Argumentation-Based Discussion for User Forum: A Research Preview

Itzel Morales-Ramirez[1,2] and Anna Perini[1]

[1] Software Engineering Research Unit, Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento-Povo, Italy
{imramirez,perini}@fbk.eu
[2] International Doctoral School ICT- University of Trento, Italy

Abstract. [**Context and motivation**] User forums provide a virtual space in which participants post comments, upon their experience in using a software, that analysts can eventually redirect to an issue tracking system. Before users post any comment, they should search for a request that is the closest to the one they are about to submit. In doing this, they can face with large, unstructured discussions. [**Question/problem**] Current user forum discussions are usually developed as sequential comments that hide an explicit recognition of the attitude of the participants (i.e. "in favour" or "against") wrt. the initial request. This poses difficulties to the analysts who should identify worth requests to be further analysed. [**Principal ideas/results**] The key idea in our approach is to exploit AI argumentation. The resulting argumentation-based discussion will allow participants to get an overview of the trend of such a discussion, and support analysts to identify important requests. [**Contribution**] In this research preview, we describe how we represent the forum's discussion management problem in terms of AI argumentation concepts, and a sketched algorithm for supporting the forum's participants tasks. A research plan for implementing and evaluating the proposed argumentation-based discussion is also described.

Keywords: Requirements engineering, User forum, Argumentation framework.

1 Introduction

The role of social media to enable collaboration in software projects has been discussed in recent work [1–6], which point out potential benefits and challenges. An example of a collaborative platform is, for instance, an online user forum that provides a virtual space in which participants exchange views on issues about a software application on the basis of their experience in using it. Forums are widely exploited by open source software projects where users of such applications post comments and other participants called "volunteer" developers act as analysts who redirect relevant requests to an issue tracking system (e.g. Bugzilla). In order to prevent the creation of complex, entangled discussions, the participants are requested to follow some rules, e.g. *"Do NOT submit a*

C. Salinesi and I. van de Weerd (Eds.): REFSQ 2014, LNCS 8396, pp. 232–238, 2014.

problem report without searching the existing ones first to ensure that the issue you are reporting has not already been addressed", this rule is specified in the issue tracking system of Apache OpenOffice (AOOo) [7]. This implies that users should search for the closest request to the one they are about to submit, by simply making a text searching. Once, they find the request they might read the whole sequence of comments and provide theirs upon the read, but they could get lost if the discussion is large, as reported also in [8]. Only a careful reading of the content of each comment may reveal to which previous comment it refers to, thus inferring participants' attitude (i.e. "in favour" or "against") towards the comment that initiated the discussion. Usually an overview of participants' attitude in the discussion is missing, and this challenges also the task of analysts when evaluating what is worthy to be further analysed, as for instance to identify new candidate requirements.

The goal of our research is to define methods and techniques to support users and analysts of user forums when performing the above mentioned tasks. We structure it along the following research questions:

RQ1. How can the attitude of the participants in user forums be made explicit and recorded within the structure of a discussion?

RQ2. How can a structured discussion support decisions on what comments should be further analyzed?

To answer these questions, we propose to represent an online discussion as a structured set of arguments according to the AI argumentation [9] that describes a theory of logical inference and techniques for deriving conclusions from arguments. Based on this, participants would be able to provide new comments wrt. existing ones in the ongoing discussion, in a straightforward way. Moreover, the overall attitude towards the initial comment will be automatically inferred. This will allow analysts to recognize the relevant requests, and we believe that also the quality of comments will improve.

In this paper we give a preview of our research by stating the forum's discussion management problems in terms of the AI argumentation in Section 2. The related work is mentioned in Section 3. A research plan for implementing and evaluating the proposed approach along with some concluding remarks are presented in Section 4.

2 Argumentation-Based Discussion Forum

We propose an extension to the Dung's abstract argumentation framework [9] to enable a structured discussion in user forums. According to [9] an *abstract argumentation framework* (AF) is a pair $\langle \mathcal{A}, Def \rangle$. $\mathcal{A}$ is a set of arguments and $Def \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation of *defeat* between arguments. *Defeat* means that an argument y_i *attacks* an argument y_j, therefore $(y_i, y_j) \in Def$.
The concepts *conflict-free* and *defence* are defined as follows.

