Evolving Relationships between Social Networks and Stakeholder Involvement in Software Projects

Soo Ling Lim
Department of Computer Science
University College London
Gower Street, London, WC1E 6BT

s.lim@cs.ucl.ac.uk

Peter J. Bentley
Department of Computer Science
University College London
Gower Street, London, WC1E 6BT

p.bentley@cs.ucl.ac.uk

ABSTRACT

Software projects often fail because stakeholder communication and involvement are inadequate. This paper proposes a novel method to understand project social networks and their corresponding stakeholder involvement. The method uses five types of model social network, which represent various types of stakeholder activity in a project. It exploits evolutionary computation to correlate the social network of a real software project against each model. Experiments show that the real project most resembles the "rational" model where stakeholders who are more highly connected in the social network are more involved in the project.

Track: Search-Based Software Engineering (SBSE)

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications

General Terms

Design, Experimentation, Human Factors.

Keywords

Search-based software engineering, Social network analysis, Stakeholder analysis.

1. INTRODUCTION

Building software systems is a social activity. Customers pay for the software system, users interact with the system to get their work done, developers build and maintain the system, and legislators impose rules on the development and operation of the system [1, 2]. All these diverse groups of stakeholders must communicate effectively in appropriate social networks for the project to be successful [3-5].

When software projects fail (which is extremely common [6]), the reasons are often stakeholder related. One common reason is that the social network that connects the stakeholders together is malformed (i.e., there is poor communication among stakeholders). Another common reason is insufficient stakeholder involve-

ment in the project [7, 8]. In fact, one of the biggest concerns that developers have with project stakeholders is that some stakeholders lack commitment or skill to be adequately involved [7]. Identifying these stakeholders in advance enables other more suitable stakeholders to be sought. It is therefore of great interest to understand and predict likely problems with project social networks and corresponding stakeholder involvement at the start of the project, so that potentially disastrous problems can be mitigated.

This work addresses this issue, and provides a method for analysing project social networks and the likely corresponding future involvement of the constituent stakeholders. This novel method involves the creation of five types of model social network, which represent various types of stakeholder activity in a project. It uses evolutionary computation to enable the social network of a real software project to be correlated against each model and understood in terms of the stakeholder activity they represent. Using predictive models such as these, it is anticipated that issues relating to communication and involvement of stakeholders can be identified at the start of software projects before they detrimentally affect the outcomes.

The paper is organised as follows. Section 2 introduces the background for this work. Section 3 describes the modelling method. Section 4 describes the use of the evolutionary algorithm and its fitness function. Section 5 describes the experimental setup. Section 6 discusses the results and Section 7 concludes.

2. BACKGROUND

2.1 Software Project Stakeholders

In requirements engineering, the traditional approach for the identification of stakeholders in a new software project is to use semi-structured approaches [8, 9]. Stakeholders are identified by considering categories such as those who interact directly with the system and those with interests. Alternatively, some projects use checklist-based approaches that map generic stakeholder roles (e.g., regulator and maintenance operator) to project specific roles [7]. In both approaches for stakeholder identification, the categories are broad, and stakeholders are likely to be omitted [10]. In both approaches the social networks of the stakeholders are implicit and form during meetings and everyday communication.

Recent research in requirements engineering propose a more explicit use of social networks to identify and prioritise stakeholders. At the start of a project, the stakeholders are asked to recommend other stakeholders, and their recommendations are used to build a social network, with the stakeholders as nodes and their recommendations as links. StakeNet is an example of such a method [10]. In StakeNet, the social network of stakeholders consists of people who recommend other stakeholders or are recommended as stakeholders [10]. (To support StakeNet, Lim et al.

also developed a software tool, which was used in more than 10 projects for stakeholder analysis [11, 12].) The work described in this paper uses the social network data gathered by this approach.

2.2 Modelling

Models provide a useful method for prediction and analysis and are used in this work. Here, we regard a model as the representation of a hypothesis underlying an explicit real world phenomenon [13]. The model's hypothesis specifies the components and interactions thought to be sufficient to generate the desired behaviour of the real world phenomenon. Thus, by implementing the model, we are able to view the consequences of the assumptions underlying the hypothesis [13, 14]. If the behaviour of the model is similar to that of the target, then it is reasonable to assume that the model's assumptions are in fact true [13]. Nevertheless, the correctness of the assumptions is not guaranteed – a model that correctly generates the desired behaviour may still erroneously explain the target behaviour for one reason or another. If the model does not generate the target behaviour, then it is assumed that the underlying assumptions are not enough to generate the target.

Ideally, the model implements the hypothesis and nothing besides, so that the generated behaviours can in fact only be attributed to the hypothesis [13]. However, the process of implementation normally requires elaborations or simplifications of the hypothesis for it to be tractable, and these may not have an underlying theoretical justification, and so, the actual model is likely to contain some elements that are not a part of the hypothesis [13].

2.3 Search-Based Software Engineering

The work described here makes use of evolutionary computation in order to learn the correlation between model social networks and a social network of a real software project. It is not the first use of evolutionary computation in software engineering.

