A Goal-oriented Analysis to Guide the Development of a User Feedback Ontology

Renata Guizzardi[1] , Itzel Morales-Ramirez[2,3], Anna Perini[2]

[1]Ontology and Conceptual Modeling Research Group, UFES, Brazil
{rguizzardi@inf.ufes.br}
[2]Software Engineering Research Unit, Fondazione Bruno Kessler, Italy
[3]International Doctoral School ICT, University of Trento, Italy
{perini,imramirez}@fbk.eu

Abstract. Nowadays, developers and service providers put a lot of effort on collecting and analyzing user feedback with the purpose of improving their applications and services. This motivates the proposal of new tools to collect and analyze feedback. In our work, we develop a user feedback ontology, aimed at clarifying the concepts of this domain. For that, we follow a goal-oriented methodology to identify the competency questions that represent the ontology requirements. In this paper, we discuss an excerpt of the goal model used to guide the development of our ontology. Moreover, we present examples of competency questions identified through the analysis, and the corresponding fragment of the user feedback ontology.

Keywords: User feedback, Ontology, Goal-oriented analysis, Competency questions.

1 Introduction

Users currently place comments and opinions about software applications and services in a variety of online social media and repositories, such as user forums, App stores and mailing-list archives. User feedback represents an invaluable source for improving the software, attracting the attention of researchers in Requirements Engineering, who started investigating it along different perspectives. Recent initiatives in this area include: characterizing user feedback in App stores [1]; providing methods and techniques to collect it [2,3], and approaches to analyze it so as to elicit relevant information for evolving software requirements [4,5]. In a sense, user feedback analysis may be seen as a kind of customer opinion mining [1], but differently, also single comments pointing out critical bugs or new ideas to support software evolution are of paramount importance.

In our ongoing work, we are developing a user feedback ontology that aims at clarifying the concepts of this domain. This ontology can be applied to support the development of feedback collection and analysis techniques, (a) by providing a taxonomy of user feedback to support structured feedback collection, and (b) to support the feedback analysis phase, by guiding analysts to understand the intentions behind unstructured user feedback. Current tools that aim at providing information management strategies are mainly focused on supporting marketing purposes [1]. For instance, tools performing sentiment analysis to discover the success or failure of a certain product. The ontology we propose supports a better understanding of the different types of feedback, its format, and the purpose for what it can be used.

To develop the user feedback ontology, we follow a methodology that consists in performing a goal-oriented analysis of the domain to identify the competency questions that serve as the ontology's requirements [1]. This methodology (summarized in Section 2) is particularly effective for managing the problem domain complexity. Moreover, it profits from the Tropos' constructs and analysis techniques to help delineating the ontology's scope and capture competency questions. In Section 3 of this paper, we provide a glimpse of the performed goal-oriented analysis, which considers the main stakeholders involved in the user feedback domain, namely, *software users*, *feedback collectors*, *feedback analysts* and *requirement analysts*. Additionally, we present some examples of questions identified throughout this analysis and the corresponding fragment of our user feedback ontology. Section 4 concludes the paper.

2 Methodology

In existing Ontology Engineering methodologies, the requirements for the ontology under development are typically captured through competence questions. CQs are natural language questions to which the ontology should be able to respond. However, no specific technique is used to elicit and analyze such questions. Conversely, it is assumed that the ontology engineer will be able to formulate them from scratch. To fill in this gap, Fernandes, Guizzardi and Guizzardi [1] propose a methodology based on the use of Tropos [7] to elicit and analyze the competence questions of an ontology-to-be. The main benefit of this approach relies on the use of Tropos's analysis techniques to systematically capture the ontology's requirements, by analyzing the goals of the domain stakeholders. This enables the ontology engineer to elicit and refine competence questions.

The methodology starts by applying the Tropos's existing analysis method in early requirements, i.e. by analyzing the goals of the domain's actors. In late requirements, some competence questions are defined to accomplish these actors' goals. Then, refining the goal model naturally leads to the refinement of the competence questions. Besides, at this stage, we may already elicit a few concepts used to define user feedback, directly deriving them from such questions. These concepts will later compose the actual ontology model. In the ontology modeling activity, such initial concepts are specified and refined, applying OntoUML, a well-founded UML-based language developed to specify ontologies [8].

