StakeSource2.0: Using Social Networks of Stakeholders to Identify and Prioritise Requirements

Soo Ling Lim
Dept. of Computer Science
University College London
United Kingdom
s.lim@cs.ucl.ac.uk

Daniela Damian
Dept. of Computer Science
University of Victoria
Victoria, Canada
danielad@cs.uvic.ca

Anthony Finkelstein
Dept. of Computer Science
University College London
United Kingdom
a.finkelstein@cs.ucl.ac.uk

ABSTRACT
Software projects typically rely on system analysts to conduct requirements elicitation, an approach potentially costly for large projects with many stakeholders and requirements. This paper describes StakeSource2.0, a web-based tool that uses social networks and collaborative filtering, a "crowdsourcing" approach, to identify and prioritise stakeholders and their requirements.

Categories and Subject Descriptors
D.2.8 [**Software Engineering**]: Requirements/Specifications

General Terms
Management, Documentation, Human Factors.

Keywords
Requirements elicitation, social networks, collaborative filtering.

1. INTRODUCTION
StakeSource2.0 is a novel tool that identifies and prioritises stakeholders and their requirements using social networks and collaborative filtering. This tool "crowdsources" stakeholders at the start of a software project, using their stake or interest in the project to motivate their input. StakeSource2.0 builds on, and substantially extends, StakeSource, a tool that uses social networks for stakeholder analysis [1]. StakeSource was presented in the formal tool demo track of ICSE'10, and has been used in various software and non-software projects to identify stakeholders [2]. StakeSource2.0 adds support for *requirements identification* and *prioritisation*. This paper describes these new features.

Requirements elicitation is a critical step in software engineering, as incomplete requirements are frequently cited as one of the main causes for project failure [2]. Requirements need to be prioritised, as there are usually more requirements than time and budget allow to be carried forward. Traditional techniques, in which system analysts interview stakeholders to identify and prioritise requirements, do not scale to large projects with many stakeholders and requirements [3]. Often, only a subset of stakeholders is consulted. As a result, elicitation and prioritisation is biased towards the perspective of those stakeholders, and critical requirements may be omitted or wrongly prioritised [2].

ICSE'11, May 21–28, 2011, Waikiki, Honolulu, HI, USA.
ACM 978-1-4503-0445-0/11/05.

To address these problems, StakeSource2.0 automates a method developed in previous research, which consists of two parts [2]:

- **Stakeholder analysis**. *Identifies stakeholders* by asking them to recommend other stakeholders, *builds a social network* of stakeholders from their recommendations, and *prioritises the stakeholders* using social network measures [4].
- **Requirements elicitation and prioritisation**. *Identifies requirements* by asking stakeholders to suggest and rate requirements, *recommends other requirements of interest* to them using collaborative filtering, and *prioritises the requirements* using their ratings weighted by their priority in the social network [5].

This method has been evaluated in a substantial real-world software project, RALIC (the Replacement Access, Library and ID Card project) in University College London [2, 4, 5]. Results show that the method identifies a comprehensive set of stakeholders and requirements, and accurately prioritises requirements.

StakeSource2.0 supports both the stakeholder analysis, and the requirements identification and prioritisation part of the method. It is intended for projects with a large number of stakeholders in distributed locations. It automatically collects requirements and their ratings from each stakeholder, runs the collaborative filtering algorithm, recommends other requirements of interest to the stakeholders, and calculates the priority of each requirement. The tool complements existing requirements management tools, such as CaliberRM, DOORS, Objectiver, RequisitePro, and Scenario Plus[1], which are largely intended for specialist use [6].

The StakeSource2.0 features described in this paper include: the collection of requirements and their ratings, recommendation of other requirements of interest, prioritisation of requirements, and visualisation of the stakeholders' requirements preferences on the social network.

2. STAKESOURCE2.0
StakeSource2.0 provides the following features to identify and prioritise requirements[2]. The features are illustrated using data from the RALIC project.