Search-based software engineering (SBSE) is a field which applies search-based optimisation techniques to address software engineering problems [15]. A wide range of optimisation techniques has been used, such as local search, simulated annealing, genetic algorithms, and genetic programming, as reviewed in [15]. These techniques have been applied to various software engineering activities such as requirements analysis, project management, software refactoring, test data generation, and bug fixing, as reviewed in [15, 16]. A common characteristic in these works is that perfect solutions are impractical or impossible in many software engineering problems [15]. As such, metaheuristic search–based optimization techniques are used to search for good solutions. The search is guided by a fitness function, which serves to differentiate good solutions from poor ones. This shifts the emphasis from solution construction to solution description, and the researcher defines what is required rather than how the solution should be constructed. The search-based methods are then left to generate creative solutions for the defined fitness function.

This paper uses the evolutionary computation known as Cartesian Genetic Programming (CGP) [17]. CGP[1] is a form of Genetic Programming in which a program is represented as an indexed graph [17]. It is Cartesian in the sense that it considers a grid of nodes that are addressed in a Cartesian coordinate system. The graph is directed and encoded in the form of a linear string of integers, with the inputs or terminal set, node outputs, and node functions numbered separately. The genotype, which is a list of

---

[1] CGP code is available at:

https://sites.google.com/site/julianfrancismiller/professional

node connections and functions, is mapped to an indexed graph that can be executed as a program. CGP implements the neutral search strategy, which replaces the fittest genotype by another equally fit (or fitter) genotype. The representation in CGP is simple, flexible and convenient for many problems [18].

CGP was originally developed by Miller and Thomson to evolve digital circuits [17]. Since then it has been applied to an increasing number of domains such as tumour classification, evolutionary art, control systems, image processing, and bioinformatics [18]. This work is one of the first to apply CGP on social relationships. Previous studies have shown CGP to be efficient and more evolvable in comparison with other GP techniques [18], and for these reasons it was chosen for the work described in this paper.

3. MODELLING STAKEHOLDERS

This work aims to understand and predict the characteristics of software project social networks and corresponding stakeholder involvement. The work is based on the assumption that there is a relationship between a specific type of social network and certain patterns of future stakeholder involvement in the project. For example, a "rational" social network might mean that stakeholders that are more connected to other stakeholders are likely to be more involved in the project (i.e., a stakeholder with good communication with other stakeholders will be more involved compared to those with worse communication). In contrast an "incorrect" social network might mean that there is no relationship between the number of connections and involvement (i.e., even if a stakeholder has communication with many others, this provides no indication about the likely involvement of that stakeholder).

It is the hypothesis of this work that the relationship between social network and involvement can be learned, and that learned relationship can then be used to understand the nature of new software projects. For example, suppose the relationship between the "rational" social network and its stakeholder involvement is described by equation R1, and the relationship between the "incorrect" social network and its stakeholder involvement is described by equation R2. We then apply both equations to the social network of a new project and have two predictions for its likely stakeholder involvement (Figure 1). Comparing the data, we may then discover that the second prediction seems to be better: R2 describes its stakeholder involvement more accurately compared to R1.
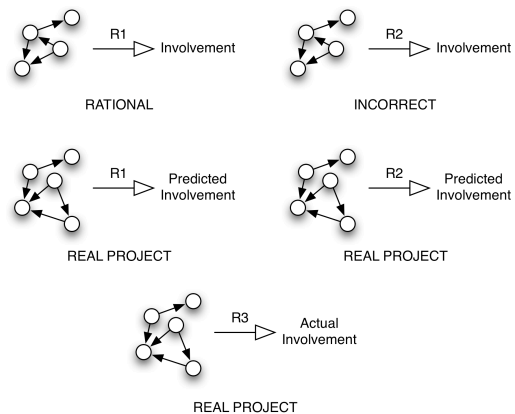

Figure 1. Learning relationships R1 and R2 and applying to the real project produces two predictions of involvement for the real project. Comparing the actual involvement with the predictions allows the real project to be classified as more similar to the rational or incorrect model.

We are now able to conclude that the new project's social network may share important properties of the "incorrect" network, and if this is not desirable, then measures can be taken to alter the communication and involvement of stakeholders within the new project and correct likely problems before the project is detrimentally affected.

To validate the finding, a reverse correlation can also be performed. The relationship between the social network and involvement of the new project could be learned (R3) and then applied to the "rational" and "incorrect" social networks. If R3 provides a better match for stakeholder involvement of "incorrect" compared to "rational" then again we can conclude that the new project resembles the "incorrect" model.

The method described in this paper uses five types of model project which represent various types of stakeholder activity in a project (these are not intended to be an exhaustive list of all possible types of activity). To enable comparison with real world data, social networks of projects are comprised of stakeholders and their recommendations, as created by the StakeNet system described in Section 2.1. While networks of recommendations do not always imply similar networks of communication, the data is deemed sufficiently representative for the purposes of this work. CGP described in Section 2.3 is used to learn the various relationships between the social networks (described by different social network measures for each network – see Section 3.3) and stakeholder involvement.

3.1 Model Project

The base model project was constructed based on case studies about large software projects and their stakeholder engagement experiences [8, 12, 19, 20].

We model a situation in which a fictional project is initiated to build a software system to be used throughout organisation X (for example, an expense claims system). Organisation X is large, comprising multiple departments. Coordination and cooperation from other departments are essential for the success of this project. The new system depends on data from other systems in the organisation. The employees of organisation X will be the users of the new system, which means all the departments need to be involved in requirements elicitation.
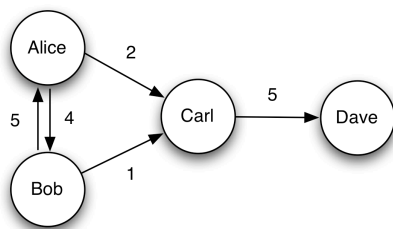

Figure 2. Example stakeholder network [10].

