).

The SkillWorld consists of a population of N nodes. Each node may have zero or more links (up to a maximum of 20) to other nodes. Links are undirected such that the entire population can be considered as an undirected graph G with each vertex being a node and each edge being a link. Each vertex (or node) is composed of three state variables a "skill type" $s \in \{1, 2, 3, 4, 5\}$, an "altruism flag" $a \in \{0, 1\}$ and a satisfaction score or "utility' $u \in R$ (where R is a positive real number).

Periodically, with uniform probability, a node i is selected from the population N. A "job" J is then generated marked with a randomly chosen skill sJ. The skill is selected, again randomly with uniform probability, from the domain $\{1, 2, 3, 4, 5\}$. Job J is then passed to node i. If node i posses the correct matching skill (i.e. if $si = sJ$) then node i may process the job itself without any help from other nodes. For successfully processing a job J the receiving node gains one unit of credit: $u \leftarrow u + 1$.

This process of generating and passing jobs to nodes represents user-level requests for services such as, for example, searching for a particular file, performing some processing task or storing some data. In the SkillWorld we dont represent the actual jobs to be done, rather, we represent the skill required to perform the job. In our minimal scenario, each job only requires one skill to be completed.

But what if node i receives a job for which it does not have the correct skill (i.e. if $si \neq sJ$) ? In this case i passes the job request to each neighbour in turn until all have been visited or one of them, j, agrees to process the job J. A neighbour j will only agree to process J if its skill matches ($sj = sJ$) and the altruism flag is set ($aj = 1$). If j does agree to process the job then this costs j a quarter unit of utility ($uj \leftarrow uj - 0.25$) yet increases the utility of i by one unit ($ui \leftarrow ui + 1$).

What this means is that node i looks for an altruistic neighbour with the correct skill to process job J. If i finds such a neighbour (j) it increases its utility as if it had completed the job itself whereas j decreases its utility. This reflects the notion that j is altruistically processing J for the benefit of i and that users are happy when jobs submitted to their nodes are completed but are not happy when jobs from other nodes use their node resources with no immediate benefit to themselves.

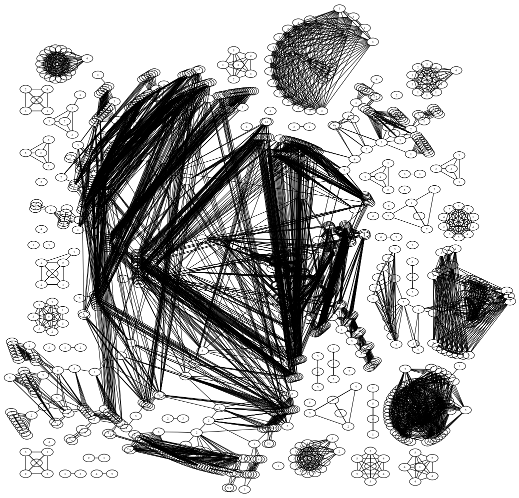
When the SLAC algorithm was applied in the SkillWorld sceanrio it was found that high levels of cooperation emerged with internally specialised and altruistic clusters or 'tribes' emerging. Figure 4 shows a typical evolution for a 1,000 node network. The output using in the figure was produced by augmenting the Peersim environment so that it could output directly network structures in the Graphviz [32] format. Graphviz is a free publicly available tool for visualisation purposes. A time series of various network measures of the same run can be seen in figure 5. These output measures used to characterise network performance are implemented as observer functions within the Peersim environment. The SkillWorld implementation is available from the Peersim website [31]. This allows for parameters to be changed, experiments run and visualisations and network statistics to be obtained for network characterisation.

The SkillWorld results were scalable - producing better performance with larger numbers of nodes (tested to over 100,000 nodes) and also robust to noise and node churn. The essential result we obtained was that the SLAC algorithm can structure populations into internally specialised clusters behaving altruistically and hence working as a team or "tribe" even though the nodes use simple greedy rules of behaviour.