Feature 1: Identify Requirements

StakeSource2.0 identifies requirements from a large set of stakeholders to increase the completeness of requirements.

Example. In StakeSource2.0, the system analyst provides the list of requirements elicited from interviewing an initial subset of

[1] http://www.volere.co.uk/tools.htm
[2] A video demonstration of StakeSource2.0 is available at: http://www.youtube.com/watch?v=brk4zk5UF20

stakeholders. Based on this list, StakeSource2.0 returns the requirements from all stakeholders.

How StakeSource2.0 does it. Requirements are organised in a simple hierarchical structure in StakeSource2.0, where child nodes are more specific than parent nodes. Analysts and stakeholders can provide requirements and their descriptions at any level. StakeSource2.0 identifies requirements by asking stakeholders to rate a given list of requirements and suggest other requirements.

To use the tool, the analysts create the project and enter a project description. StakeSource2.0 asks stakeholders for requirements by sending an email to each stakeholder, and uses the project details to inform the stakeholders about the project. The email contains a link that will bring the stakeholder to a requirements elicitation and rating form (Figure 1). In this form, the stakeholders provide new requirements, and rate new and existing requirements. A five-star rating means the requirement is very important to them, one-star means it is unimportant. Stakeholders can also vote against requirements (e.g., as shown by the "no-entry" symbol in Figure 1).

Feature 2: Prioritise Requirements

StakeSource2.0 prioritises requirements using the stakeholders' ratings on the requirements and their influence in the project.

Example. Because the system must be delivered rapidly, not all requirements can be implemented. StakeSource2.0 informs the analyst that the requirement "centralised management of access and ID information" is more critical than "exporting data to student systems."

How StakeSource2.0 does it. StakeSource2.0 aggregates each stakeholder's private judgements about a requirement's importance into a prioritised list of requirements. To do so, the stakeholders' ratings on a given requirement are weighted by their project influence [2]. Each requirement is then given a relative priority calculated by the sum of the weighted ratings for that requirement and presented as a prioritised list for the analysts (Figure 2(A)). A stakeholder's influence in the project is calculated using the betweenness centrality measure in the stakeholder network; this measure ranks a stakeholder by summing the number of shortest paths between other pairs of stakeholders that pass through that stakeholder [4]. The measure is used as it produces the most accurate prioritisation [2, 4].

To improve the quality of prioritisation, the analysts can merge different statements referring to the same requirement. Future work will consider crowdsourcing the stakeholders to detect duplicates and improve the quality of the requirements, as well as integration with existing requirements management tools to support other methods of eliciting requirements (e.g., use cases, user stories, and goal modelling).

Feature 3: Recommend Requirements of Interest

StakeSource2.0 predicts a stakeholder's preference on unrated requirements using collaborative filtering techniques, and then recommends requirements with the highest predicted ratings to the stakeholder.

Example. StakeSource2.0 sends an email to Janet, one of the stakeholders, suggesting that she may be interested in the requirement "combine ID card and session card." The requirement is very important to Janet and she gives it a five-star rating.

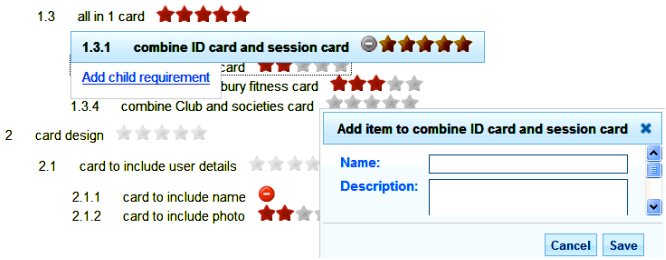

Figure 1. Requirements elicitation and rating form.

How StakeSource2.0 does it. Based on all the stakeholders' existing ratings on the requirements, StakeSource2.0 uses collaborative filtering to predict a stakeholder's preference for an unrated requirement. This feature supports requirements prioritisation by recommending requirements that are identified by only a few stakeholders to other stakeholders who may need it.