We model the situation at the start of the project, where the stakeholders recommend other people whom they think should be involved in the project. Each recommendation is in the form of <stakeholder name, stakeholder role, rating>. The rating is the extent of the stakeholders' importance to the project. The recommendations are used to build a social network of stakeholders. Figure 2 illustrates an example stakeholder network where Alice recommends Bob with a rating of 4 and Carl with a rating of 2, Bob recommends Alice with a rating of 5 and Carl with a rating of 1, and Carl recommends Dave with a rating of 5.

The stakeholder network for the model project consists of 50 stakeholders (Figure 3). Negative stakeholders are stakeholders who resist the new system. Non-stakeholders are mistakenly recommended as stakeholders. Data providers are stakeholders who manage the systems that the new system takes data from. Legislators impose business rules on the expense claims process and also data protection.
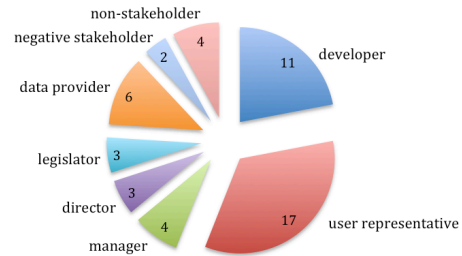

Figure 3. Types of stakeholders (total 50 stakeholders).

The stakeholders' actual involvement in the project is calculated using the method proposed in the existing literature [7, 10] as follows: a stakeholder's overall involvement in a project is the sum of their involvement in (1) financing the project, (2) making decisions about the development of the system, (3) developing the system, (4) imposing rules on the development or operation of the system, (5) using the system or its output, or (6) threatening the success of the system. The following steps were used to calculate a stakeholder's involvement [10, 12].

- For each generic involvement (1 to 6), specific types of involvement are listed that are relevant to the project. For example, in the model project, stakeholders who finance the project include those who finance the human resources, and those who finance the hardware and software. For using the system or its output, specific involvement includes using the system to claim expenses, and using the system to approve expenses.

- Each stakeholder is rated High, Medium, or Low on their specific involvement. For financial involvement, a stakeholder who pays more has a higher rating. For management, a stakeholder who is more accountable has a higher rating. In development, a stakeholder with more responsibilities has a higher rating. For usage, a stakeholder who uses the system more frequently, and is more affected by the system, has a higher rating. For example, in the model project, Dave was responsible to finance human resources (rated High), and used the system occasionally to claim expenses (rated Medium).

- The ratings are converted into numerical values (High = 3, Medium = 2, and Low = 1). The overall involvement of a stakeholder is the sum of their ratings. For example, Dave's overall involvement is 3 + 2 = 5.

The model stakeholder involvement graph forms a power law [21, 22], whereby a few stakeholders are heavily involved in the project, and many stakeholders are lightly involved in the project (Figure 4). Only 60% of the modelled stakeholders in the stakeholder network were actually involved in the project. This means that 20 stakeholders had an involvement value of 0. These characteristics were based on previous studies of stakeholder networks by Lim et al. [10].

Using this base model, we construct five "worlds" that the project could be in. Stakeholder recommendations differ in each

world, resulting in different connectivity in the network. All the networks comprise the same 50 stakeholders previously described. Only their recommendations or their involvement differ. The worlds are: *rational*, *enthusiastic*, *incorrect*, *perfect*, and *random*. Each world and its characteristics are described below.
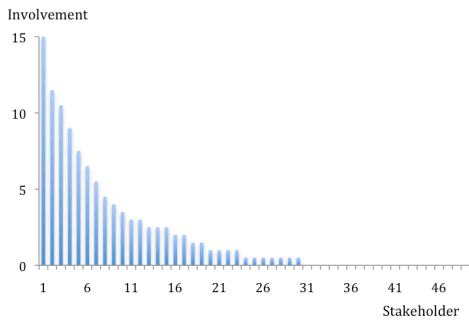

Figure 4. Power law graph for stakeholder involvement.

3.1.1 Rational

In the *rational* world, stakeholders recommend the stakeholders that they think will be involved in the project. In addition, all the stakeholders who made recommendations were involved in the project. The rational world occurs if the stakeholders are busy and will only take their time to make recommendations if they have sufficient stake in the project to be involved in the future.

The *rational* network follows the recommendation characteristics of a real-world project [12] as follows. The number of recommendations received by the stakeholders forms a power law graph whereby a few people receive many recommendations, and many people receive a few recommendations. Only half of the stakeholders in the network made recommendations. In this world, all the stakeholders who were not involved in the project also did not make recommendations. To model busy individuals, five stakeholders who were involved did not make recommendations.

For stakeholders who recommend, the number of recommendations ranged between 1 and 13, and the average recommendation is 6.9. Stakeholders do not recommend themselves.

Stakeholders with more connectivity (high number of incoming recommendations with high values and non-zero number of outgoing recommendations) tend to have higher involvement.

The *rational* network is illustrated in Figure 5(a).

3.1.2 Enthusiastic

In the *enthusiastic* world, all the stakeholders made recommendations, regardless of whether they were involved in the project. This model occurs if stakeholders are altruistic, even if they are not interested in the project, they recommend other stakeholders who are more suitable than them.

The total number of recommendations is twice the rational model. As such the network also has twice the number of edges (Figure 5(b)). Similar to the rational model, the number of recommendations received by the stakeholders forms a power law graph.

