Task-Driven Tools for Requirements Engineering

Pedro Campos

Department of Mathematics and Engineering,
University of Madeira, Portugal

E-mail: pcampos@uma.pt

Abstract

This research aims at designing and evaluating a new generation of usable and multimodal Requirements and Analysis Tools, capable of promoting artifact co-evolution in a useful manner, enabling cooperation and communication of multiple stakeholders over a common semantic model. The main goal is to leverage the elicitation of functional and non-functional requirements by using multimodal interaction techniques, and driving software development using a conceptual architecture easily extracted from user task flows.

1. Problem Statement

Requirements management and elicitation is widely recognized to be one of the major problems in modern software development. This stage of development involves multiple stakeholders, usually with different backgrounds, and is currently faced with the advent of multi-platform development [10]. In this context, new tools are required to enable cooperation and communication of multiple stakeholders over a common semantic model that is capable of driving modern software development. Lack of user involvement has traditionally been "the number one reason for project failure" and requirements engineering tools seem to have "the most significant impact on a project's success" [13].

The available Requirements Engineering (RE) tools are currently limited to modeling and management tools [14] that are hard to use and only of interest to disciplined engineers, leaving all other stakeholders (executives, marketers, clients and end-users) aside. The lack of adequate and usable tools has also been blocking technology transfer from academia to industry [14, 15, 6].

Co-authoring and co-evolution of requirements models are not adequately dealt with by current tools. Applying User-centered design to the design of new tools for promoting co-authoring and co-evolutionary development of requirements over a common semantic model could bring many benefits, such as increased stakeholder involvement and information sharing, increased traceability and usable ways to negotiate requirements as well as prioritize development tasks.

2. Prior Research

In the quest for more usable and useful requirements tools, architectures for intelligent support have been proposed [14]. These are believed to help tool developers build scalable, integrated and expanded tools. However, little attention is given to the usability aspects that need to be addressed in order to promote stakeholder involvement.

The ART-SCENE Scenario Presenter is a web-based tool aimed at providing support for scenario-based requirements engineering [12]. ARENA [7] provides negotiation methods to foster mutually satisfactory agreements between stakeholders and was based on the WinWin negotiation model [1], one of the most well-known attempts to support distributed requirements negotiation.

Recognizing the potential of mobile tools, Seyff et al. [11] proposed a tool using PDA's to foster the capture of requirements with an easier participation from stakeholders.

These examples demonstrate that there is a clear trend towards building a new generation of requirements tools, fully coupled with the whole software engineering process and fully supporting the work-styles [4] of all stakeholders in usable and useful environments.

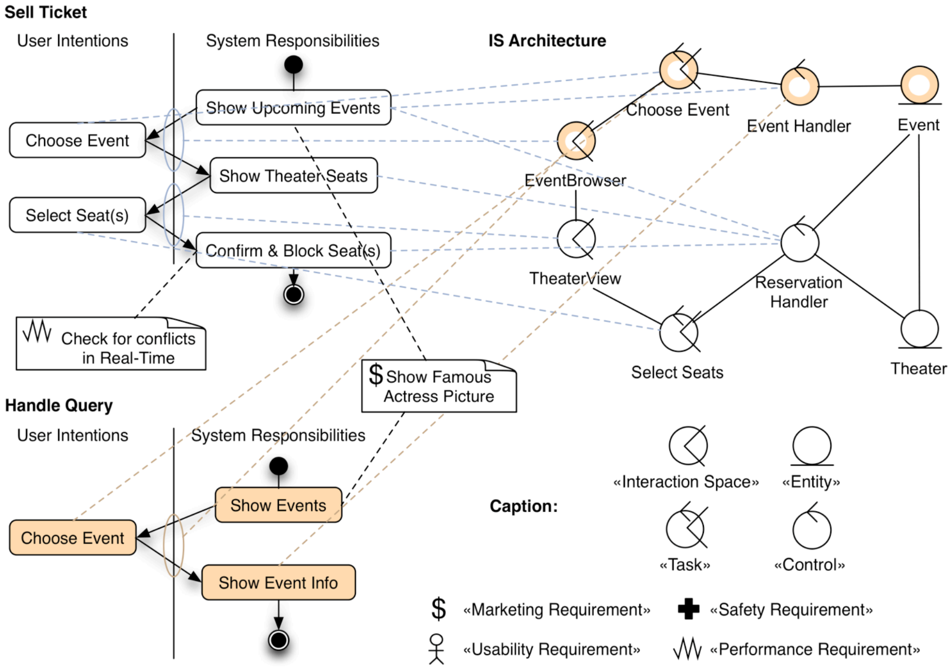

Figure 1: Tracing use cases and task activities to the conceptual architecture of a System.