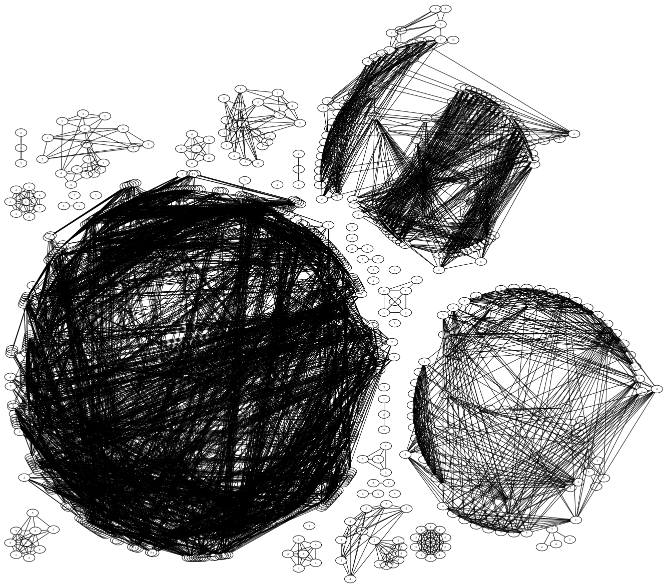
3.3 FirmWorld

The FirmWorld [23, 24] is a non-network based simulation based on ideas and theories from organisational theory and evolutionary economics. It is concerned with skill dynamics between companies or firms. Essentially, agents (representing workers) may move between container agents (representing firms) in order to get the best salary offer. Each agent has a skill (again from 1..5) which determines - depending on current market conditions - what a potential employer will offer the worker in salary. Firms store a model of the economey, which may or may not be correct, indicating what skills are believed to be valuable for attaining profits. When companies go bankrupt by losing too much money they are replaced by new "start-up" firms that preferentially copy the models of the more profitable firms. Figure 6 shows a schematic of the FirmWorld scenario.

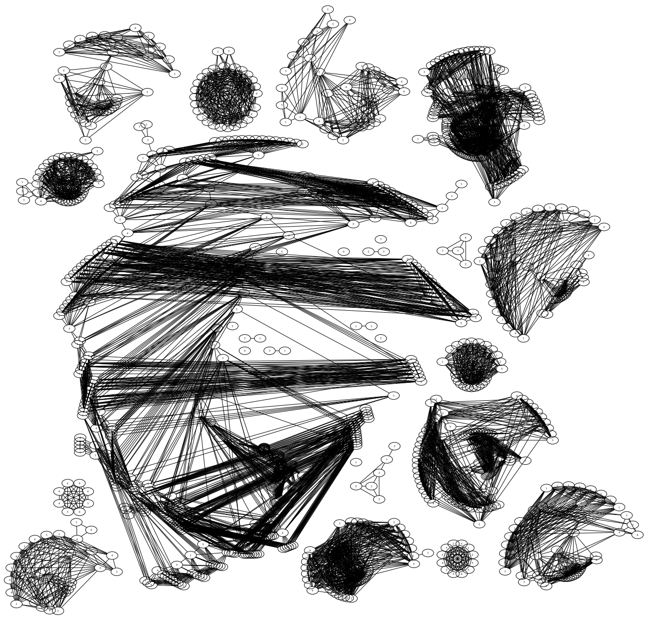
A prevalent claim is that we are in a knowledge economy. In this work, we take the view that



