Using Intentional Modeling to Discover the User Preferences in Existing Software Systems

Eric-Oluf Svee, Jelena Zdravkovic

Department of Computer and Systems Sciences
Stockholm University, Forum 100, SE-164 40 Kista, Sweden
{eric-sve, jelenaz}@dsv.su.se

Abstract. Information systems consist of many low-level components, such as source code, functioning together as a unified whole. However, higher-level requirements, such as the preferences of users concerning quality intentions of the system, are often under-documented, if documented at all. Consequently it can be highly important to elicit the intentions underlying the system for driving or justifying system acceptance in a certain business context. This paper utilizes intentional modeling to discover just such user preferences in existing software systems, in this case a system for managing of the student thesis process in a Swedish university. To accomplish this, stepwise guidelines are proposed for evaluating what user preferences an extant software system expresses. These are presented in a feature model, which is then mapped to the software systems goals, which are themselves represented in i*.

Keywords. Requirements Engineering, Consumer Preferences, Goal Modeling, iStar

1 Introduction

This work proposes that user preferences for software systems can be thought of as *consumer preferences*. We proposed a solution in our previous work [9,13] to capture consumer preferences and introduce them into the development of software systems that will support them. In this study a real case is used to illustrate the feasibility of deriving user preferences of an existing software system by following a value typology classified in the Consumer Preference Meta Model [9,13] to generate feature models as defined in FODA [5], which is in turn are used to generate goal model in i* [4].

The significance of having a reverse-engineering method for the elicitation of goal models—in this case accomplished by demonstrating the support for desired or mandated consumers preference that are then classified according to a chosen user value framework—comes from the many systems that are faced either with non-existing or poorly formed requirements. In a perfect world all requirements would be well written, in addition to being firmly grounded on consumer preferences. The novelty of this approach allows for existing software systems to be evaluated for supporting, and enabling, consumer preferences. It differs from prior approaches due to its explicit adoption of techniques that are commonly deployed in usability testing and evaluation. This allows for low-costs and low-training demands, creating a low-threshold for adoption of the technique.

This paper is organized as follows: Section 2 presents prior work in this area, a brief overview of the consumer preference framework utilized, as well as the usability techniques adopted. Section 3 describes the proposed reverse-engineering method for the elicitation of goal models while Section 4 presents the conclusions and directions for future research.

2 Background

Prior work in this area includes [1, 9, 11,12,13]. Each adopts a form of intentional modeling as its primary approach. In contra poise, the novelty of the present approach lies in its adoption of low-cost, easily accessible, off-the-shelf techniques commonly utilized within usability testing and design.

User preferences and their relation in terms of a consumer preference meta-model were conceptualized in [9,13]. This included a classification of, and the mappings between, several common consumer frameworks that are considered to represent preferences of users at different abstraction levels. One of those well-known classifications of user preferences we find as a relevant for this study is Holbrook's Consumer Value typology, coming from the marketing sector [3]. Using three dimensions—*Extrinsic/Intrinsic*, *Self-oriented/Other-oriented*, and *Active/Reactive*, Holbrook's Typology of Consumer Values identifies eight archetypes that represent distinct types of value in the consumption experience: Efficiency (e.g., Convenience), Excellence (e.g., Quality), Status (e.g., Success), Esteem (e.g., Reputation), Play (e.g., Fun), Aesthetics (e.g., Beauty), Ethics (e.g., Justice), and Spirituality (e.g., Faith).

Commonly used techniques from usability testing are adopted for their low-threshold for implementation; costs and training for both are quite minimal and numerous free resources exist. Inputs that are part of a Heuristic Evaluation [8] (a prototype, typical tasks, and user profile) are utilized, in addition to those of a Cognitive Walkthrough [6]. Whereas heuristic evaluation focuses on individual elements in the interface, a cognitive walkthrough focuses on individual actions in the sequence, asking a number of questions about the learnability of each action. It requires an explicit sequence of actions for each task—the path that the walkthrough process follows.

3 Scientific Contributions

This paper utilizes intentional modeling to discover the user preferences in existing software systems, in this case a tool for administering the thesis process at a university. To accomplish this, stepwise guidelines are proposed for evaluating what user preferences an extant software system expresses. These are then presented in a feature model, which is then mapped to the software systems goals, which are themselves represented in i*.

***Guideline 1*: Document current system features**

The software system that is the subject of this work is a shared portal for students and supervisors to manage the process for bachelor's and master's theses. Brought into service in Spring, 2012, through the tool, students and instructors are able to collaborate on thesis projects from idea inception through thesis completion and archival.

At a high level, the reconstruction of the structural aspects of a software system through *structural re-documentation* could be utilized [7] while on the lowest level, its software code itself could be used [10]. In this work, a middle road between the two—the implemented system itself—was evaluated using both a Heuristic Evaluation [8] and a Cognitive Walkthrough of [6].