3. Proposed Approach

3.1 Hypothesis and Goals

We hypothesize that (a) the structure of a system's use influences the conceptual system architecture and that influence can be used for requirements negotiation and prioritizing development tasks; (b) the UML can be successfully used as a common semantic model to facilitate communication and promote artifact co-authoring by different-background stakeholders, as long as this is supported by user-centered tools; and (c) requirements elicitation can be leveraged through multimodal, user-centered collaborative environments, not just descriptive modeling tools like the current ones.

The main goals of our research are:

- Achieve a technological solution for facilitating information sharing during requirements elicitation by a background-diverse group of stakeholders;

- Trace the requirements of a system, in terms of user intentions and system responsibilities, to the conceptual architecture of that same system, and easily extract that architecture from task flows;

- Ease the process of prioritizing development tasks as well as requirements negotiation by allowing all stakeholders to view the impact of a given set of use cases in the conceptual architecture of a system.

3.2 Method

To test our hypothesis, we propose to: (i) develop, adapt and evaluate notations from different fields (software engineering, marketing, management and usability engineering) that enable multiple representations of requirements over a common semantic model; (ii) design, implement and test a traceability model between different representations that enables synchronized artifact changes from the different stakeholders; and (iii) develop and evaluate prototype tools in the sequence of CanonSketch [2, 3], that support and demonstrate automated support for the methods, techniques and models.

Our methodology is based on Design Research [8], which has as final output an *instantiation* which "operationalizes constructs, models and methods".

TaskSketch. TaskSketch[1] focuses on linking and tracing use cases to the conceptual architecture of a system. The idea is to use the Wisdom extension to the UML [9], which can be summarized as the UML class stereotypes in the lower right part of Figure 1: «Interaction Space» models the interaction between the system and human users within the user interface of that system; «Task» models the structure of the dialogue between the user and the system in terms of meaningful and complete sets of actions required to achieve a goal; «Control» encapsulates complex derivations and calculations, such as business logic, that cannot be related to specific entity classes and «Entity» models perdurable, often persistent, information.

Figure 1 shows a simple example of two UML activity diagrams representing task flows of two distinct

[1] For more info such as screenshots and videos of the tool, please refer to http://dme.uma.pt/tasksketch

use cases in an Arts Center ticket selling IS[2]: Sell Ticket (in blue) and Handle Query (in orange). Each use case is detailed using two swim lanes: User Intentions and System Responsibilities (the tool also provides a participatory view and a narrative view). For example, in the use case "Sell Ticket", it is the system's responsibility to show upcoming events so that the user chooses one of them. The system then shows the available theater seats for that event and the user selects the desired seats, which are blocked by the IS. Each crossing of the swim lane originates an interaction space (Event Browser and Theater View). Each action on the User Intentions' swim lane corresponds to a task and each action on the System Responsibilities' swim lane is associated with a control. Figure 1 describes these relations for these two distinct use cases. Using color, the developer can look at the architectural view of the system and see which classes handle which use cases. This simple support to requirements traceability can be very powerful for, e.g., prioritizing development by deciding which classes are more urgent to implement. Figure 1 also shows an example of two non-functional requirements: a marketing requirement attached to the Show Upcoming Events action ("Show Famous Actress Picture") and a performance one ("Check for Conflicts in Real-Time"). Figure 2 shows part of the tool.

