POLSKO-JAPOŃSKA WYŻSZA SZKOŁA TECHNIK KOMPUTEROWYCH

Oskar Jarczyk
Department of Social Informatics
Polish-Japanese Institute of Information Technology, 2013

Agent-organized evolution of virtual teams in multi-dimensional social networks

Abstract

Multi – agent systems consists of a complex networks of agents located in a discrete cyber space. We try to use multi-agent systems to simulate the process of emerging virtual teams of open source developers. Such process of *emergent* exists in real life, in web portal like SourceFourge or GitHub, were developers join teams according to social and personal motivations. They interact based on environmental knowledge and communication constraints. Success of an open source developers teams depends on many characteristics described in various literature. This research will focus on the distribution of characteristics of emerged teams and the time periods of team evolution. Teams are virtual because they rather exist in mentioned internet portals and their members are registered users of those portals. We want to find effective network structures and propose several strategies of team formation.

Categories and Subject Descriptors
I.2.11 [Distributed Artificial Intelligence]: Multi-agent Systems;

General terms

Simulations

Keywords

Multi – agent simulations, multi dimensional social networks, OSS (open source software), team emergence, dynamic team formation, OSSD (open source software development)

1.1

Introduction

1.2

Including both virtual and classically defined teams, organizations recognized as being successful usually consists of teams which are highly dependent upon a structure that fosters productive and efficient behaviour at both the individual and the organizational levels. Thus in many applications of multi-agent systems, groups of agents must coordinate effectively in order to solve problems, allocate tasks across a distributed organization, collectively distribute knowledge and information, and achieve collective and personal goals.

1.3

The organizational structure of a multi-agent system dictates the interactions among the agents, and can play a significant role in the overall performance of the agent network (Gaston, 2005). An OSSD cooperation model in which programmers develop a free and open software is a promising way of creating a high quality computer software (Raymond, 2001). In this simulation of OSSD society we want to allows individual agents to organize their local network connectivity and adaptation mechanisms. Hahn *et.al* (2005) proved that prior collaborative ties have a profound effect on developers' project joining decisions. The results of their research suggest that it is not only the perceived expected benefits of joining the project that is salient. In addition to such perceived benefits, developers seem to be concerned about the process of realizing those potential benefits. For instance, most of the motivations, as prescribed by the prior literature, seem to focus on extrinsic and intrinsic benefits. Extrinsic benefits are benefits derived from the outcome (*e.g.*, development of software specific to one's needs, increase in reputation after participating in a successful software project, learning effects *etc.*); whereas intrinsic benefits relate to those that are attainable by virtue of participation in the project itself (e.g., enjoyment, affiliation, community identification etc.).

1.4

[right here something about matrix of skills]

1.5

The role of prior collaborative ties seems to be related to reducing uncertainties in the process of project participation. In fact, software development is not only a production process but also a social process that heavily involves inter-personal communication and coordination (Curtis *et al.* 1988, Robey and Newman 1996, Sawyer *et al.* 1997, Sawyer and Guinan 1998). Moreover, in the OSSD context, the difficulty inherent in the social process becomes even greater than in traditional software projects since members of an OSSD project are typically from geographically dispersed locations, have diverse cultural backgrounds, and have limited (if not any) face-to-face interactions.

1.6

Related work

1.7

Gaston *et al.* [1] studied the team formation problem in a setting where networked agents form teams in a decentralized manner. Agents are able to locally rewire their social network. Anagnostopoulos *et.al.* (2012) considered a setting in which people possess different skills and compatibility among potential team members is modelled by a social network. Authors focused on proposing first online algorithms that assemble teams to deal with tasks. *Balanced Social Task Assignment* problem was introduced with online competitive algorithms. Scholars analysed previous work connected with team formation, i.e.: scheduling with load balancing (Graham *et.al.* 1972) (scheduling of jobs on a set of machines with the goal of minimizing the maximum load on a machine), matching people to tasks (a matching problem for which several systems have recently been proposed, for instance easychair.org, linklings.com and softconf.com).[3] The problem of providing efficient solutions in the bidding model has been addressed by Mehlhorn (2009). Team formation with coordination costs: Lappas et al. (2009) introduce the problem of team formation in social networks. The objective is to minimize the coordination cost, for example, in terms of diameter or weight of the minimum spanning tree for the team. This problem has been extended to