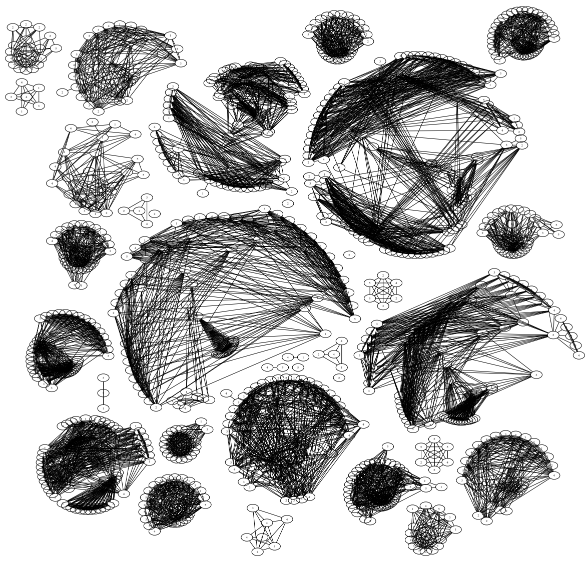
(a) Cycle 10



(b) Cycle 20



(c) Cycle 30



(d) Cycle 40

Figure 4: Evolution of clusters or 'tribes' within the SkillWorld. From an initially random topology, skill set and behaviours (not shown here) network components quickly evolve containing selfish nodes (a). Then from about cycle 20 a large cooperative component emerges in which nodes with complementary skills share jobs and increase their performance (b). By cycle 30 the large component begins to break apart as selfish nodes invade the large cooperative component and make it less desirable for cooperative nodes (c). Finally by cycle 40 an ecology of cooperative components dynamically persists as new components form and old components die (d). Note: the numbers in the nodes represent skills [1..5], the cooperative status of a node is not shown but from about cycle 20 almost all nodes are cooperative, prior to this most nodes are selfish.

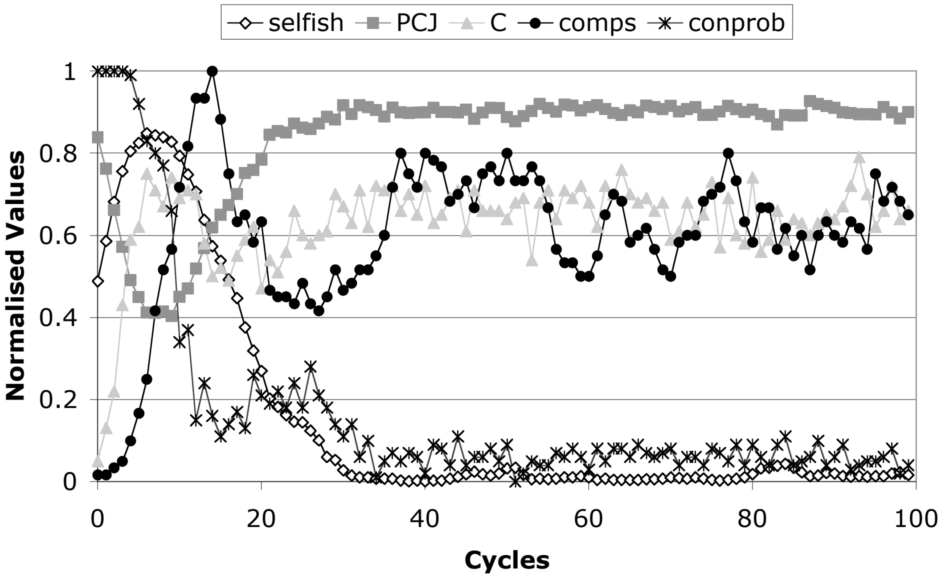


Figure 5: Time series of the run shown in figure 4 in SkillWorld ($N = 1000$). Shown are the number of selfish nodes as a proportion of the entire population (selfish), the proportion of submitted jobs that get completed (PCJ), the clustering coefficient (C), the number of components in the population (*comps*, which is normalised by dividing by 60) and the average probability that a route exists between any two nodes (*conprob*).