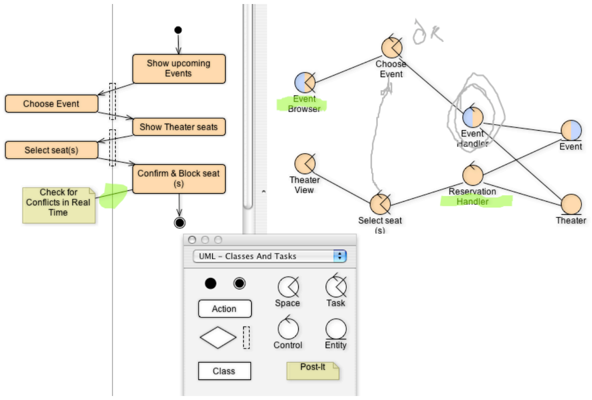

Figure 2. The TaskSketch tool.

Through easy drag-and-drop between the views, the conceptual architecture of a system can be easily extracted from the use cases' task flows.

Requirements Elicitation. We are also exploring the possibilities offered by gesture recognition, mixing formal and informal notations and collaborative development using speech recognition and a shared display.

In this context, there is evidence [5] that real-time collaboration tools incorporating speech recognition and displaying information about a group's dynamics can positively impact the group's interaction. In some decision tasks, in particular during requirements elicitation, there is a risk that some stakeholders holding important information will not effectively share it, thus leading the team to less informed discussions.

In the "Brainstorm Environment" we propose (as part of the TaskSketch tool mentioned above), each stakeholder is associated a color and types in ideas for requirements of the system being developed. Every time someone sends a requirement to the screen, a shape color-coded by the user who sent it starts slowly falling through the center of the window.

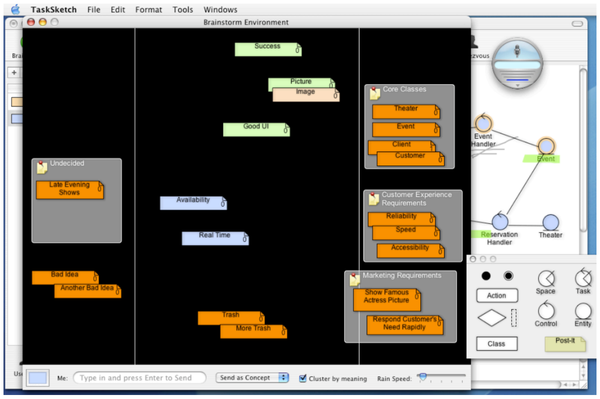

Figure 3. Collaborative Elicitation screenshot.

The content of this shared display can be manipulated by anyone, so it becomes useful to cluster concepts manually. Dragging a shape to the left or right sides of the window makes it stop falling. Concepts that remained untouched become grouped in the bottom of the window [5]. Clustering of requirements can also be made partially automatically, because this system uses a thesaurus and every time someone sends a common concept, such as "client" and "customer", the two shapes become aggregated. The speech recognition system is set to dynamically recognize any of the phrases or words in the shared display. Every time a concept is recognized, the shape shows a number, which counts the number of times that concept has been spoken during the meeting. Figure 3 shows the look of this environment.

We foresee the following usage scenario for an environment like this one: different stakeholders meet to discuss and elicit functional and non-functional requirements. This includes clients, marketers, programmers and interaction designers. Each uses a microphone and has its own color. As they suggest ideas, they watch them fall and the display becomes color-filled. In this manner, the system will attempt to in-

[2] This example was provided by L. Constantine and was thoroughly used and implemented in both MSc. and BSc. HCI courses at the University of Madeira.

crease the discussion of ideas as well as to foster collaboration between people with different backgrounds through an engaging environment. It is also expected that under-speakers will participate more and over-speakers will participate less, like [5] have shown. However, by the end of the meeting, it is also expected to achieve a better clustering and definition of concepts as well as have an idea of what requirements and concepts are more important (by looking at which words were more referred to during a given context in the meeting).

4. Current Progress and Expected Contributions

The TaskSketch tool as described here is almost completely implemented and ready for usability tests. A predecessor tool, CanonSketch [2, 3, 4] was already tested and results were promising.