- Let $\mathcal{B} \subseteq \mathcal{A}$.
- A set $\mathcal{B}$ is *conflict-free* iff there exist no y_i, y_j in $\mathcal{B}$ such that y_i *defeats* y_j.

– A set $\mathcal{B}$ *defends* an argument y_i iff for each argument $y_j \in \mathcal{A}$, if y_j *defeats* y_i, then there exists y_k in $\mathcal{B}$ such that y_k *defeats* y_j.

A full implementation of this framework in terms of a directed graph is given in [10]. We adapt and extend it to represent an *Argumentation-based Discussion Forum* (ADF) as follows. A *comment* is an abstract argument, the Def relation is refined into the *support*, *rejection* and *neutral* relations between pairs of comments[1], and we include the explicit representation of the *participant* with her knowledge confidence (i.e. $know_c$) as a weight associated to her comments. We represent an ADF as a directed acyclic graph (DAG), referred to as $\mathcal{G} = (V, E)$ where:

– $\boldsymbol{V}$ is the set of comments in the discussion, i.e. vertices in $\mathcal{G}$.
– $\boldsymbol{E}$= $S \cup R \cup N$ is the set of pairs of comments represented as edges in $\mathcal{G}$, where S is the set of support relations, R is the set of rejection relations and N is the set of neutral relations that are defined as follows:
  - $R \subseteq V \times V$ is the set of pairs of comments between which a rejection relation holds, i.e. $Reject(y_i, y_j)$, if y_i rejects (attacks) y_j. This is based on the set Def introduced above.
  - $S \subseteq V \times V$ is the set of pairs of comments between which a support relation holds, i.e. $Support(y_i, y_j)$.
  - $N \subseteq V \times V$ is the set of pairs of comments (y_i, y_j), such that y_i adds extra information to y_j.

Currently, we consider that a vertex (comment) contains information such as an identifier (ID), participant's name, description and the $know_c$ parameter. This last is a real number ranging from 0 to 1, which represents participants' perception on their own expertise level. The $know_c$ is asked to the participants using a likert-type scale, e.g. novice=0, initiate=0.25, apprentice=0.50, advanced=0.75, and expert=1, this is adapted from [12] and [13]. This parameter is used for the computation of support and rejection relations between a pair or comments, see Algorithm 1.

To exemplify an ADF let's look at an excerpt of a discussion found in AOOo bugzilla, see Figure 1. As can be observed, the current format of a discussion (top) is sequential and hinders the trend of supporting and rejecting comments wrt. a given comment.

On the other side, the ADF graph (down) makes explicit that comment 3 rejects comment 1, while comment 8 and 11 support the comment 0 and 3, respectively. Comments 0, 3, 8, and 11 define a set of relevant comments for comment 0 (i.e. $RelC_{y_0}$), which is a set of comments where given an initial comment y_0 belonging to the set, there are no comments, included in the set, that reject y_0.

We use DAG search algorithms to define procedures that can help participants and analysts. Example of procedures for participants are: *(i)* finding if a given

[1] Analogous works extend Def relation with the support relation but not with the neutral one (e.g. [11])

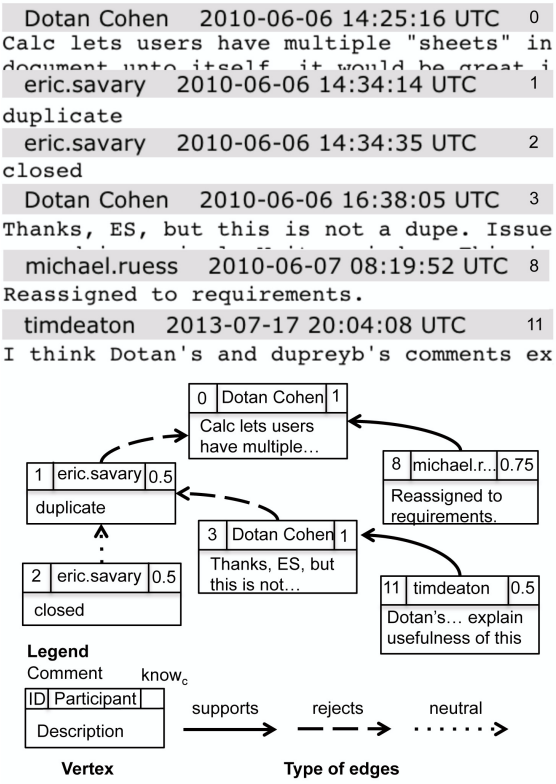

Fig. 1. Excerpt of the discussion #112163 in AOOo bugzilla (top), representation of it as an ADF directed acyclic graph (down)

comment y_0 in the actual discussion is supported or rejected; *(ii)* finding the set of comments that support, reject or are neutral wrt. a selected comment y_0; *(iii)* computing the effect of adding a new comment (either reject or support type) to the actual discussion. To support analysts: *(iv)* finding the most supported request in the discussions that are active in a forum.