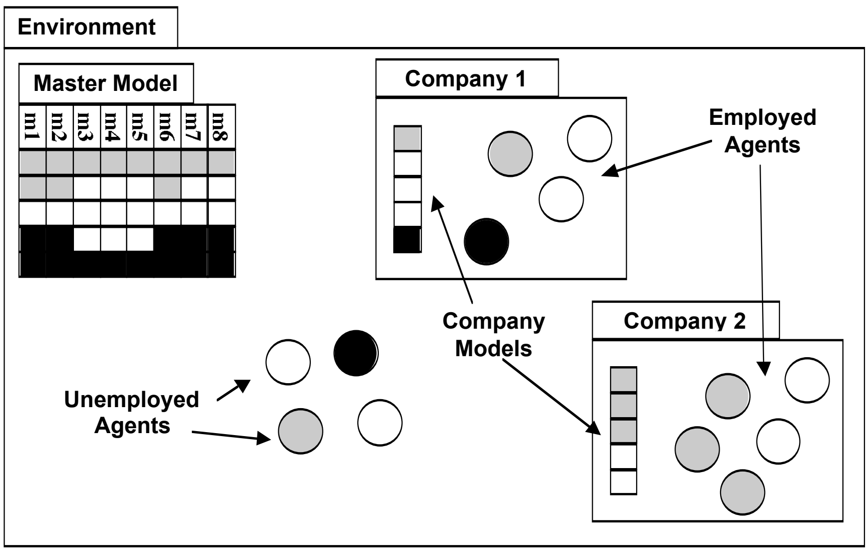


Figure 6: A schematic diagram of the main entities in the FirmWorld. The environment contains a master model giving the optimal set of employee skills for each cycle (here we only see eight cycles m1..m8 we use shades to indicate three skill types). Each firm contains a company model and some employee agents. Each firm attempts to make its workforce match its model by hiring and firing. In this case firm 2 has managed to archive this (it has 3 grey agents and two white agents) but firm 1 is one white agent short. To calculate earnings the workforce is compared to the master model for the given cycle and the distance calculated

what characterises a knowledge economy is the growing importance of human capital in productive processes [7] and the increasing knowledge intensity of jobs [17]. In particular, an influential argument is that, in a knowledge economy, the division of labour is becoming complex and firms can be viewed as networks of knowledge nodes [7], that is, sets of interacting individuals with key skills and competencies. Such networks crystallises firm-specific knowledge and provide ground upon which firms build their heterogeneity. For these reasons, an intriguing camp for research concerns the application of agent-based modelling to the analysis of firms as network of skills. The fact that the knowledge content of jobs is increasingly sparse in a network of actors raises questions concerning emerging organisational forms. Hodgson, [17], for example, suggests that managerial control on knowledge-based network of jobs decreases, especially when knowledge is tacit and cannot be codified.

This fact impairs and bounds the appliance of traditional employment contracts. Hodgson proposes that the nature of contracts evolves along with the evolution of the distribution of bargaining power. As a matter of fact, employers maintain a de iure ownership of produced goods or services and of the physical means of production but these latter have a decreasing impact in a firms value-creation processes. On the other hand, employees have got ownership on knowledge-based means of production and have and increasing control on production processes. Yet, firms maintain ownership on the mechanisms of knowledge accreditation, which increases rents extracted from knowledge-based network of jobs. For example, the brand Microsoft allows to extract rents from the jobs of many computer scientists and IBM brand allows extracting rents from the jobs of information system experts and consultants. Along similar lines, Liebeskind, [21], advises that firms have institutional capabilities that protect knowledge from expropriation and imitation thereby creating unique networks of knowledge assets. Firm-specificity is a further characteristic of knowledge-based networks of jobs that contributes to influence the evolution of employment relations. Learning processes are largely grounded upon exchange of tacit knowledge [26, 25] in groups of actors working together [1, 29]. Thus, being a part of knowledge-based networks of job require workers to invest in firm-specific learning; in exchange, workers might want security and long-term employment [17]. On the other hand, if by learning-by-doing processes, workers develop unique ways to perform tasks, the emergence of idiosyncratic jobs makes internal labour markets an efficient organisational mode [30].

From experimentation with FirmWorld we discovered a number of interesting properties. Firstly, that in highly dynamic economies, where the required skills to make high profits change often, firms offering long-term contracts tended to be more successful than those that only retained workers on a temporary basis. Essentially, it was worth firms losing money by employing many workers with unwanted skills in the present to ensure a good mix of skills for possible future changes - when desirable skills become scarce. This is because when certain skills are desirable the workers with those skills are free to negotiate among firms such that they can extract very high salaries from firms and hence reduce their profits.

In an evolutionary sense, within FirmWorld successful firms “smooth out” the shocks of a highly dynamic economy by recruiting and retaining workers with no immediate value to the firm. In this way a stable and productive organisation can be built. In order to do this firms borrow and lose money over certain periods but make it back in the long term. This bares some comparison with the so-called “Baldwin Effect” from evolutionary biology where individual, within life-time, learning can allow evolution to find optima even when search space gradients are very localised - i.e. spike or step functions in the search space. In this sense a successful firm (selected by profits as fitness at firm level) anticipates, recruits and retains.