3 Contribution

Figure 1 depicts the main stakeholders involved in the user feedback domain, along with the strategic dependencies among them. As argued in Section 1, software users' (represented by the Software User actor) wants and needs are a central concern in this domain, driving the process and motivating the dependencies in this scenario. The

Requirements Analyst is, as usual, responsible to maintain software requirements, relying on the Feedback Analyst to understand how the user feedback affects the current requirements. The Feedback Collector is mainly concerned with gathering detailed feedback while also facilitating both the Software User and the Feedback Analyst in accomplishing their own goals.

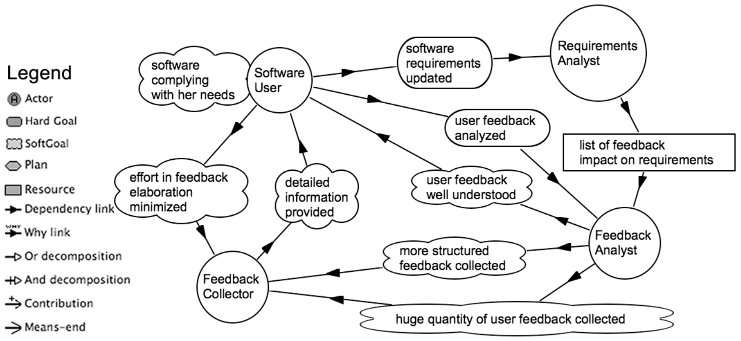
Legend
Actor
Hard Goal
SoftGoal
Plan
Resource
Dependency link
Why link
Or decomposition
And decomposition
Contribution
Means-end
software complying with her needs
Software User
software requirements updated
Requirements Analyst
user feedback analyzed
list of feedback impact on requirements
effort in feedback elaboration minimized
detailed information provided
user feedback well understood
Feedback Analyst
more structured feedback collected
Feedback Collector
huge quantity of user feedback collected

Fig. 1. Tropos actor model showing the stakeholders in the user feedback domain.

For reasons of space, we give special attention to the Feedback Analyst. In this respect, it is important to emphasize that the Feedback Analyst depends on the Feedback Collector to collect and send her a huge amount of feedback (huge quantity of user feedback collected softgoal) because the more information she has on the needs of the software's community of users, the more accurate is the result of the feedback analysis. Moreover, the Feedback Analyst would also like the feedback to be as structured as possible (more structured feedback collected softgoal), because this facilitates the analysis and has a positive impact on the quality of the feedback.

Figure 2 analyzes in detail the perspective of the Feedback Analyst.

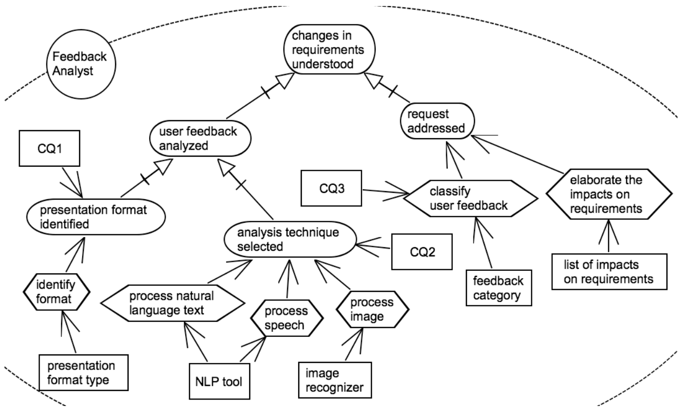

Fig. 2. Tropos goal diagram analyzing the feedback analyst's perspective.

The Feedback Analyst has the main goal of understanding to which extent the feedback reflects needs for software requirements changes (changes in requirements understood goal). For accomplishing this main goal, the Feedback Analyst aims at analyzing the user feedback (user feedback analyzed goal) and address the request present in the user feedback (request addressed goal). To accomplish the feedback analyzed goal, the Feedback Analyst aims at identifying the feedback's presentation format (presentation format identified goal) and selecting the analysis technique (analysis technique selected goal). For obtaining the former, the analyst identifies the format (identify format plan) and to achieve the latter, the analyst may choose among several techniques, which also depend on the feedback presentation format. For example, the analyst may choose process natural language text, process speech or process image. The information regarding the presentation format type is a resource produced by the identify format plan. Regarding the analysis techniques, both process natural language text and process speech require the use of an NLP tool, while process image requires the use of an image recognizer.