To give a flavour of the basic algorithms used, Algorithm 1 sketches how to compute *RelC* for a given ADF. Lines 14 and 16 show how the parameter $know_c$ is used to compute the strength of the support and rejection relations.

To update *RelC* once a discussion is modified, by the addition of a new comment, we can use an incremental search algorithm that considers only the subgraph affected by the change.

3 Related Work

Recent work shows a growing interest within the Requirements Engineering research community towards social media as distributed, collaborative work

Algorithm 1. proc_RelC(G, v): pseudocode for computing $RelC$ in ADF

Input A DAG G representing an ADF, v is a starting $vertex\, in\, V(G)$
Output RelC which is the set of relevant comments of v. {//$RelC$ is initialized to $\emptyset$}

if v is a *leaf* **then**
  return $RelC \leftarrow RelC \cup \{v\}$ {//iif v is *root*}
else
  neutral=0 {//Counts the number of neutral relations}
  supportS=0.0 {//Counts the weight $know_c$ of v in the support relation}
  rejectionS=0.0 {//Counts the weight $know_c$ of v in the rejection relation}
  for each edge $e \in G.adjacentEdges(v)$ **do**
    $v' \leftarrow G.adjacentVertex(v, e)$
    switch $v'.outgoingRelation$
      case NEUTRAL
      neutral++ {//Recursive call and addition of neutral comments}
      proc_RelC(G, v')
      case SUPPORT
      supportS+=proc_RelC(G, v') + $v'.know_c$/(supportS + rejectionS + neutral){//Recursive call and normalisation of supportS}
      case REJECTION
      rejectionS+=proc_RelC(G, v') + $v'.know_c$/(rejectionS + supportS + neutral){//Recursive call and normalisation of rejectionS}
    end switch
    $RelC \leftarrow RelC \cup \{v'\}$ {//iif $supportS_{v'} > rejectionS_{v'}$}
  end for
end if

enablers. For instance, the discovery of stakeholder communities by using concept lattices to extract hidden profiles for the set of requirements of a certain project [6], or the StakeRare method described in [2] uses social networks and a collaborative filtering to elicit and prioritise requirements in large projects. Similarly in [14], participants' opinions posted through social networks are analyzed and exploited for requirements prioritisation. In [15] collaborative filtering is used to facilitate online discussions for requirements identification. These three works assume to start with an initial requirements set to be refined through social collaborations, while in our research we see the discussion as the potential source for requirements.

In [4] is presented a software platform, called Requirements Bazaar, that supports gathering and negotiation on user feedback about software applications. Focusing on on-line discussions, the IdeaTracker [8] tool provides a way to support interface design review via discussion, by associating a color-code to comments classified along their affective tone (i.e. negative-red, positive-green, or both-yellow). Both works require users to explicitly express their preference (vote) on the emerging requests, while in our approach we aim at automatically inferring the effects of the developed argumentation on the statement that initiated it.

Concerning the use of argumentation-based approaches in RE, worth mentioning are the work of Jureta et al. [16] that proposes the ACceptability Evaluation Framework (ACE), to support stakeholders when performing requirements validation. This framework was applied in [17] to support validation of law-compliance of software requirements by a team including law experts and software engineers. Analogously in [18] argumentation techniques are proposed to validate requirements and to highlight inconsistencies that may foster the elicitation of missing requirements.

An extension to the Dung's framework [9] is described in [11], this work adds the possibility of assigning a strength to the argument, and use it for inference. We make extensions to Dung's framework to include for instance the participants' knowledge confidence that will be used for the computation of their attitude in the discussion.