The task selected for both the Evaluation and Walkthrough was "Activation of Progress State Indicator", shown in Figure 1. The Progress State Indicator is a visual aid that shows the level of completion for a particular stage in the thesis process. Modeled on traffic lights, the red, orange, and green maintain their commonly understood meanings in the Progress State Indicator. This is meant as an easy to recognize sign for supervisors. However, because this is student rather than system-driven, there is an ethical component to the activity, where the student make a public claim about the status of their work.

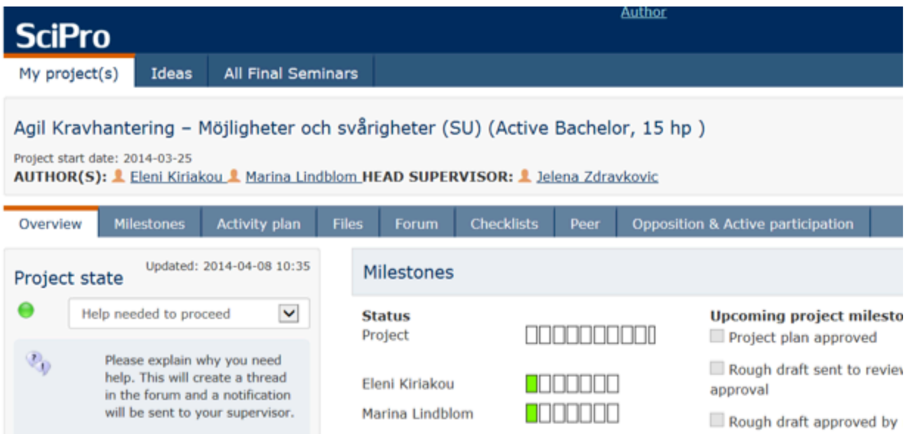

Fig.1. Student view of SciPro system with Progress Status Indicator highlighted

Table 2 displays a subset of the steps of the process used in the Walkthrough (Column 1), System Feature Utilized (Column 2) and Supported Heuristics (Column 3). Note that some steps are omitted from Table 2 to improve readability, but the full feature set is captured in the feature model (Figure 3).

Table 2: Abridged Data from Heuristic Evaluation and Cognitive Walkthrough utilized for "Creation of Progress State Indicator"

Walkthrough Step	**System Feature Utilized**	**Supported Heuristics**
Login (#1)	Single sign-on (SSO)	-Visibility of system status
Navigate to Project Overview page (#2)	User profiles that support Profile-based views	-Consistency and standards -Error prevention
Creation of Progress State Indicator (#6)	Profile-based activities	-Match between system/real world -Recognition rather than recall -Aesthetic and minimalist design

With the Evaluation and Walkthrough complete, a feature model (Figure 2) is developed utilizing the work of Kang et. al.,[5]. The Cognitive Walkthrough identified a series of actions that define a particular activity, and those system components that interact with and support that activity become the source for the feature model.

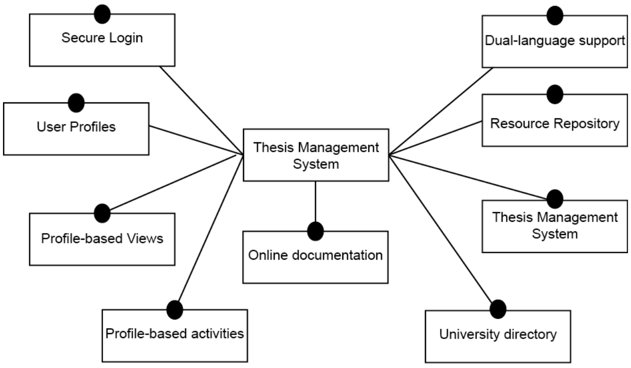

Fig. 2. Feature Model developed for "Creation of Progress State Indicator"

***Guideline 2*: Link features to user preference framework**

The Heuristic Guidelines of Nielsen used above are mapped to Holbrook's Typology [3] below in Table 3. The relationships are heavily dependent on the Dimensions of the typology; for example, Efficiency is Self-oriented, Active, and Extrinsic. This indicates that the feature benefits the user directly, is initiated by the user, and is utilitarian in nature.

Table 3: Nielsen's Usability Heuristics mapped to Holbrook's Typology of Consumer Values

Walkthrough Step	**System Feature**	**Supported Heuristics**	**Holbrook's Consumer Value**
Login (#1)	Single sign-on (SSO)	-Visibility of system status	-Efficiency -Ethics

Navigate to Project Overview page (#2)	User profiles that support Profile-based views	-Consistency and standards -Error prevention	-Efficiency -Excellence
Creation of Progress State Indicator (#6)	Profile-based activities	-Match between system/ real world -Recognition rather than recall -Aesthetic and minimalist design	-Aesthetics -Play -Esteem

***Guideline 3*: Generate i* model**

This guideline proposes the use of i* based on the Consumer Preference Meta-Model (CPMM) through mappings proposed in [9,13]. The i* framework was chosen in both works because it is designed for capturing intentions of a group of dependent actors, such as stakeholders in a requirements engineering process. Furthermore i* provides a rich modeling notation in this context [4].