After the feedback is properly analyzed, the Feedback Analyst may finally address the request (request addressed goal). For that, the analyst classifies the feedback according to the user's intention (classify request plan), i.e. the analyst identifies if the user submitted a *bug report*, a *software rating*, a *clarification request* or a *new feature request*. Moreover, the analyst analyzes how the feedback affects the current software requirements (elaborate the impacts on requirements plan). The former generates the feedback classification resource while the latter produces the list of impacts on requirements, made available to the Requirements Analyst (see resource dependency between the Requirements Analyst and the Feedback Analyst) so that she can update the system's requirements.

At this point, a few competency questions (CQs) can be identified from the goals and plans of the Feedback Analyst. In the model of Fig. 2, we can see CQ1 linked to the presentation format identified goal, CQ2 linked to the analysis technique selected goal, and CQ3 linked to the classify request plan. Table 1 enunciates these CQs.

Table 1. Competency Questions derived from the Tropos goal model.

Competency Questions
CQ1. In which formats can a user feedback be presented?
CQ2. What analysis techniques can possibly be selected for feedback analysis?
CQ3. In which categories can a user feedback be classified?

In general, CQs are connected to goals (both hard and soft), since as goals, they are related to requirements (e.g. CQ1 and CQ2). However, at times, the goal diagram is already refined, typically from (soft)goals into plans. In this case, competency questions can also be attached to plans (e.g. CQ3 because the goal in this case does not provide a concrete answer). This is actually reasonable, since plans and goals are very interrelated concepts. In a sense, every time a plan is modeled, a goal is implicitly there (the goal of accomplishing that particular plan), as goals may be defined in different levels of abstraction. Allowing CQs to be connected either to (soft)goals or plans make the methodology more flexible, i.e. it can be used both when the model is particularly made to develop an ontology or when the model supports an existing software system. Given a goal diagram, the ontology engineer should inspect all (soft)goals and plans, as it is up to her to define the scope of the ontology-to-be. Typically, when the model is made from scratch, more elements will lead to the definition of CQs, since in this case, the goal model itself more likely shares the scope of the ontology-to-be.

Now, a question remains whether to place the CQ closer or farther from the leaf in the intentional element tree. Coincidentally, all CQs in Fig. 2 are at the same level in the tree. Nevertheless, taking a closer look, while CQ3 is linked to a plan operationalizing a goal, CQ2 is connected to a goal that is then refined into three plans. Thus, one might ask: why connecting CQ2 to the analysis technique selected goal and not to the process image plan? This decision is directly related to how the question is enunciated (refer to Table 1). Note that our feedback ontology would be less flexible if CQ2 were "May process image be selected for feedback analysis?"

After modeling and enunciating CQs, we can elicit some possible ontological concepts from them (typically the nouns), for instance, *presentation format*, *analysis technique* and *category*. Figure 3 shows an excerpt of the ontology created to respond to the CQs. These three concepts are there, along with some others that refine them. Here are a couple of CQs derived from other actors in our scenario: what are the common written expressions of users to describe a problem of a software application? What is the preferred format of conveying user feedback? A complete version of this ontology can be found in [9].