In this model, higher incoming recommendations (number and value) imply more involvement, but outgoing recommendations do not reflect involvement.

3.1.3 Incorrect

In the *incorrect* world, the stakeholders recommend stakeholders who are not necessarily involved in the project. Those who are recommended highly are not necessarily highly involved in the

project. The incorrect network occurs if stakeholders are unable to make good recommendations, i.e., those they recommend highly are not necessarily highly involved in the project. This could happen if the stakeholders lack knowledge, they are malicious, or there is really no connection between recommendations and involvement.

In the *incorrect* network, the connections are non-random, but there is no relationship between network connectivity and involvement. To create an *incorrect* network, the rational network (Figure 5(a)) was used but the involvement was randomised. As a result, stakeholders who are highly recommended may be mildly involved in the project, and vice versa.
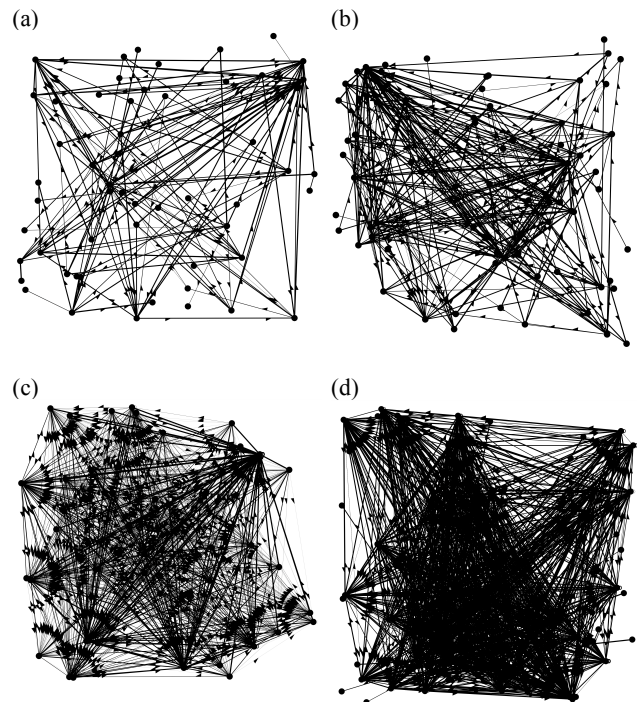

Figure 5. The model social networks: (a) rational (b) enthusiastic (c) perfect (d) random[2].

3.1.4 Perfect

In the *perfect* world, each stakeholder's recommendations are the actual list of stakeholders and their actual involvement in the project. The rating is the stakeholders' actual involvement scaled to the maximum of 8. The *perfect* world occurs if stakeholders know exactly who the other stakeholders are and the extent their project involvement will be, and recommend accurately. This network is unlikely in the real world and is included for experimental purposes.

The *perfect* network is fully connected. The number of incoming and outgoing connections do not relate to the involvement, as each stakeholder has the same number of incoming and outgoing recommendations. The *perfect* stakeholder network is illustrated in Figure 5(c).

3.1.5 Random

In the *random* world, the stakeholders make recommendations at random. This network occurs if all stakeholders provide random

---

[2] The incorrect network is the same as the rational network. The networks were visualised using Gephi (http://gephi.org/).

recommendations. This network is unlikely in the real world and is included for experimental purposes.

The *random* recommendations were created as follows. Each stakeholder recommends all stakeholders, and the rating is a random number between 0 and 8. A rating of 0 means the stakeholder is not recommended. The connections are random, and there is no relationship between involvement and connectivity (Figure 5(d)).

3.2 Real World Project

This work assesses the ability of CGP to learn relationships from the five models and make predictions about involvement on a real project. The real world project used here is RALIC.

RALIC stands for Replacement Access, Library and ID Card. It was a large-scale software project to replace the existing access control system at University College London and consolidate the new system with library access and borrowing [12]. The project duration was 2.5 years, and the system was deployed in 2007. RALIC has a large and complex stakeholder base with more than 60 stakeholder groups. Besides students, staff and visitors who use the system, the stakeholders include faculty and academic departments, as well as administrative divisions such as the estates and facilities division, human resource division that manages staff information, library systems, security systems, and so on. The success of the project relied on the coordination and involvement of all the stakeholders [12].

The stakeholder recommendations for the RALIC project[3] were collected in previous work by Lim et al. [12]. In that work, the RALIC stakeholders were invited for a face-to-face interview to collect their recommendations (Figure 6).

Figure 6. Recommendations from a RALIC stakeholder [12].

The stakeholder network has 127 stakeholders. The ratio of stakeholder breakdown (Figure 7(a)) is similar to the base model project. About half of the stakeholders made recommendations.

The stakeholder network (Figure 7(b)) is very dense due to the high number of nodes. In addition, stakeholders were allowed to recommend only the roles, and role recommendation is connected to all the stakeholders with the recommended roles, causing stakeholders who recommend roles to seem like they recommended many stakeholders. Stakeholders who made recommendations made an average of 25 recommendations each. There were a total of 1714 recommendations.

The stakeholder involvement was built by Lim et al. [10, 12] using the method described in the previous section.

3.3 Datasets

To produce the datasets for CGP, social network measures were applied to each model and real stakeholder network[4]. The social network measures prioritise each stakeholder S as follows.