3.4 FriendWorld and CastWorld

Although we have shown how the SLAC algorithm (section 3.1.2) can be applied to a number of scenarios it has certain limitations. As can be seen in the output from SkillWorld (figure 4) SLAC partitions the network into disconnected components. This means that the population becomes structured into fluid yet disconnected "tribes". So although there is altruistic and cooperation within the tribe, there is none between tribes. This can be considered a form of "extreem tribalism" which although not a problem for some tasks can preclude application of SLAC to domains where network-wide cooperation is required.

In order to tackle this problem we modified the SLAC algorithm such that links held by each node are only dropped probabilistically (with some high probability) as opposed to unconditionally. This means that when nodes rewire in the network they probabilistically retain some links to their old neighbours.

Essentially inspired by the biological idea of noise during copying and the social intuition that individuals within social networks rarely drop all their current contacts when moving between social groupings we speculated that this modification could keep the network connected into overlapping tribes such that chains of cooperative nodes could be found between distant nodes in the network.

We called the modified algorithm SLACER (Selfish Link Adaptation for Cooperation Excluding Rewiring) - and experimented with it in a PD scenario FriendWorld in which we experimented with different link drop probabilities [15]. We found that by reducing the probability from 1 (which equates to the SLAC algorithm when all links are dropped when a node moves) to lower values the amount of tribalism can be reduced - hence keeping the network connected but reducing cooperation. In the FriendWorld scenario - in which we looked for chains of cooperative nodes linking all pairs in the population we found that by reducing the drop probability to 0.9 a fully connected small-world network was produced in which almost all nodes were connected by chains of cooperators. The resulting networks produced by SLACER are similar to friendship networks in social systems and we have argued that such Artificial Social Networks could be a good basis for many P2P application domains.

We are in the process of applying SLACER to a simple P2P application task of broadcasting - CastWorld - where a message is required to be broadcast over the entire network requiring nodes to act altruistically by passing on the message. Using a similar method we used to modify and apply NetWorld to the FileWorld (applying SLAC to a file sharing scenario) we will adapt and apply SLACER.

3.5 Coalitions and CoalitionWorld

Interestingly, we have begun to realise that both TagWorld and FirmWorld (and perhaps all the models derived so far) may be seen as special cases of coalitions - which have been studied within formal frameworks with a substantial body of literature. The payoff functions of the coalitions (tribe or cluster in our models) depend on the specific scenario and current population structure. The individual payoffs to agents (workers or nodes) depend on the redistribution mechanism implemented within each coalition.

We realise that in our models, since agents or nodes follow an individual greedy adaptive process, to grow a successful coalition - one that continues to exist by retaining members - resources need to be distributed such that membership remains attractive to those members required to maximise the coalition level payoff. In the FirmWorld, firms (effectively managers within the coalition) were able to borrow money to modify the short term individual payoffs to grow the long-term coalition level payoffs (actual profits). In the SkillWorld altruistic job sharing between individuals helped to build the coalition (or tribe) towards an optimal structure *without* the need for centralised redistribution. This latter kind of "anarchic redistribution" also appears to be the way that all the tag-inspired models work. In the PD by selecting to cooperate, agents (or nodes) effectively choose an optimal redistribute (collective) policy. Since interactions are dyadic in these simple scenarios they grow well - as you add more cooperators the coalition continues to perform optimally without the need for centralised redistribution of utility.

Much of the formal work on coalitions involves determining the conditions under which optimal population partitions can be obtained - for given payoff and redistribution functions. The assumption then is that utility can be freely redistributed among agents within the coalition. It has been shown that certain kinds of optimal coalition formation may be NP-hard for example.

It appears little work has focused on stochastic or evolutionary approaches to coalition building. What then becomes of interest is that given a formally specified coalition scenario, what kinds of agent (or node) behaviour can lead to optimal outcomes. Can any of these be proved? Indeed for classes of evolutionary behaviour, can spaces of coalition problems be defined that they are able to solve or not?

From the point of view of P2P applications the redistribution of utility concept is interesting because it begs the question of how this could be done without existing a priori trusted accounting (e.g. money) from the bottom-up in a distributed way. It appears that the existing tag-inspired models allow this to occur since agents tend to act altruistically (redistributing resources unconditionally) to their tribe. However, this depends on the specific nature of the payoff structure coded into the task domain and more work needs to be done here to understand when the tag process works (by effective redistribution) and when it does not.