cases in which potentially more than one member possessing each skill is required, and where density-based measures are used as objectives [8,9].
1.8
Scholars Jin, Girvan, Newman in their paper "The structure of growing social networks" (2001) proposed two models of growth of social networks which have properties of fixed number of vertices, limited degree, clustering and decay of friendship. Simulation of those models shown emerging societies and existing clustering. Researchers made a conclusion that complex and reasonable patterns of social networks, and evolution of them, can emerge from simple rules, furthermore, general form of those patterns is not influenced by micro details of the rules.

1.9
Problem definition
2.1
During research we will try to simulate evolution (emerging) virtual teams in social networks. Portals for OSS group work (mentioned SourceForge and GitHub, but also BitBucket and similar) are multi-dimensional social networks because of their interaction between users features, i.e. friends list and discussion boards. They are also social constructs which can be described as COINs (Collaborative Innovation Networks). Users are mostly programmers, developers, designers and experts in the field of software and computer technology. A user and team have dynamic characteristics, which are changing over time. A user is moving forward his collaboration path while he gains experience and knowledge, and a team evolves within its characteristics, structure, needs for new and different users.

2.2
There are several problems which we consider to evaluate:
2.3
- Distribution of characteristics of created teams: sizes and comparing required competencies with competencies of a team

It is deemed it's expected to draw a hierarchical tree or dendrogram showing emerged team structures. Moreover, we plan to introduce clustering coefficient as function of time for different parameters. Because of a dynamic aspect of team formation, virtual team emerging is affected by streams of events.

2.4
- Time of the process: of team emergence from the starting point to the state of being a complete team by the definition of required competences

The additional purpose of this dissertation is to find recommendations for improving team emergence process. Evaluating and supporting the team formation can lead to better team performance, software quality of OSS created by those virtual teams, and finally learning process of team members.

2.5
- Learning rate: How quickly does a chosen strategy increases performance

Performance can be changed over time as a result of local network adaptation. Number of adaptation changes over time proportionally to changes in structure (Gaston, figure no 3).

2.6
- Stability: Does the strategy lead to a stable network topology

It can be disputed if stability is beneficial for organizational performance (Gaston, p. 1 pnt. 2). A power law distribution is observed in structure changes of open source software teams evolution (Wu, 2006).

2.7

- Global structure: What are the properties of structure of created teams

There is interest in analysing centrality/decentrality of teams. While there is no leadership included in our model, still its possible to analyse web of acquaintance inside teams.

2.8

Problem of completeness

A complete virtual team have a state of equilibrium which allows for the most efficient work, moreover, further changes in the team structure won't be as dynamic and intensive as they were in previous stages of emergence of this particular team. Capability maturity model created by Software Engineering Institute (SEI) defines 5 stages of *software maturity* (which starts at stage 3). Zhang (2007) analysis Open Source Software maturity model based on linear regression and bayesian analysis.

2.9

Competencies

competency	*k1*	*k{1+i..j-1}*	*kj*
man-hour	*x1*	*x2*	*x3*

Table 1: Sample model of competency matrix, stricte competencies required by a team

3.1

Agent-organized network (AON)

3.2

There are several considerations for AON modelling:

- Perception of team performance
  How agent assesses the collective performance of a virtual team?
- Perception of team completeness
  When a team is in a complete state ?
- Adaptation triggers
- Rewiring

People. We consider a set of OSS contributors *P = {pj ; j = 1, 2, . . . , n}*. Each person has a subset of competencies (skills) in her profile, so she is also represented by a point in the competency space: pj 2 S. We use pji = 1 to denote that the jth person has the ith competency, while pji = 0 otherwise. Thus, we have pj = (pj1, pj2, . . . , pjm).