---

[3] RALIC recommendation and involvement dataset is available at: http://www.cs.ucl.ac.uk/staff/S.Lim/phd/dataset.html

[4] This work uses the social network measures implemented in NetworkX (http://networkx.lanl.gov/).
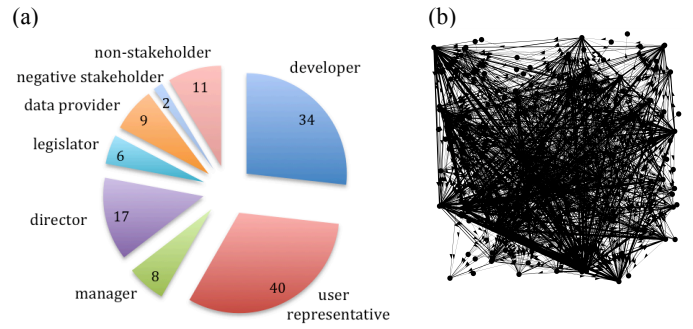
(a)   (b)

Figure 7. RALIC project (a) types of stakeholders (b) social network.

- **Betweenness centrality** sums the number of shortest paths between pairs of stakeholders that pass through S [23].

- **Load centrality** sums the amount of information passing through S [24].

- **Closeness centrality** sums the inverse average shortest-path distance from S to all reachable stakeholders [25].

- **PageRank** ranks S in terms of its relative importance to all other stakeholders [26]. This measure is recursive in that a stakeholder who is strongly recommended by many highly ranked stakeholders are ranked higher, and the recommendations of a highly ranked stakeholder are given more weight, which, in turn, makes their recommended stakeholders rank high.

- **Degree centrality** ranks S based on the number of incoming and outgoing recommendations S has [25]. Stakeholders who have a high number of direct connections with others are ranked higher.

- **In-degree centrality** ranks S based on the number of stakeholders who recommend S [25]. Stakeholders who are recommended by many stakeholders are ranked higher.

- **Out-degree centrality** ranks S based on the number of recommendation S makes [25]. Stakeholders who recommend many stakeholders are ranked higher.

4. EVOLVING INVOLVEMENT

We use CGP to predict the involvement of a stakeholder based on the connections in the stakeholder network. The closer the predicted involvement is to the actual involvement, the better the solution. As such, the fitness for each stakeholder's involvement, i, is calculated as follows:

$$Fitness_i = \frac{1}{1 + Error_i^2}, \qquad \textbf{(Equation 1)}$$

where $Error_i = |\ predicted\_involvement_i - actual\_involvement_i\ |$. *Error* is squared to increase the penalty for larger errors.

The overall fitness is the sum of the fitness for all the stakeholders' involvement. Using this fitness function, the closer the result is to the total number of stakeholders, the better the fitness. As such, the maximum fitness for the model dataset is 50; the maximum fitness for the real dataset is 127.

In the real-world data from RALIC, about 50% of the stakeholders in the network had 0 involvement. This means that a naïve function which always outputs zero or close to zero will receive a

high fitness value. This is confirmed in our preliminary experiments, where the naïve function produced good fitness of about 70%, but the predicted involvement was always between 0 and 1, with no distinction between stakeholders with high and low involvement.

To differentiate between high and low involvement, the training data was sorted into order, with highest involvement values presented first, and an incremental fitness function was employed. This approach only presents the next training item(s) if the average fitness for the first item(s) is equal to or better than 0.5, otherwise a score of zero is given for the remaining items. This ensures evolution focuses first on the more important training items, and only progresses to the next once a satisfactory solution has been found. This enables larger involvement values to be predicted accurately (as this indicates important stakeholders), before the lower involvement values are tackled.

Although the actual involvement is always non-negative, initial experiments showed that this fitness produces negative predicted involvement for some stakeholders. Initial experiments were conducted to investigate two constraint handling approaches described by Yu and Bentley [27] to impose non-negative predicted involvement.

- **Legal map (correct illegal phenotypes)**. If a predicted involvement is less than zero, the predicted involvement is corrected to become 0. This constraint provided CGP with too much freedom, resulting in the predicted involvement to be highly inaccurate.

- **Phenotype penalty (penalise illegal phenotypes)**. If a predicted involvement is less than zero, the fitness for that prediction is reduced to 0. This constraint is too hard. The best solution for this constraint produced solutions twice the size compared to no constraint and legal map.

No constraints were used in the real experiments described in the next section, as this gave the best results.

5. EXPERIMENTS

The experiments investigate the ability of CGP to learn the relationship between stakeholder recommendations and their project involvement for the five model worlds. These learned relationships are then applied to the RALIC real world project in order to obtain predictions for the RALIC involvement. Those learned relationships that most accurately predict RALIC involvement will indicate which model worlds most resemble the RALIC project. In addition, a reverse test was performed. CGP was used to learn the relationship between RALIC's social network and its involvement, and this was used to predict the involvement for the five model worlds. Again, the model world(s) with involvement that most closely matches the prediction learned from RALIC will indicate which world(s) resemble RALIC the most.

The two experiments were thus:

- **Experiment 1**. Correlation of real project to model worlds. Each model world was used as the training set, and the real data was used as the test set.

- **Experiment 2**. Correlation of model worlds to real project. The real data was used as the training set, and each model world was used as the test set.

Both experiments were run for 25 times. Experiment 1 was run for 5 million generations. Experiment 2 was run for 20 million generations because the real data is larger and preliminary experiments showed that it required more generations for training. The termination criteria for the experiments were either the maximum number of generations or when evolution obtained a specific fitness value for the training data. The termination criteria were based on preliminary experiments on the training set to stop the CGP solution from overfitting the training set.