Collaborative filtering is a technique used in recommender systems to predict a user's preference on an unrated item [7]. It does so by collecting preference information from many users. For example, Amazon uses collaborative filtering to recommend products to their customers. Recent research has also used collaborative filtering to recommend discussion forums of interest to stakeholders during requirements elicitation [8].

StakeSource2.0 uses the item-to-item collaborative filtering algorithm [7], whereby predictions are generated based on similarities between items (a user who likes item *x* may also like item *y*). The algorithm matches each stakeholder's rated requirements to similar but unrated requirements, and then combines those similar requirements into a list of recommendations for the stakeholder. To determine the most similar match for a given requirement, the algorithm finds requirements that stakeholders tend to need together. For stakeholders who provided insufficient ratings to generate recommendations, StakeSource2.0 presents them with a list of most highly rated requirements. Recommendations approved by the stakeholders are added to their set of ratings.

Feature 4: Highlight Stakeholders in Conflict

StakeSource2.0 highlights stakeholders with conflicting preferences for requirements and reveals their position in the social network.

Example. StakeSource2.0 reveals that many stakeholders oppose the use of their access cards as bank cards. One of them is Richard, the director who is central in the stakeholder network.

How StakeSource2.0 does it. Selecting a requirement highlights stakeholders in the social network who rated it positively (in green) and those who rated it negatively (in purple) (Figure 2(B)). Attention should be given to a requirement if many stakeholders are in conflict, or if the stakeholders in conflict occupy central positions in the network. Selecting a stakeholder enables the analysts to view their requirements (Figure 2(C)). Future work involves clustering stakeholders by their similarity to identify stakeholders who have highly dissimilar requirements preferences.

3. IMPLEMENTATION

For the implementation of StakeSource2.0, previous research [2, 4] identified the following key requirements.

- The tool should be widely available and easy to use to encourage a sufficient number of stakeholders to contribute their requirements and ratings.

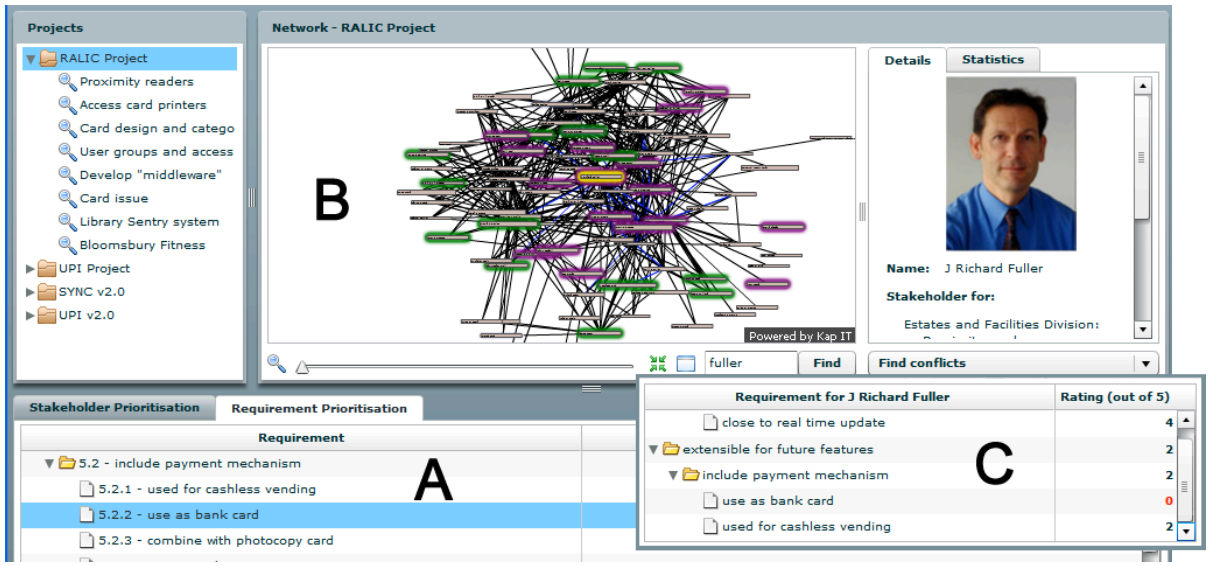

Figure 2. StakeSource2.0 screenshot: (A) Prioritise requirements (B) Highlight stakeholders with conflicting preferences (C) Display stakeholder requirements.