Teams. Each task needs to be assigned to a team of experts. We let *Qj # P* denote the team assigned to the jth task. We use *qj i = 1* to denote that the ith skill is covered by the jth team, so we have qj =

(qj 1, qj 2, . . . , qj m). For each team Qj we compute its team profile qj 2 S in the additive skill model [2] that defines the expertise of the team as the (binary) sum of the competencies of each individual:

$$\mathrm{q}_i^j = \min\{ \sum_{\mathbf{p}^\ell \in Q^j} \mathrm{p}_i^\ell, 1\}_{i=1,\ldots,m}.$$

Agent is a computer programmer who searches for a team which he can join. With every tick of time, an agent can decide to:

- join a team
- leave a team
- work in a team
- improve skills, experience, modify acquaintance list
- stay iddle

Agents have characteristics:
- time of joining the society;
- vector of interests, experience

Agents are part of multi-dimension social network. Agents can be classified with labels characterizing their type of commitment to team work, i.e.: student, expert, leader, follower, innovator. Those labels are part of psychological theories and management theories, which describe team work problems. Agents are getting into acquaintance with other agents, therefore making for possibility of inviting other agents to their teams.

Team formation strategies

In social and collaboration networks virtual teams are emerged, characterized by their name, set of vectors of demanded competence, their members and the size of a team. Multiple teams are emerging and they are attracting multiple agents to join them. Agents join teams basing on below general classification of heuristics:

1) prefential attachement
2) homophyly/heterophyly
3) by invitation

Agents join the teams for particular period of *x* time, and during this time, they maybe be busy with some activity, or stay idle. A variety of factors may affect the choice of project team to join. Both critical mass theory and expectancy value theories suggest that developers would be influenced by project size since size is a highly visible indicator of the probability of successful outcome and value of the group (Karau and Williams 2000, Markus 1987). However these theories do not explain the decision of developers who choose to join newly formed projects.

Continual addition

Continual addition of vertices and edges to the network as time passes. (Jin, 2001)

Structure-based adaptation strategy

In this strategy, the agents adapt their network connectivity based on the notion of *preferential attachment*.

Performance-based adaptation strategy

AON strategy based on performance and referals was analyzed and proposed by Gaston *et.al.* During every time tick, agent decides if to rewire when 1) at least one of agents neighbours have valid performance measure 2) when it's performing below the average of neighbour's performance measure.

Homophyly *versus* heterophyly strategy

Expert recommendation strategy

Simulation parameters defined in *parameters.xml* file with their interpretation are described in table 1.

Simulation parameters for Repast Simphony model	
numNodes	Number of start nodes, which is a number of persons – agents.
numTeams	Start number of teams.
percStartMembership	Which percentage of people is already assigned to a team.
allowMultiMembership	Can a person be a member of more than one team.
avgStartMembership	In average, to how many teams a person will be assigned in the start of a simulation.
numSteps	Number of steps which the simulation will last.

Table 2: AON repast simphony 2.0 simulation parameters

References

[1] "Agent-Organized Networks for Dynamic Team Formation" Matthew E. Gaston Marie desJardins 2005
[2] "Emergence of New Project Teams from Open Source Software Developer Networks: Impact of Prior Collaboration Ties" Jungpil Hahn
[3] "Online Team Formation in Social Networks", Aris Anagnostopoulos 2012
[4] "Learning and Predicting Dynamic Behavior with Graphical Multiagent Models" Quang

Duongy Michael P. Wellmany Satinder Singhy Michael Kearns
[5] R. L. Graham. Bounds on multiprocessing anomalies and related packing algorithms. In AFIPS, 1972.
[6] K. Mehlhorn. Assigning papers to referees. In ICALP, 2009.
[7] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In KDD, 2009.
[8] A. Gajewar and A. D. Sarma. Multi-skill Collaborative Teams based on Densest Subgraphs. In SDM, 2012.
[9] C.-T. Li and M.-K. Shan. Team Formation for Generalized Tasks in Expertise Social Networks. In SocialCom, 2010.
[10] Raymond, E. S. 2001. The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary. O'Reilly and Associates, Sebastapol, CA.
[11] The Structure of Growing Social Networks, Emily M. Jin, Michele Girvan, M.E.J. Newman, Santa Fe Institute 2001
[12] Open Source Software Evolution and Its Dynamics, Jingwei Wu, 2006