Bringing Usability to the Early Stages of Software Development

Luiz Marcio Cysneiros and Andre Kushniruk
Department of Mathematics and Statistics
York University
cysneiro@mathstat.yorku.ca

Abstract. *Usability has been increasingly recognized as an important factor in the acceptance of systems by end users. Usability requirements can be considered to be requirements that capture the usability goals and associated measures for a system under development. In order to ensure usable systems we must ensure identification of appropriate requirements regarding these critical aspects of systems. There is a basic need for systematic approaches to reason, model and analyze usability from the early stages of the software development. Furthermore, it is necessary to develop a usable ontology or classification of measurable aspects of usability that can be used to aid in the specification of usability requirements. These ontologies should be represented in a way that facilitates their use as guidelines for the requirements elicitation process. This work builds on review of literature in the area of human-computer interaction and of usability engineering in developing a catalog of aspects of usability that can be considered during requirements gathering. This catalogue is used to guide the requirements engineer through alternatives for achieving usability. The approach is based on the use of the **i*** framework, having usability modeled as a special type of goal.*

1. Introduction

Due to the enormous pressure towards deploying software as fast as possible, functionalities have been the main focus of software development process at the expense of implementing non-functional requirements (NFRs) such as performance, safety, accuracy and usability. Among these NFRs, usability has been pointed out as being one of the most important factors in the acceptance of systems by end users. Usability requirements can be considered to be requirements that capture the usability goals and associated measures for a system under development [3][5]. Usability goals may include a range of system aspects related to effectiveness, efficiency, learnability and memorability.

Usability aspects may strongly influence future design decisions. For example, if one is developing a system to help physicians to retrieve possible treatments for a disease from many existing databases, usability aspects may play a crucial role. In this case, the use of a hand-held PDA (Personal Digital Assistant) for deploying a decision support system to end users could prove to be an imperative, which might demand the use of alternative system architectures.

However, usability does not live alone. When one chooses to achieve one usability goal in one way he could be introducing conflicts with other NFRs. One of the main characteristics of NFRs is that they frequently conflict with other NFRs [1][2]. For example, when one decides to use a double password check for ensuring security this might bring considerable damage to usability concerns, since remembering two different passwords can be difficult for many people. Usability, as well as other Non-Functional requirements should be dealt with from the early stages of software development [1][2].

This work builds on review of literature as well as on some personal experiences aimed at developing a catalog of aspects of usability that can be considered during requirements gathering. We use this catalogue to produce a systematic approach to guide the requirements engineer through alternatives for achieving usability. Each alternative is modeled in order to allow the requirements engineer to evaluate these alternatives in contrast with other NFRs.

Nowadays, many application domains involve humans, hardware and software interacting in complex ways to achieve shared goals while allowing local autonomy. Agent orientation has been proposed as a suitable approach for requirements engineering [7]. Agent abstractions are used to hide the detailed actions within the agent's discretion. The agent-oriented modelling approach offers a higher-level description that is a more faithful representation of reality as real-world agents frequently depart from standardized routines and procedural process specifications. Because of that we based our work on the ***i**** framework [6], having usability modeled as a special type of goal.

2. Usability Ontology

In our approach, usability is interpreted by refining it into subgoals and subsubgoals, eventually linking them to implementable mechanisms. Various subgoals and mechanisms may contribute to usability to varying degrees. Each stakeholder's interpretation of usability may lead to different goal refinements and mechanisms. The various interpretations of usability can be collected and organized into a catalogue for reference during requirements elicitation, analysis and design.