The default settings for CGP were used [18]. The population size was 5 and mutation rate was 0.5. The function set included addition, subtraction, multiplication, division, power, square root, modulus, and reciprocal.

For each dataset, the inputs to CGP were the outputs from the social network measures, as well as the constant values 1, 2 and 3. The choice of inputs to CGP was informed by preliminary experiments. In the initial experiments on the real data, other attributes were used in addition to the social network measures, such as, the total number of stakeholders in the social network, maximum possible rating, each stakeholder's time in the organisation, whether they were available during the recommendation period, and whether they knew about the survey. In this case the larger number of inputs resulted in worse evolved results. The inability of CGP to perform effective feature selection using a large number of attributes may be due to the complexity of the problem.

6. RESULTS

6.1 Correlation between Worlds and RALIC

Table 1 (top half) and Figure 8 (a) to (e) provide a summary of the results for the first experiment.

As can be seen, the *rational* world provided the best predictions of involvement for the RALIC project. Among the five worlds, it had the highest average test fitness, and the lowest standard deviation. The test fitness has an average of 57.73 (45%), however this was affected by a single unlucky run, which only achieved a fitness of 8 (16%) training and 1.29 (1%) for the test set. Examining the best evolved solution, the predicted involvement from the *rational* world resembled the RALIC involvement very closely. The predicted involvement formed a power law graph, and the top 4 stakeholders were consistent. The main differences were in the magnitude of involvement, for example, the maximum actual involvement was 31 but the maximum predicted involvement was 15 (Figure 9).

The *incorrect* and *random* worlds were difficult for CGP to learn. In both models, the training fitness had not plateaued after 5 million generations. As anticipated (given that RALIC was a successful project) the *random* world least resembled the RALIC data. It had the lowest average test fitness of 17.66 (14%). This was followed by the *incorrect* world with the second lowest average test fitness. These results suggest that stakeholders in RALIC tend to recommend correctly, and the people that they recommend highly are likely to be actively involved in the project.

The *perfect* world was the easiest for CGP to learn, requiring the least number of generations to achieve the cutoff fitness. (In most runs, CGP was able to achieve a perfect 50 or 100% in less than 5 million generations.) Although CGP achieved good average fitnesses for training and the best maximum fitness for RALIC, this model had the highest standard deviation of 26. The graph in Figure 8(d) illustrates that the accuracy for RALIC is highly inconsistent, dropping from more than 70 to less than 30, and back to more than 70 again in succession. This suggested that good solutions and bad solutions are very similar, and so predictions made using this model may be difficult to rely on. Indeed, it seems that evolution "cheated" for this world, predicting that all involvement values should very close to zero (shown in Figure 9), which is a strategy that will always gain fairly good fitness scores, as described in Section 4.

Table 1. Results

Experiment	Cutoff (F=Fitness, G=Generation)	Avg. generation to reach cutoff	Avg. train fit	Stdev. train fit	Avg. test fit	Stdev. test fit	Min. test fit	Max. test fit
1.1 Train on rational	F = 45 (90%)	977,718	43.27 (87%)	7.48	57.73 (45%)	13.85	1.29 (1%)	72.01 (57%)
1.2 Train on enthusiastic	F = 45 (90%)	1,967,407	44.84 (90%)	0.44	35.20 (28%)	15.88	9.00 (7%)	64.13 (50%)
1.3 Train on incorrect	G = 5,000,000	4,921,517	31.90 (64%)	7.83	23.48 (18%)	16.59	4.86 (4%)	64.52 (51%)
1.4 Train on perfect	F = 48 (96%)	67,808	48.32 (97%)	0.36	36.21 (29%)	26.23	0.00 (0%)	74.75 (59%)
1.5 Train on random	G = 5,000,000	4,860,891	32.92 (66%)	8.60	17.66 (14%)	19.66	0.00 (0%)	62.59 (49%)
2.1 Test on rational	G = 20,000,000	19,602,026	83.64 (66%)	27.83	21.16 (42%)	9.74	0.73 (1%)	31.68 (63%)
2.2 Test on enthusiastic	G = 20,000,000	19,602,026	83.64 (66%)	27.83	9.14 (18%)	6.06	0.02 (0%)	24.31 (49%)
2.3 Test on incorrect	G = 20,000,000	19,602,026	83.64 (66%)	27.83	13.25 (27%)	5.27	1.53 (3%)	20.63 (41%)
2.4 Test on perfect	G = 20,000,000	19,602,026	83.64 (66%)	27.83	2.39 (5%)	5.35	0.00 (0%)	19.88 (40%)
2.5 Test on random	G = 20,000,000	19,602,026	83.64 (66%)	27.83	3.98 (8%)	7.11	0.00 (0%)	26.42 (53%)