In order to begin this line of enquiry we are developing an initial simplified behavioural model "CoalitionWorld" and applied it to a formally specified coalition scenario in which the optimal outcome is known.

4 New directions in evolutionary analysis

4.1 Adaptive routing by selfish nodes

We have considered the problem of adaptive routing in networks by selfish users that lack central control. Our main focus is on simple adaption policies, or dynamics, that make use of possibly stale load information. Our analysis covers a wide class of dynamics encompassing the well-known replicator dynamics and other dynamics known from evolutionary game theory. It is a known problem, that always choosing the best option based on out of date information can lead to undesirable oscillation effects and poor overall performance.

We show that it is possible to cope with this problem, i.e., guarantee convergence towards an equilibrium state, for all of this broad class of dynamics, if the function describing the cost of an edge depending on its load is not too steep [6]. It turns out that whether or not convergence can be guaranteed depends solely on the size of a single parameter describing the greediness of the agents.

While the best response dynamics, corresponding to always choosing the best option, performs well if information is always current, it is clear that this policy fails when information is stale.

We present a dynamics which approaches the global optimal solution in networks of parallel links with linear latency functions as fast as the best response dynamics does but which does not suffer from poor performance when information is out of date.

4.2 Qualitative Difference of ESS and Nash Equilibrium Concepts

Evolutionary game theory is the study of strategic interactions among large populations of agents whose behavior evolves in time, and their decisions are based on simple, myopic rules.

A major goal of the theory is to determine broad classes of decision procedures which both provide plausible descriptions of selfish behavior and include appealing forms of aggregate behavior. For example, properties such as the correlation between strategies' growth rates and payoffs, the connection between stationary states and the well-known game theoretic notion of Nash equilibria, as well as global guarantees of convergence to equilibrium, are widely studied in the literature.

In [20] we address several recent developments in as evolutionary game theory, that give a new viewpoint to Complex Systems understanding. In particular, we discuss notions like the anarchy cost, equilibria formation, social costs and evolutionary stability. We indicate how such notions help in understanding Complex Systems behaviour when the system includes selfish, antagonistic entities of varying degrees of rationality. Our main motivation is the Internet, perhaps the most complex artifact to date, as well as large-scale systems such as the high-level P2P systems, where where the interaction is among humans, programmes and machines and centralized approaches cannot apply.

In [19] we concentrate on the notion of the Evolutionary Stable Strategies (ESS) and we demonstrate their qualitative difference from the Nash Equilibria, by showing that a random evolutionary game has on average *exponentially less* number of ESS than the number of Nash Equilibria in the underlying symmetric $2-$person game with random payoffs. In particular, although it is trivial to show that each ESS indicates a (symmetric) Nash Equilibrium for the underlying strategic game, it was not known (until the present work) whether there is a qualitative difference between the set of ESS in an evolutionary game and the set of Nash Equilibria of the underlying strategic game. Indeed, [3] gives some evidence that the notion of ESS may be of the same hardness as that of Nash Equilibria by giving exponential (worst-case) bounds on their number. On the other hand, in [19] we show that the number of ESS in a random evolutionary game is *exponentially* smaller than the number of Nash Equilibria in the underlying random symmetric 2-person game. We prove this by exploiting a quite interesting necessary and sufficient condition for a strategy being an ESS of an evolutionary game, given that the underlying symmetric strategies profile is a Nash equilibrium (this characterization was provided by [9]). Our approach is based on constructing sufficiently many (independent of each other) certificates for an arbitrary strategy (such that the underlying profile is a symmetric Nash Equilibrium) being an ESS, for which the joint probability of being true is very small.

This is the first time that ESS are explicitly demonstrated to significantly differ from the notion of stability in classical strategic games, ie, the notion of Nash equilibria.