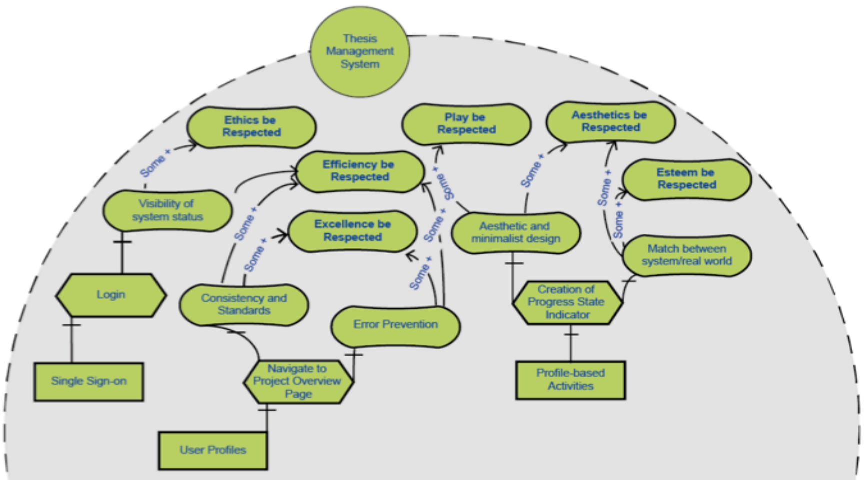

Fig. 3 SRM for Thesis Management System

4 Conclusions

We have proposed a step-based method for reverse engineering extant systems using feature modeling to elicit those stakeholder goals that reflect the consumer-oriented preferences the system is fulfilling. The primary motivation is the need for eliciting users' preferences in software systems in today's conditions, a move that will ensure systems' quality assurance, competitiveness, and openness, among others. The approach relies on a previous forward-engineering proposal for capturing different preferences of users for a system to configure lines of requirements for that system. The usability of the method has been illustrated with a case of a real system for thesis management expected to support the quality requirements mandated by the state Council for Higher Education. These requirements we have presented using Holbrook's consumer value framework [3]; yet in other business contexts, different frameworks can be considered.

The proposed method needs further research to facilitate a more systematic and more generic way for discovering the preferences of users of different kinds from existing software systems to be able, in a common lack of documentation for such qualities to assess system's compliance with needs and requirements in this direction.

References

1. Ali, R., Dalpiaz, F., & Giorgini, P. Location-based software modeling and analysis: Tropos-based approach. In Conceptual Modeling-ER 2008 (pp. 169-182). Springer (2008).
2. Gordijn, J., Akkermans J.M., and van Vliet J.C. Business Modeling is not Process Modeling, Conceptual Modeling for E-Business and the Web, LNCS 1921 (pp. 40-51). Springer (2000).
3. Holbrook, M.B. Consumer Value: A Framework for Analysis and Research. Routledge. London, GBR (1998).
4. i* Wiki, http://istar.rwth-aachen.de/tiki-view_articles.php, last accessed 2014-03-28.
5. Kang K.C., Cohen S.G., Hess J.A., Novak W.E., Peterson A.S. Feature-oriented domain analysis (FODA) feasibility study, Technical report No. CMU/SEI-90-TR-21. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (1990).
6. Lewis, C. and Wharton, C. Cognitive Walkthroughs in Handbook of Human-Computer Interaction, Helander, M. G., Landauer, T. K., & Prabhu, P. V. (Eds.). Elsevier (1997).
7. Müller, H. A., Tilley, S. R., & Wong, K. Understanding software systems using reverse engineering technology perspectives from the Rigi project. In Conf of the Centre for Adv. Studies on Collab. Research: Software Eng.Vol. 1 (pp. 217-226). IBM (1993, October).
8. Nielsen, J., & Molich, R. Heuristic evaluation of user interfaces. In Proc. of the SIGCHI conference on Human factors in computing systems (pp. 249-256). ACM (1990, March).
9. Svee, E.-O., Giannoulis, C., & Zdravkovic, J. Towards Consumer Preference-Aware Requirements. In Advanced Information Sys Eng. Workshops (pp. 531–542). Springer (2012).
10. Yu, Y., Wang, Y., Mylopoulos, J., Liaskos, S., Lapouchnian, A., & Sampaio do Prado Leite, J. C. Reverse engineering goal models from legacy code. In 13th IEEE Int'l Conf. on Requirements Engineering. (pp. 363-372). IEEE (2005, September).
11. Yu, Y., Mylopoulos, J., Lapouchnian, A., Liaskos, S., & Leite, J. C. S. P. From stakeholder goals to high-variability software design. Technical Report CSRG-509, University of Toronto (2005).
12. Yu, Y., do Prado Leite, J. C. S., Lapouchnian, A., & Mylopoulos, J. Configuring features with stakeholder goals. In Proceedings of the 2008 ACM symposium on Applied Computing (pp. 645-649). ACM (2008, March).
13. Zdravkovic, J., Svee, E. O., & Giannoulis, C. Capturing consumer preferences as requirements for software product lines. Requirements Engineering, Journal 10.1007/s00766-013-0187-2, (pp. 1-20). Springer (2013, November).