(a) *rational* world   (b) *enthusiastic* world   (c) *incorrect* world

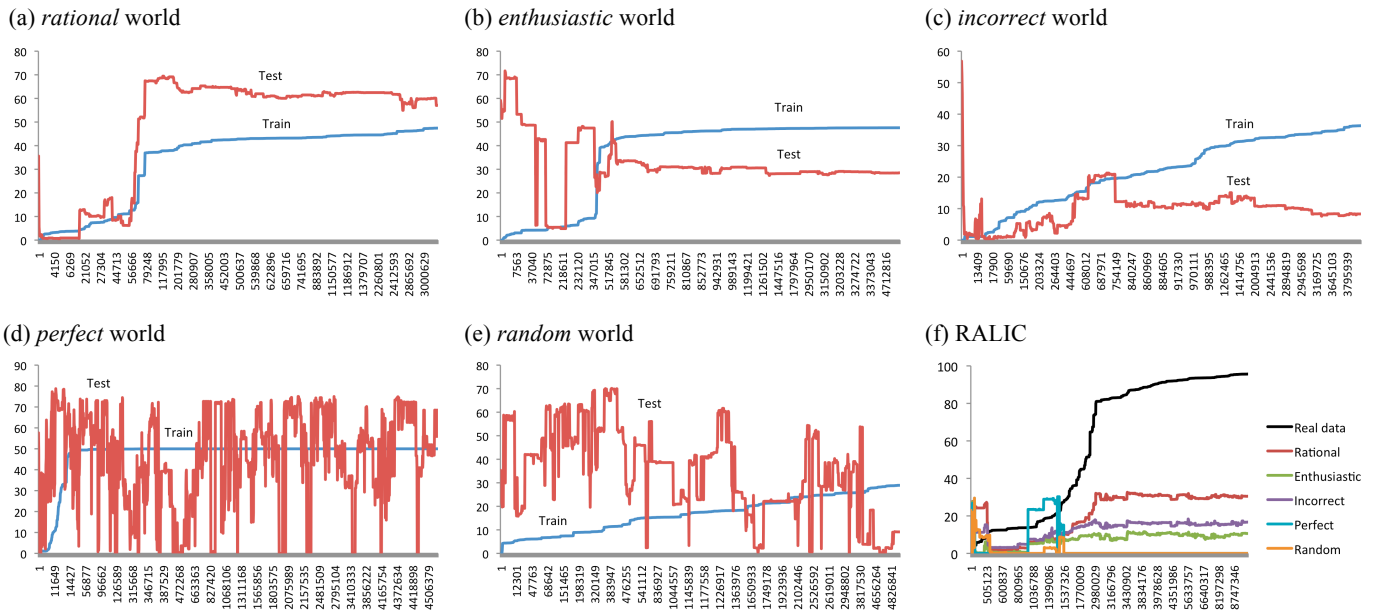(d) *perfect* world   (e) *random* world   (f) RALIC

Figure 8. Representative graphs of individual runs showing: (a) to (e) Training model fitness (blue) vs. test RALIC fitness (red). (f) Training RALIC fitness (black) vs. test model fitnesses. In all graphs, the x-axis is number of generations, y-axis is fitness. Best fitness for models = 50, best fitness for RALIC = 127.
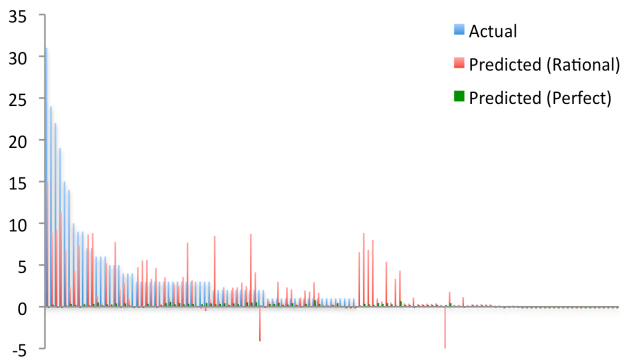

Figure 9. Actual vs. predicted involvement.

It is thus clear that the *random* and *perfect* models least resembled the RALIC project. This means that in RALIC, stakeholders who make recommendations at the start or are strongly recommended are also more likely to be involved in the project. Similarly, stakeholders who do not make recommendations are likely not to be involved in the project. Stakeholders are more likely to recommend correctly, than make mistakes in their recommendations. Their recommendations usually make sense, which means there is correlation between the people they recommend and how good those people are.

The *enthusiastic* world had a similar average test fit compared to the *perfect* model. The results suggest that this world does not match RALIC well – stakeholders who will not be involved in the project, are also likely to be less interested in the project at the start, which meant they are not likely to spend time recommending other stakeholders.

As expected, CGP evolved a different equation for each run and for each model. Analysis of all the evolved equations for each run revealed that some components are repeated in all solutions. The better solutions (giving better test fit for RALIC) are also shorter. Closeness centrality, in-degree centrality, and PageRank appeared most frequently in the equations.

6.2 Correlation of Real Project to Worlds

Table 1 (bottom half) and Figure 8(f) provide a summary of the results for the second experiment. CGP was able to learn the real data of RALIC. The training fitness was generally better than individual fitness for the social network measures. However, the problem was more difficult for evolution and 20 million generations were not always sufficient.

Again, the *rational* world achieved the highest test fit, followed by the *incorrect*, *enthusiastic*, *random* and *perfect* worlds. This suggests that in RALIC, stakeholders who made recommendations are more likely to be involved in the project in the future. Stakeholders are more likely to make mistakes rather than all recommend.

As before, the *random* and *perfect* worlds least resembled the RALIC project. Figure 8(f) illustrates the train and test fitness for the real data and models in one run. As can been seen, the evolutionary trend for the test fitness for *rational*, *incorrect* and *enthusiastic* resembles the steady improvement of the training data. In contrast the *random* and *perfect* fitnesses have good test fitnesses at the start, but soon drop to zero, showing no correlation.