4.3 Studying Discrete Dynamics in Networks

In this line of research our main goal is to study simple (discrete) strategies that lead the population residing at the nodes of some network at stable states. We start from simple, discrete, one–pass algorithms for assigning players of a network congestion game to their final (equilibrium) positions. We characterize broad families of network congestion instances where this is possible. Consequently we consider the case where the players themselves form (static) coalitions, in order to achieve better payoffs (collectively, and not in an atomic basis). Finally we are interested in situations where the players keep moving along the edges of a network and clash with each other whenever they meet at the endpoints of an edge. Our goal is to study the eventual states of the system, and the affection of the network structure to it. In particular, in [8] we discuss some new algorithmic and complexity issues in coalitional and dynamic/evolutionary games, related to the understanding of modern selfish and Complex networks. This first study achieves the following goals:

(a) We examine the achievement of equilibria via natural distributed and greedy (and one–pass) approaches in networks.

(b) We present a model of a coalitional game in order to capture the anarchy cost and complexity of constructing equilibria in such situations.

(c) We propose a stochastic approach to some kinds of local interactions in networks, that can be viewed also as extensions of the classical evolutionary game theoretic setting.

5 Summary

In this deliverable we have summarised on-going work with emphasis on tools and techniques so far developed and applied. For simulation work, we have built on and extended an existing open source platform Peersim [31] which was initially developed within the complementary EU BISON project [33]. We have made the code for published models and protocols publicly available on the Peersim open source web site. New analysis methods using ideas from the replicator dynamics have been developed and applied to network routing issues [6]. Also, new methods have characterised a significant difference between Nash Equilibrium and Evolutionary Stable Strategy concepts applied to a space of two player games [19] which has important implications for analysis of evolving networks. We have begun to develop both formal and complementary simulation models based on coalition theory which appears to offer a new way to combine analysis and evolutionary simulation toward characterisation of behavioural algorithms supporting socially optimal (or acceptable) outcomes in evolving networks.

References

[1] Aoki, M. (1990) The Participatory Generation of Information Rents and the Theory of the Firm. In *Aoki, M., Gustafsson, B. and Williamson, O. (eds) The Firm as a Nexus of Treaties*, London: Sage, pp. 26-51.

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

[3] Broom M. (2000) Bounds on the Number of ESSs of a Matrix Game. *Mathematical Biosciences 167(2):163–175, October 2000.*

[4] Cohen, B. (2003) Incentives Build Robustness in BitTorrent. Presented at the *1st Workshop on the Economics of Peer-2-Peer Systems*, June 5-6, 2003, Berkley, CA. Available at: http://www.sims.berkeley.edu/research/conferences/p2pecon/

[5] Edmonds, B and Hales, D. (2005) Computational Simulation as Theoretical Experiment. Journal of Mathematical Sociology 29(3):209-232 [DELIS-TR-0122]

[6] Fischer, S. and Vöcking, B. (2005) Adaptive Routing with Stale Information. In: Marcos Kawazoe Aguilera and James Aspnes (editors) *Proc. 24th Ann. ACM SIGACT-SIGOPS Symp. on Principles of Distributed Computing (PODC)*, July 2005. [DELIS-TR-0173]

[7] Foss, N. (2005) Strategy, Economic Organization, and the Knowledge Economy. *The coordination of firms and resources*, Oxford University Press.

[8] Fotakis D., Kontogiannis S., Panagopoulou P., Raptopoulos C., Spirakis P. (to appear) Algorithmic Issues in Coalitional and Dynamic Network Games. *6th International Heinz Nixdorf Symposium – New Trends in Parallel and Distributed Computing (HNI 2006)*, Lecture Notes in Computer Science, Springer. [DELIS-TR-????]

[9] Haigh J. (1975) Game Theory and Evolution, *Advances in Applied Probability. (7), pp. 8–11, 1975.*

[10] 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]

[11] Hales, D. and Patarin, S. (2005) Computational Sociology for Systems "In the Wild": The Case of BitTorrent. IEEE Distributed Systems Online, vol. 6, no. 7, 2005. [DELIS-TR-0204]

[12] Hales, D. (2005) Emergent Group-Level Selection in a Peer-to-Peer Network. Proceedings of the 2nd European Conference on Complex Systems, Paris, Nov. 2005. [DELIS-TR-0200]

[13] Hales, D. (in press) Choose Your Tribe! - Evolution at the Next Level in a Peer-to-Peer Network. Presented at the 3rd Workshop on Engineering Self-Organising Applications (EOSA 2005) located with the AAMAS 2005 conference, July 26th, 2005, Utrect, Netherlands. [DELIS-TR-0200]