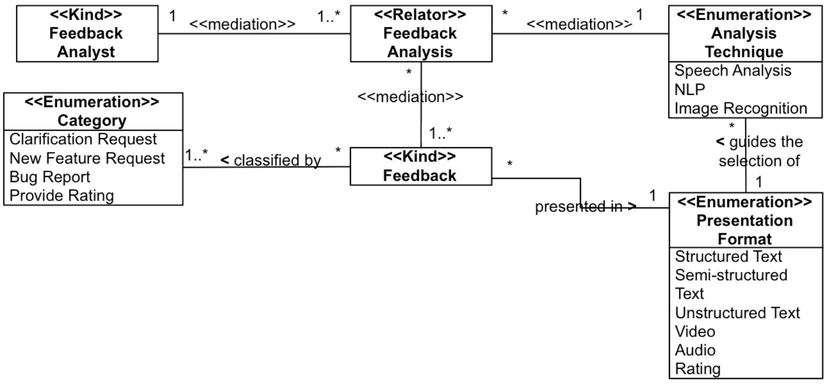

Fig. 3. Excerpt of the user feedback ontology

Following the chosen Ontology Engineering methodology, we applied OntoUML to design the ontology, so the concepts are founded according to the categories (depicted as UML stereotypes) of the UFO foundational ontology. For reasons of space, we refrain from describing UFO's ontological categories, detailed described in [8].

In our ontology, the Feedback Analyst analyzes the Feedback using Analysis Technique of a particular type (e.g., Speech Analysis, NLP or Image Recognition). The Feedback Analysis represents an analysis session and binds Feedback Analyst, Feedback and Analysis Technique. It can be thought as a complex dependent object that aggregates a number of Feedback instances that are analyzed as part of the same session, performed by the same unique Feedback Analyst, and using the same Analysis Technique. Moreover, a Feedback is presented following a specific Presentation Format, which can be Structured Text, Semi-Structured Text, Unstructured Text, Audio, Video, or Rating. This Presentation Format guides the choice for a particular Analysis Technique, for instance, if the format is Unstructured Text, NLP is a good candidate to support the analysis. Additionally, the Feedback is classified in one of the following categories: Bug Report, Clarification Request, New Feature Request and Rating. These categories are chosen based on user feedback literature, e.g. [1,10].

4 Conclusion

In this paper, we described the initial phase of the ontology construction process we are following to build an ontology of user feedback. It consists of a goal-oriented analysis of the domain that helped us identify the main competency questions that the resulting ontology should support answering. We illustrated the ontology construction process using an excerpt of the goal model and the corresponding fragment of the resulting ontology. We believe that this user feedback ontology can be useful to support defining appropriate feedback collection tools, and help the analysis of unstructured user feedback. Specifically, it can provide a well-founded model for the tool-supported method for the analysis of unstructured feedback presented in [11].

References

1. Pagano, D., Maalej, W.: User feedback in the Appstore: An Empirical Study. In: RE'13 (2013) 125-134.
2. Seyff, N., Graf, F., Maiden, N.: Using mobile RE tools to give end-users their own voice. In: RE'10 (2010) 37–46.
3. Schneider, K.: Focusing spontaneous feedback to support system evolution. In: RE`11 (2011) 165–174.
4. Carreño, L., Winbladh, K.: Analysis of user comments: an approach for software requirements evolution, In: ICSE. IEEE / ACM (2013) 582–591.
5. Chen, N., Lin, J., Hoi, S., Xiao, X., Zhang, B.: AR-Miner: Mining Informative Reviews for Developers from Mobile App Marketplace. In: ICSE 2014. To appear.
6. Fernandes, P., Guizzardi, R., Guizzardi, G.: Using Goal Modeling to Capture Competency Questions in Ontology-based Systems. Journal of Information and Data Management. v. 2 (2011) 527-540.
7. Bresciani, P., Perini A., Giorgini P., Giunchiglia F., and Mylopoulos J.: Tropos: An Agent-oriented Software Development Methodology. Journal of Autonomous Agents and Multi-Agent Systems 8(3): (2004) 203–236.
8. Guizzardi, G. Ontological Foundations for Structural Conceptual Models. Ph.D. thesis, University of Twente, The Netherlands (2005).
9. Morales-Ramirez, I., Perini, A., Guizzardi, R.: Providing Foundation for User Feedback Concepts by Extending a Communication Ontology. In: ER'14 (2014). Submitted.
10. Mory, E.H.: Feedback research revisited. Handbook of research on educational communications and technology 45(1) (2004) 745–784.
11. Morales-Ramirez, I., Perini, A.: Discovering Speech Acts in Online Discussions: A Tool-supported method. In: CAiSE'14 Forum. (June 2014) To appear.
12. Cambria, E., Schuller, B., Xia, Y. and Havasi, C. New Avenues in Opinion Mining and Sentiment Analysis. IEEE Intelligent Systems 28(2):15-21, March 2013.