7. CONCLUSIONS

Software projects frequently fail, and the cause of their failure is often the lack of communication or involvement of stakeholders. This work introduced a novel method to understand and predict characteristics of project social networks and corresponding stakeholder involvement, so that potential problems resulting from inadequate stakeholder involvement can be mitigated. The method uses five types of model social networks, which represent various types of stakeholder activity in a project. It exploited evolutionary computation to correlate the social network of a real-world software project (RALIC) against each model. Experiments showed that RALIC most resembled the *rational* world, and least resembled the *random* and *perfect* worlds, confirming that the stakeholders of RALIC were able to communicate effectively and have useful involvement in the project.

Future work will investigate the creation of models that represent specific problems in project communication or involvement. Using predictive models such as these, it is anticipated that issues relating to communication and involvement of stakeholders can be identified before they detrimentally affect software projects.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] Sharp, H., Galal, G. H., and Finkelstein, A. 1999. Stakeholder identification in the requirements engineering process. In *Proc. of the Database & Expert System Applications Workshop (DEXA)*, p. 387–391.

[2] Nuseibeh, B., and Easterbrook, S. 2000. Requirements engineering: a roadmap. In *Proc. of the Conf. on the Future of Soft. Eng.*, p. 35 - 46.

[3] Wolf, T., Schroter, A., Damian, D., and Nguyen, T. 2009. Predicting build failures using social network analysis on developer communication. In *Proc. of the 31st Int. Conf. on Soft. Eng. (ICSE)*, p. 1-11.

[4] Meneely, A., Williams, L., Snipes, W., and Osborne, J. 2008. Predicting failures with developer networks and social network analysis. In *Proc. of the 16th Int. Symp. on Foundations of Soft. Eng. (FSE)*, p. 13-23.

[5] Bird, C., Nagappan, N., Gall, H., Murphy, B., and Devanbu, P. 2009. Putting it all together: using socio-technical networks to predict failures. In *Proc. of the 20th Int. Symp. on Software Reliability Engineering*, p. 109-119.

[6] The Standish Group. 1994. *The CHAOS Report*. p. 8.

[7] Alexander, I., and Robertson, S. 2004. Understanding project sociology by modeling stakeholders. *IEEE Software*. 21(1): p. 23-27.

[8] Gause, D. C., and Weinberg, G. M. 1989. *Exploring Requirements: Quality Before Design*. Dorset House Publishing Company, Inc.

[9] Cockburn, A. 2000. *Writing Effective Use Cases*. Addison-Wesley Professional.

[10] Lim, S. L., Quercia, D., and Finkelstein, A. 2010. StakeNet: using social networks to analyse the stakeholders of large-scale software projects. In *Proc. of the 32nd Int. Conf. on Soft. Eng. (ICSE) - Vol. 1*, p. 295-304.

[11] Lim, S. L., Damian, D., and Finkelstein, A. 2011. StakeSource2.0: using social networks of stakeholders to identify and prioritise requirements. In *33rd Int. Conf. on Soft. Eng. (ICSE), in press*.

[12] Lim, S. L. 2010. *Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation*. PhD Thesis, University of New South Wales, Sydney, Australia.

[13] Webb, B. 2001. Can robots make good models of biological behaviour? *Behavioral and Brain Sciences*. 24(6): p. 1033-1050.

[14] Bullock, S. 1997. *Evolutionary Simulation Models: On their Character, and Application to Problems Concerning the Evolution of Natural Signalling Systems*. PhD Thesis, University of Sussex, Brighton, UK.

[15] Harman, M. 2007. The current state and future of search based software engineering. In *Proc. of the Future of Soft. Eng. (FoSE)*, p. 342-357.

[16] McMinn, P. 2004. Search-based software test data generation: a survey. *Software Testing, Verification and Reliability*. 14(2): p. 105-156.

[17] Miller, J., and Thompson, P. 2000. Cartesian genetic programming. In *Proc. of the 3rd European Conf. on Genetic Programming, LNCS*. Springer-Verlag, p. 121-132.

[18] Miller, J., and Harding, S. 2010. Cartesian genetic programming. In *Proc. of the 12th Annual Conf. Companion on Genetic and Evolutionary Computation*, p. 2927-2948.

[19] Potts, C., and Catledge, L. 1996. Collaborative conceptual design: a large software project case study. *Computer Supported Cooperative Work (CSCW)*. 5(4): p. 415-445.

[20] Kraut, R., and Streeter, L. 1995. Coordination in software development. *Communications of the ACM*. 38(3): p. 69-81.

[21] Anderson, C. 2008. *The Long Tail: Why the Future of Business is Selling Less of More*. Hyperion Books.

[22] Barabási, A. L., and Albert, R. 1999. Emergence of scaling in random networks. *Science*. 286(5439): p. 509-512.

[23] Brandes, U. 2001. A faster algorithm for betweenness centrality. *J. of Mathematical Sociology*. 25(2): p. 163-177.

[24] Brandes, U. 2008. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*. 30(2): p. 136-145.

[25] Scott, J. 2000. *Social Network Analysis: A Handbook*. Sage.

[26] Page, L., Brin, S., Motwani, R., and Winograd, T. 1999. *The Pagerank Citation Ranking: Bringing Order to the Web*. Stanford InfoLab.

[27] Yu, T., and Bentley, P. 1998. Methods to evolve legal phenotypes. In *Parallel Problem Solving from Nature (PPSN)*. Springer, p. 280-282.