[14] Hales, D.; Arteconi, S.; Babaoglu, O. (2005) SLACER: randomness to cooperation in peer-to-peer networks. In *Proceedings of the 1st International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Workshop on Stochasticity in Distributed Systems (STODIS'05), IEEE Computer Society Press. [DELIS-TR-0119]

[15] Hales, D. and Arteconi, S. (2005) Friends for Free: Self-Organizing Artificial Social Networks for Trust and Cooperation. Submitted to IEEE Intelligent Systems Special Issue on Self-management through self-organization in information systems. Available: http://arxiv.org/abs/cs.MA/0509037. [DELIS-TR-0196]

[16] Hales, D. (2004) From selfish nodes to cooperative networks — emergent link based incentives in peer-to-peer networks. In *Proc. of the 4th IEEE International Conference on Peer-to-Peer Computing (P2P2004)*. IEEE Computer Soc. Press. [DELIS-TR-0111]

[17] Hodgson, G. M. (1999) *Economics and Utopia*, London, Routledge.

[18] Jelasity, M.; Montresor, A.; Babaoglu, O. (2005) Gossip-based aggregation in large dynamic networks. *ACM Trans. Comput. Syst., 23(1):219-252.*

[19] Kontogiannis S., Spirakis P. (2005) Counting Stable Strategies in Random Evolutionary Games. *16th Annual International Symposium on Algorithms and Computation (ISAAC'05)*, Lecture Notes in Computer Science (LNCS 3827), pp. 839–848, Springer. [DELIS-TR-0176]

[20] Kontogiannis S., Spirakis P. (2005) The contribution of game theory to complex systems. *10th Panhellenic Conference of Informatics*, Lecture Notes in Computer Science (LNCS 3746), pp. 101–111, Springer. [DELIS-TR-0179]

[21] Liebeskind, J. (1996) Knowledge, Strategy, and the Theory of the Firm. *Strategic Management Journal, Vol. 17, Winter Special Issue: 93-107.*

[22] Marcozzi, A.; Hales, D.; Jesi, G.; Arteconi, S.; Babaoglu, O. (2005) Tag-Based Cooperation in Peer-to-Peer Networks with Newscast. Proceedings of the Self-Organisation and Adaptation of Multi-agent and Grid Systems (SOAS05) Conference, Dec. 2005. [DELIS-TR-0198]

[23] Mollona, E. and Hales, D. (2005) Knowledge-Based Jobs and the Boundaries of Firms. Accepted for publication in the Journal of Computational Economics. University of Bologna, Dept. of Computer Science Tech. paper UBLCS-2005-14. [DELIS-TR-0230]

[24] Mollona, E. and Hales, D. (2005) Modeling Firm Skill-Set Dynamics as a Complex System. Proceedings of the 2nd European Conference on Complex Systems, Paris, Nov. 2005. [DELIS-TR-0230]

[25] Nelson, R.R. and S. G. Winter (1982) *An Evolutionary Theory of Economic Change*, The Belknap Press of Harvard University Press, Cambridge, Mass.

[26] Polanyi, M. (1962), *Personal Knowledge*, Chicago, IL: University of Chicago Press.

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

[28] Qixiang Sun and Garcia-Molina. (2004) SLIC: A Selfish Link-based Incentive Mechanism for Unstructured Peer-to-Peer Networks. *Proceedings of the 24th IEEE international Conference on Distributed Systems*. IEEE computer Society.

[29] Teece, D. J. and G. Pisano (1994) The Dynamic Capabilities of Firms: An Introduction. *Industrial and Corporate Change, 3(3): 537-536.*

[30] Williamson, O., Watcher, M. L. and J. E. Harris (1975) Understanding the Employment Relation: The Analysis of Idiosyncratic Exchange. *The Bell Journal of Economics, 6(1): 250-278.*

[31] Peersim Peer-to-Peer Simulator, http://peersim.sf.net

[32] Graphviz graph visualisation software, http://www.graphviz.org/

[33] The BISON Project, http://www.cs.unibo.it/bison

[34] Kazaa Web Site, http://www.kazaa.com

[35] PlanetLab Planetary-Scale Testbed, http://www.planet-lab.org