4 Concluding Remarks and Research Plan

In this paper we introduced the two research questions that drive our work on argumentation-based discussion for user forum, which rests on an extension of Dung's framework and exploits DAG algorithms to compute the supported and rejected comments, whose participants' knowledge confidence about the topic under discussion is also taken into account. We illustrated on a simple example taken from the Apache OpenOffice bugzilla how participants' attitudes towards the initial statement can be made explicit in a structured discussion ADF. We are implementing the proposed ADF using Neo4j [19] and collecting experimental evidences on the scalability of the ADF management algorithms on artificial dataset containing argumentation with increasing numbers of comments (vertices) and different percentages of support, reject and neutral relations (edges). We plan to use different functions to compute the relevant comments, thus performing a sort of sensitivity analysis. As a longer term objective, we aim at integrating the proposed argumentation-based discussion into an collaborative platform, e.g. [8], thus a further task will be that of selecting an hosting platform and integrating our approach in it. This will allow us to perform an empirical evaluation on the effectiveness of our Argumentation-based Discussion Forum with users and analysts.

References

1. Begel, A., Bosch, J., Storey, M.: Bridging software communities through social networking. IEEE Software 30(1), 26–28 (2013)
2. Lim, S.L., Finkelstein, A.: Stakerare: Using social networks and collaborative filtering for large-scale requirements elicitation. IEEE Trans. Softw. Eng. 38(3), 707–735 (2012)
3. Pagano, D., Maalej, W.: How do open source communities blog? Empirical Software Engineering 18(6), 1090–1124 (2013)

4. Renzel, D., Behrendt, M., Klamma, R., Jarke, M.: Requirements bazaar: Social requirements engineering for community-driven innovation. In: 21st IEEE International Requirements Engineering Conference, RE 2013, pp. 326–327. IEEE (2013)
5. Sutcliffe, A., Sawyer, P.: Requirements elicitation: Towards the unknown unknowns. In: 21st IEEE International Requirements Engineering Conference, RE 2013, pp. 92–104. IEEE (2013)
6. Azmeh, Z., Mirbel, I., Crescenzo, P.: Highlighting stakeholder communities to support requirements decision-making. In: Doerr, J., Opdahl, A.L. (eds.) REFSQ 2013. LNCS, vol. 7830, pp. 190–205. Springer, Heidelberg (2013)
7. Apache: Apache openoffice bugzilla (2010), `https://issues.apache.org/ooo/`
8. Zilouchian Moghaddam, R., Bailey, B.P., Poon, C.: Ideatracker: An interactive visualization supporting collaboration and consensus building in online interface design discussions. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011, Part I. LNCS, vol. 6946, pp. 259–276. Springer, Heidelberg (2011)
9. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–357 (1995)
10. Bex, F., Prakken, H., Reed, C.: A formal analysis of the aif in terms of the aspic framework. In: Proceedings of the 2010 Conference on Computational Models of Argument: Proceedings of COMMA 2010, pp. 99–110. IOS Press, Amsterdam (2010)
11. Leite, J., Martins, J.: Social abstract argumentation. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI 2011, vol. 3, pp. 2287–2292. AAAI Press (2011)
12. Ericsson, K.A.: The Cambridge handbook of expertise and expert performance. Cambridge University Press (2006)
13. Likert, R.: A technique for the measurement of attitudes. Archives of Psychology (1932)
14. Fitsilis, P., Gerogiannis, V., Anthopoulos, L., Savvas, I.: Supporting the requirements prioritization process using social network analysis techniques. In: 2010 19th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), pp. 110–115 (2010)
15. Castro-Herrera, C., Cleland-Huang, J., Mobasher, B.: Enhancing stakeholder profiles to improve recommendations in online requirements elicitation. In: 17th IEEE International Requirements Engineering Conference, RE 2009, pp. 37–46 (2009)
16. Jureta, I., Mylopoulos, J., Faulkner, S.: Analysis of multi-party agreement in requirements validation. In: 17th IEEE International Requirements Engineering Conference, RE 2009, pp. 57–66 (2009)
17. Ingolfo, S., Siena, A., Mylopoulos, J., Susi, A., Perini, A.: Arguing regulatory compliance of software requirements. Data & Knowledge Engineering 87, 279–296 (2013)
18. Mirbel, I., Villata, S.: Enhancing goal-based requirements consistency: An argumentation-based approach. In: Fisher, M., van der Torre, L., Dastani, M., Governatori, G. (eds.) CLIMA XIII 2012. LNCS, vol. 7486, pp. 110–127. Springer, Heidelberg (2012)
19. Neo Technology, I.: Neo4j (2013), `http://www.neo4j.org/`