Requirement Elicitation using Recommender Systems

Ishan Tewari
Computer Science and Engineering Department
Institute of Technology, Nirma University
Ahmedabad, India
18bce080@nirmauni.ac.in

Labdhi Sheth
Computer Science and Engineering Department
Institute of Technology, Nirma University
Ahmedabad, India
18bce101@nirmauni.ac.in

Abstract—**Requirement Elicitation is the first and the major step of Requirement Engineering. In a software project, the interaction between developers and end-users is needed by requirements gathering and thus the quality of the elicitation process can be governed by the communication between them. It includes identifying the stakeholders and prioritising their requirements. This becomes difficult for large scale project development. This paper proposes a Machine Learning-based approach for requirement elicitation. This approach uses social network and collaborative filtering techniques for eliciting requirements. This method firstly creates a social network of Stakeholders, asks them to rate an initial set of requirements and then recommends a list of requirements from like-minded Stakeholders using collaborative filtering. The approach uses the k-Nearest Neighbor (KNN) algorithm to compute the degree of similarity between the requirements, where N is the number of recommendations proposed to the user.**

Keywords— Requirement Elicitation, Requirement Analysis, Recommender System, Collaborative filtering, Social Networks, Requirement Prioritization, KNN.

I. Introduction

Requirement Elicitation is the first and the major step is Requirement Engineering. Often, problems of proper Requirement Elicitation and Analysis are faced by large-scale software systems. Also, these projects involve vast numbers of stakeholders – the individuals or groups that can influence or be influenced by the success or failure of a software project. In these projects, these stakeholders are cut across various divisions and departments and hence, they have diverse requirements which create a conflict. The three major problems faced in requirement elicitation for large scale projects are:

- Information Overload :

  Overloading of information is something which is unavoidable in large scale projects. The classical requirement elicitation methods such as intensive interactions with the stakeholders, interviews, meetings etc cease to work for projects with a large number of stakeholders, say thousands or hundreds of thousands. Inevitably, some of the stakeholders are omitted and their requirements are overlooked. In the end, the Customers who pay for the project bear the cost.

- Inadequate Stakeholder Input :

  Now as stakeholders are omitted, the problem of inadequate stakeholder input arises. Inadequate input leads to the wrong product being built.

- Biased Prioritization of requirements :

  As there is a large number of stakeholders, it becomes almost impossible to prioritize the correct stakeholders and their requirements. So, as the important requirements are known to a limited set of stakeholders, they often tend to get lost in this vast amount of information.

  To overcome these problems, this paper proposes a method which uses Social Networks and Recommender Systems which use Collaborative Filtering for requirements elicitation.

  This method addresses the problem of Information Overload by using recommender systems for recommending relevant requirements to the stakeholders, prioritizing the stakeholders and their requirements.

  It solves the problem of Inadequate Stakeholder Input by creating a social network of stakeholders and asking the stakeholders to recommend other stakeholders, and then asking each and every one of them to rate requirements and recommend other stakeholders.

  Lastly, it solves the problem for Biased Prioritization of Requirements by prioritizing the requirements based on the stakeholders' ratings and their influence in the social network.

II. Background

A. Large Scale Organizations

Generally, large-scale software projects are defined from the already existing measures of project size and definitions of large-scale software projects. Some of the popular measures of project size are lines of code (LOC), function points (FP), number of developers, man-hours, etc. But as the focus of this paper is requirements elicitation the definition measures the size of the requirements engineering tasks, rather than the size of the software system.

This paper defines a large-scale software project as a project with dozens of stakeholder groups consisting of tens of thousands of stakeholders. Also, a stakeholder group contains one or more stakeholder roles.

B. Requirement Elicitation and Prioritization

The traditional techniques for requirement elicitation include interviews and focus groups. In these interviews, the requirement engineer approaches the stakeholders and asks them about their requirements. Although these methods are effective and efficient for small scale projects, they do not scale well with large scale projects and important requirements are often let go.

This paper proposes a method which uses the built social network of stakeholders and after collecting their initial set of requirements, proposes them new ones by the help of recommender systems, which are then prioritized on the basis of the stakeholders' influence and the rating of the requirement.