Having this catalogue available, we can reuse its knowledge or add new knowledge to it. This knowledge will be represented using a graph structure to allow the representation of the organization's knowledge starting with the higher-level goals to achieve usability. The catalogue also allows for representing different ways of achieving the same goal so one can choose the way that is best suited to the domain being analyzed. Along with the operationalizations for usability, we also show possible correlations to other, possibly conflicting requirements. By doing so we are able to show that one specific solution might achieve usability and contribute positively or negatively to other requirements such as performance and security. The catalogue can be found at: http://www.yorku.ca/~cysneiro/nfrs/usability/usability.pdf

The catalogue was built using ***i**** constructs including: softgoals, goals, tasks, and beliefs [6]. The softgoal concept is used in ***i**** to express non-functional requirements. NFRs frequently interact with each other in complex ways. Qualitative reasoning can be carried out using contribution links among softgoals. The semantics of the links are based on the concept of satisficing[1] [1]. The most common contribution types are Help/Hurt (positive/negative but not sufficient to meet the parental goal), Some+/Some- (positive/negative of unknown degree), whereas Make/Brake indicates positive/negative of sufficient degree. Although these distinctions are coarse grained, they are enough to decide whether we need further refinement and search for more specific softgoals and operationalizations or not. Contribution links allow one to decompose NFRs to the point that one can say that the operationalizations to this NFR have been reached (i.e., the goals are no longer "soft"). Operationalizations can be viewed as functional requirements that have arisen from the need to meet NFRs.

3. Modelling Usability

To elicit usability concerns we visit every agent asking ourselves and the stakeholders what usability aspects should be of concern to this agent. For that purpose we use the catalogue shown in Section 2 for guidance.

Let us take for example the health care domain. In this domain, one of the most important relationships happens between the Patient and the Physician. Patients expect to be assessed by Physicians who in turn expect Patient to follow a prescribed treatment that might be composed, among other things, of drugs, exercise and specific diet. Following a treatment may pose several challenges such as changing the diet constantly or altering dosages of medication to comply with unusual situations.

One way of helping patients to follow a treatment could rely on providing the patient with a software agent to assess the patient to change diet or medication. This could be called the GA-Patient.

To assess usability concerns for the GA-Patient, we may for example ask if the GA-Patient would have any specific concerns for Usefulness (the words in Arial font are goals/subgoals extracted from the catalogue). Being a software agent that will help monitoring Patient's progression of a chronicle disease and suggest changes in treatment, this software agent is expected to be used several times a day. It is also expected to interface with devices for monitoring diseases as glucose meters and blood pressure meters. Therefore, Usefulness is a major concern. To assess that, we determined that among the possible ways of satisficing Usefulness, Improve Cognition would be required leading to a further refinement for Assure Timeliness of Information. To guarantee that, we may have to Provide Continuous Access to information. We can do that by either Using a Desktop or by Using a PDA. However, from the catalogue we see that the use of a desktop could hurt Availability of Information because for many applications as in health care or in the retail industry, the need to reach a desktop in order to enter or consult data can be highly inconvenient. On the other hand, Using a PDA contributes positively (help correlation link) to Availability of Information. Since a patient can carry the PDA all the time, including when visiting a physician, having the software agent in a PDA will allow the patient to be the real owner of his medical records and to have them available constantly.

4. References

[1] Chung, L., Nixon, B., Yu, E. and Mylopoulos,J. "*Non-Functional Requirements in Software Engineering"* Kluwer Academic Publishers 2000.

[2] Cysneiros,L.M. and Leite, J.C.S.P. *"Using UML to Reflect Non-Functional Requirements"* Proceedings of the 11th CASCON, IBM Canada, Toronto Nov 2001 pp:202-216.

[3] Hix, D. and Hartson, H.R. "Developing User Interfaces: Ensuring Usability Through Product & Process", John Wiley & Sons, 1993.

[4] Mylopoulos,J. Chung, L., Yu, E. and Nixon, B., *"Representing and Using Non-functional Requirements: A Process-Oriented Approach",* IEEE Trans. on Software Eng, 18(6), pp:483-497, June 1992.

[5] Shneiderman, B. "Designing the User Interface", Addison-Wesley, 1997.

[6] Yu, E. "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering" *in Proc. Of the 3rd IEEE Int. Symp. on Requirements Engineering*, pp:226-235, 1997.

[7] Yu, E. "Agent-Oriented Modelling: Software Versus the World" Agent-Oriented Software Engineering AOSE-2001 Workshop Proceedings. LNCS 2222.

[1] NFR can rarely be said to be satisfied. Goal satisficing suggest that the solution used is expected to satisfy within acceptable limits. The term satisfice was coined by Hebert Simon to express "good enough" alternatives.