- The analysts should be able to interact with the UI to explore the list of stakeholders and their requirements.
- The collaborative filtering algorithm should incorporate new ratings of requirements dynamically.

The following design decisions were thus made.

- **Web-based**. StakeSource2.0 was implemented as a widely accessible web application using standard web technologies such as HTML, CSS, XHTML, PHP, and JavaScript. MySQL was used for data storage.
- **Standard interface and help**. The rating form (Figure 1) was implemented using standard survey interface from the Smarty Template Engine[3]. Tool tips and pop-up help were supplied to assist stakeholders.
- **Well-established software components**. The Slope One collaborative filtering algorithm was used as it meets the real-time recommendations requirement [9]. Flex Visualizer[4] was used for its interactive network visualisation.

StakeSource2.0 is available from the project website[5].

4. CONCLUSIONS

StakeSource2.0 is simple but has proven in early trials to be a powerful and useful tool. It extends the stakeholder analysis features in StakeSource to support the identification and prioritisation of requirements. It proposes a shift from current practices where system analysts conduct requirements elicitation, to a crowdsourcing approach where all stakeholders have a say. In doing so, it reduces the analysts' workload, increases the completeness of requirements, and accuracy of their prioritisation. To date, more than 10 projects have been using the stakeholder analysis features. The next step is to release the requirements identification and prioritisation features described in this paper to current and future projects.

[3] http://www.smarty.net/
[4] http://lab.kapit.fr/display/kaplabhome/Home
[5] http://www.cs.ucl.ac.uk/research/StakeSource/

5. ACKNOWLEDGMENTS

We thank Jason Lau for his contribution in the software development, and UCL Business for the "Proof of Concept" funding.

6. REFERENCES

[1] Lim, S. L., Quercia, D., and Finkelstein, A. 2010. StakeSource: harnessing the power of crowdsourcing and social networks in stakeholder analysis. In *Proc. of the 32nd Int. Conf. on Soft. Eng. Vol. 2*, Cape Town, p. 239-242.

[2] Lim, S. L. 2010. *Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation*. PhD, University of New South Wales, Sydney, Australia. Available at: http://www.cs.ucl.ac.uk/staff/S.Lim/phd/thesis_soolinglim.pdf

[3] Cleland-Huang, J., and Mobasher, B. 2008. Using data mining and recommender systems to scale up the requirements process. In *Proc. of the 2nd Int. Workshop on Ultra-Large-Scale Software-Intensive Systems*, Leipzig, p. 3-6.

[4] 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. Vol. 1*, Cape Town, p. 295-304.

[5] Lim, S. L., and Finkelstein, A. StakeRare: using social networks and collaborative filtering for large-scale requirements elicitation. *submitted to the IEEE Trans. on Soft. Eng.*

[6] Decker, B., Ras, E., Rech, J., Jaubert, P., and Rieth, M. 2007. Wiki-based stakeholder participation in requirements engineering. *IEEE Software*. 24(2): p. 28-35.

[7] Schafer, J., Frankowski, D., Herlocker, J., and Sen, S. 2007. Collaborative filtering recommender systems. In The Adaptive Web. p. 291-324.

[8] Castro-Herrera, C., Cleland-Huang, J., and Mobasher, B. 2009. Enhancing stakeholder profiles to improve recommendations in online requirements elicitation. In *Proc. of the 17th Int. Conf. on Req. Eng.*, Atlanta, GA, p. 37-46.

[9] Lemire, D., and Maclachlan, A. 2005. Slope one predictors for online rating-based collaborative filtering. In *SIAM Int. Conf. on Data Mining*, Newport Beach, California.