Our contributions can be summarized as follows:

1. A different way to leverage requirements elicitation between different-background stakeholders, taking advantage of user-centered design techniques and informal, multimodal interaction (using speech, gestures and shared displays);
2. Innovative tools fully supporting these methods and techniques;
3. A general development framework for RE tools, based on all lesson learned during the research.

5. Evaluation

We have already conducted several usability studies during HCI courses at our University and the results seem promising. We plan to test our hypothesis through:

- Empirical validation over the usability and effectiveness of our tools and methods; this will be accomplished mainly through usability studies with different-background users; we will take measures such as user's satisfaction, error rates, expressiveness and ease of communication.
- Proofs of concept, in which we apply real-world problems (such as [10]) to our tools and methods.
- Testing and assessing the methods, techniques and tools in an industrial setting.

References

[1] B. Boehm, A. Egyed, J. Kwan, D. Port, A. Shah, and R. Madachy. "Using the WinWin Spiral Model: A Case Study". *IEEE Computer*, July 1998, pp. 33-44.

[2] P. Campos and N. J. Nunes. "CanonSketch: a User-Centered Tool for Canonical Abstract Prototyping". *Proceedings of the International Workshop on Design, Specification and Verification of Interactive Systems*, Springer-Verlag, 2004.

[3] P. Campos and Nunes, N. "A UML-Based Tool for Designing User Interfaces". *UML Modeling Languages and Applications: UML 2004 Satellite Activities*, 2004, Revised Selected Papers, Springer-Verlag LNCS Vol. 3297.

[4] P. Campos and N. J. Nunes. "Galactic Dimensions: a Unifying Workstyle Model for User-Centered Design". *10th IFIP TC13 International Conference on Human-Computer Interaction, INTERACT 2005*, 12-16 Sep, Rome, Italy, 2005.

[5] J. M. DiMicco, A. Pandolfo, and W. Bender. "Influencing Group Participation with a Shared Display". *ACM Conference on Computer Supported Cooperative Work* (CSCW 2004). Chicago, IL.

[6] H. Kaindl, S. Brinkkemper and J. A. Bubenko Jr. "Requirements engineering and technology transfer", *Requirements Engineering*, 7(3):113-123, Sep. 2002.

[7] P. Grunbacher and P. Braunsberger. "Tool Support for Distributed Requirements Negotiation". *Cooperative Methods and Tools for Distributed Software Processes*, A. Cimitile, A. De Lucia and H. Gall, Eds., FrancoAngeli, Milano, Italy, pp. 56-66.

[8] S. March and G. Smith, G. "Design and Natural Science Research on Information Technology." *Decision Support Systems* 15 (1995): 251 - 266.

[9] N. J. Nunes and J. F. Cunha. "Wisdom: a Software Engineering Method for Small Software Development Companies". *IEEE Software*, 17(5):113-119, Sep.-Oct. 2000.

[10] L. Patrício, J. F. Cunha, R. Fisk and N. J. Nunes. "Customer Experience Requirements for Multi-platform Service Interaction: Bringing Services Marketing to the Elicitation of User Requirements". *Proceedings of the 12th IEEE International Requirements Engineering Conference*, 2004.

[11] N. Seyff, P. Grunbacher, N. Maiden, and A. Tosar. "Requirements engineering tools go mobile". In *Proceedings of 26th International Conference on Software Engineering*, ICSE 2004, pp. 713-714, 23-28 May 2004.

[12] A. G. Sutcliffe, N. A. M. Maiden, S. Minocha and D. Manuel. "Supporting Scenario-Based Requirements Engineering". *IEEE Transactions on Software Engineering*, 24(12):1072-1088, 1998.

[13] The Standish Group, *Extreme CHAOS*, Update to the CHAOS Report, http://www.standishgroup.com, 2001.

[14] Q. Zhang and Eberlein, A. "Architectural design of an intelligent requirements engineering tool", *Electrical and Computer Engineering*, IEEE CCECE 2003. Volume 2, 4-7 May 2003, pp. 1375-1378.

[15] Q. Zhang and Eberlein, A. "Deploying good practices in different requirements process models", *Proceedings of the 6th International Conference on Software Engineering and Applications*, pp. 76-80, 2002.