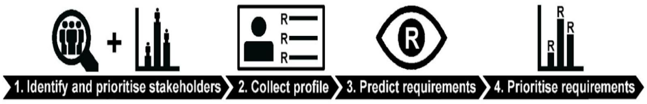
C. Recommender Systems

Recommender Systems use the technique called Collaborative Filtering which forecasts a user's preference on an item by collecting and analyzing data from like-minded users. These systems are actively used by social media websites, OTT platforms and shopping websites.

This paper proposes to use collaborative filtering to provide the stakeholders with possible apt requirements according to the stakeholder rating.

III. Proposed Method

This paper proposes a 4 step approach to solving the problem of requirement elicitation and analysis.



+
R
R
R
R
R
R
1. Identify and prioritise stakeholders
2. Collect profile
3. Predict requirements
4. Prioritise requirements

Fig. 1. Steps of the proposed method.[1]

TABLE I. Concepts

Concept	Definition
Salience	The level of influence the stakeholder has on the project. Higher the salience, higher the influence.
Scope	The work needed to complete the project successfully
Stakeholder	An individual or group of people which is influenced or influences the success or failure of some project.
Stakeholder role	The stakeholder's position in the project.
Requirement	The goals and constraints on the software system.
Rating	Numerical importance of a requirement to the stakeholder.
Profile	The set of requirements and the ratings provided by the user.

Step 1: Identify and Prioritize Stakeholders

In the first step, an initial set of stakeholders is identified and they are asked to recommend other stakeholders and stakeholder roles. A recommendation is defined as a triple

<stakeholder, stakeholder role, salience>,

where salience is a number on an ordinal scale (i.e., 1–5).

Then, various social network measures, such as betweenness centrality, degree centrality, and closeness centrality, etc are applied to prioritise the stakeholders in the network. These social network measures then produce a score for each stakeholder. The stakeholder roles are prioritised by the highest score of their stakeholders.

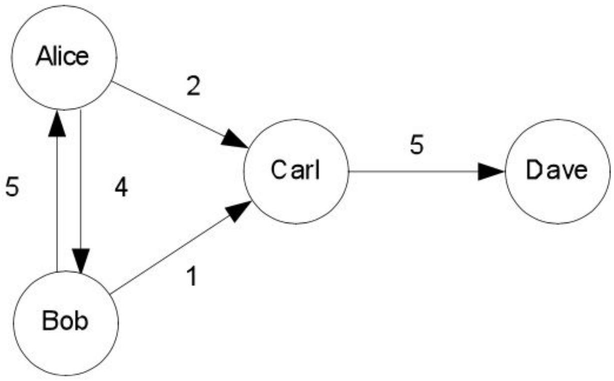


Fig. 2. Prototype stakeholder network.[1]

Step 2: Collect Profile

In the second step, a profile is collected from each stakeholder which is identified in Step 1 and a set of initial requirements are presented to the stakeholders. Existing elicitation methods such as interviews with a subset of stakeholders or focus groups can be used to identify an initial list of requirements. Interviews can be a mixture of two types of questions, one being closed-ended questions which fall under the umbrella of a predefined set of questions and the other type being open-ended questions which are more generic questions and don't expect any predefined answer.

The requirements are then organized into a hierarchy of three levels: project objective, requirement, and specific requirement. Achieving all the specific requirements means that the parent requirement is achieved, and achieving all the parent requirements means that the project objective is achieved.

The stakeholders identified in Step 1 are asked to provide their preferences on the initial requirements. A preference is a triple

<stakeholder, requirement, rating>,

where the rating is a number on an ordinal scale (i.e., 0 – 5).

Stakeholders can also rate requirements which are not on the list by adding their own requirements. The requirements so added are then available to be rated by other stakeholders. If a requirement provided by a stakeholder does not have any specific requirements, specific requirements can be identified using existing elicitation methods (e.g., interviews) and added to the list to be rated.

Step 3: Predict Requirements

Based on the stakeholders' profile developed in Step 2, Step 3 uses recommender systems to predict other requirements that each stakeholder may need. Recommender Systems use the k-Nearest Neighbour (kNN) algorithm in the background to predict similar stakeholders by measuring the similarity between the stakeholders' profiles. Cross-validation is used to find the optimal value for k. The value of k ranges from lower bound 1 to upper bound equal to the total number of samples. N is the number of requirements recommended to the user.

Then, the level of interest that a stakeholder will have in a requirement that he has not yet rated is predicted and the relevant requirements are presented to the stakeholder.

Stakeholders are then asked to rate the requirements that are recommended to them, provide new requirements, or rate other requirements. The new ratings by the stakeholders are then added to their profiles. Then, Step 3 is repeated with the updated profiles. Step 3 can be repeated until no new ratings and requirements are provided by stakeholders.

Step 4: Prioritize Requirements

Lastly, all the stakeholders' profiles are aggregated into a prioritized list of requirements. The ratings from the stakeholders' profiles, along with the priority of the stakeholders and their roles from Step 1 are used to prioritize requirements. The influence of the stakeholder's role is determined by calculating the importance of requirements in a project. The influence of the stakeholders in their role is determined as follows:

$$Influence_{role(i)} = \frac{RR_{max} + 1 - rank(role(i))}{\sum_{j=1}^{n} RRmax + 1 - rank(role(j))}$$

where role(i) is stakeholder i's role in the project, $RRmax$ is the maximum rank of the roles in the list, $rank(role(j))$ is the fractional rank of role j, and n is the total number of roles available.[1]

The influence of stakeholder i in the role is calculated as follows.

$$Influence_{i} = \frac{RS_{max} + 1 - rank(i)}{\sum_{j=1}^{n} RSmax + 1 - rank(j)}$$

where $RSmax$ is the maximum rank of all stakeholders with the equivalent role, rank(i) is the stakeholder i's fractional rank, and n is the total number of stakeholders with the equivalent role.[1]

Again, as generally, the lower rank values correspond to higher influence, the rank value is inverted by subtracting it from the upper bound of maxranks + 1, then the influence is normalized by dividing it with the sum of all the influences of all the stakeholders with the same role.

The influence of stakeholder i in a project is calculated using the following equation.

$$ProjectInfluence_{i} = Influence_{role(i)} \times Influence_{i}$$

where $Influence_{role(i)}$ is the influence of the stakeholder's role in the project and $Influence_{i}$ is the influence of the stakeholder in the role.[1]

The importance of a requirement is calculated using the following equation:

$$Importance_{r} = \sum_{i=1}^{n} ProjectInfluence_{i} \times r_{i}$$

where $ProjectInfluence_{i}$ is the stakeholder i's influence in the project, r_{i} is the rating provided by stakeholder in requirement R, and n is the total number of stakeholders who rated on requirement R.[1]

Finally, the requirements are then prioritized on the basis of their importance, where requirements with higher importance values are ranked higher. The requirements are prioritized within their hierarchy, so that the output is a ranked list of project objectives, for each project objective, a ranked list of requirements, and for each requirement, a ranked list of specific requirements. This list is the final output of the proposed method for the requirements engineers.

IV. Conclusion

In this paper, a new and much better approach to requirement elicitation is mentioned which uses the concept of the social network to build a social network of stakeholders. This network is then used to recommend requirements to the stakeholders by matching their profile with like-minded users. This eases out the work of requirement engineers and makes sure that the requirements of each and every stakeholder is taken into account. The model prioritizes the requirements accurately and handles one major issue of information overload by channelizing the stakeholders' attention to only the relevant requirements that are unknown to them. Thus, the collected list of requirements is deemed to be a near-perfect one.

V. References

[1] Soo Ling Lim, and Anthony Finkelstein. "StakeRare: Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation", IEEE Transactions On Software Engineering 2011

[2] Castro-Herrera, C. J. Cleland-Huang, and B. Mobasher Enhancing stakeholder profile to improve recommendation in online requirements elicitation Proceedings of the 17th IEEE International Conference on Requirements Engineering. 2009: IEEE Computer Society. p. 37-46.

[3] Lim, S.L., "Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation. PhD thesis, 2010,University of New South Wales."

[4] Herrmann, A., M. Danev Requirement prioritization based on benefit and cost prediction- an agenda for future research Proceedings 16th IEEE International Conference on Requirement Engineering 2008.

[5] Davis, A., O. Dieste, A. Hickey, N. Juristo, and A.M. Moreno. "Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review", in Proceedings of the 14th IEEE International Conference on Requirements Engineering. 2006. p.