Use Case Maps: A Visual Notation for Scenario-Based User Requirements

A. Alsumait, A. Seffah, and T. Radhakrishnan
Human-Centered Software Engineering Group
Computer Science Department, Concordia University,
Maisonneuve Blvd. W. Quebec, Canada
{alsumait, seffah, khrishnan}@cs.concordia.ca

Abstract
Among their popular and potential applications, scenarios and use cases provide an integrative picture of the user tasks with its context (users and work). Our goal is to build a complete and consistent user interface and usability requirement model that is simple, intuitive, unambiguous and verifiable by extending the prospective standard the Use Case Maps (UCMs). In particular, we provide step-by-step guidance for developing UCMs from scenarios.

1 Introduction and Background

During the last two decades, several HCI research efforts tried to develop appropriate representations to support user interface and usability requirements. Among the various representations, scenarios and use cases were proposed as a working design representation of user experiences with, and reactions to, system functionality in the context of pursuing a task. Other efforts have tried to combine use cases and task analysis as vehicle for bridging the current gaps between user interface/usability and functional requirements. For example, Forbrig (1999) introduced a framework for combining task models, user models, and object models. These multiple dimensions require tools enabling the manipulation of all these models along the different phases of software development (Sutcliffe, 1998; Weidenhaupt, 1998). Related to this work, Constantine and Lockwood (1999) proposed a use case model description that does not contain any implicit user interface decisions; essential use cases. Later on, user interface designers can use these essential use cases as input to create the user interface without being bound by any implicit decisions. Our research aims to complement the scenario-based requirement approach using an extended use case maps (UCMs) notation. We describe the process of building two UCM models that provide an easy way to understand and validate the user interface and usability requirements.

2 The User Interface and Usability Requirement (UIUR) Model

The proposed UIUR model makes use of the prospective standard Use Case Maps (UCMs) to bridge the gap the between users' statements of requirements in an informal natural languages and the target of formal specifications. UCMs are intended to be a useful tool for functional requirement of scenario-based software development. However, their support for designing user interfaces is still acknowledged to be insufficient (Alsumait et all., 2002). Our strong belief that UCMs are powerful for user interface and usability requirements is based on the simplicity of UCMs notation (Buhr, 1998). The basic UCM notation is very simple (See Figure 1), and consists of

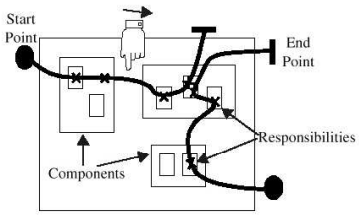

Figure 1: example of a Use Case Map (Buhr, 1998)

start-points (filled circles, representing pre-conditions), *responsibilities* (crosses, representing tasks to be performed), *end- points* (bars, representing post-conditions). The responsibilities can be bound to *components*, which are the entities or objects composing the system. The wiggly lines are *paths* that connect start points, responsibilities, and end points. UCMs notation is easily understandable by both the user and the user interface designer. In fact, this helps the designers to handle different users' understanding and expectation of the interface, and bridge the gap by refining the requirements earlier. Moreover, several studies are focusing on how to integrate UCMs to UML and how to formalize UCMs in XML (Buhr, 1998). Finally, we will show that by incorporating the extended UCMs to create our UIUR, we can express user interface and usability requirements much better.

2.1 UIUR Process and Notation

The proposed UIUR model divides the process of transforming requirements stated described in an informal language to formal specifications through a number of iterative phases. Each phase aims at increasing the formality of representation by a small step. The use of UIUR model includes four typical phases: (1) scenario analysis, (2) conceptual use case maps, (3) physical use case maps, and (4) formal specification.

2.2 Scenario Analysis

At this phase, only very limited information is available. A process model of scenario analysis consists of the following iterative and interactive activities: identification of scenarios, elicitation of information and description of scenarios as well as building use cases.

2.3 Build Use Case Maps

By extending the UCM notation, we find that it can be used in the following four dimensions:

- Task Dimension: *represents tasks that are relevant for interactions. Thus, UCMs are used as a simple and expressive visual notation that allows describing task scenarios at an abstract level in terms of sequences of responsibilities and tasks over a set of components.*
- Dialog Dimension: *explains the style of human-computer interaction and also describes the sequence of dialogs that can take place between the user and the system. New notations are introduced to provide UCMs with more expressive power for interaction design (Alsumait et. al, 2002).*
- Structural Dimension: *identifies the objects comprising the user interface, their grouping and specifies their layout, e.g. by indicating approximate placement or by indicating topological relations between groups.*
- Usability Dimension: *Measure the usability of the use case maps by using usability metrics. Different metrics such as task simplicity and task performance can be used to measure the usability of the use case map early in the requirement phase.*

In the second phase, two types of UCMs are created: the conceptual use case map *(CUCM) and the* physical use case map *(PUCM). The CUCM and PUCM together integrate the four dimensions stated above and capture a complete picture of user interface and usability requirements. This helps in analyzing the consistency between requirements of different use cases, and discovers if any conflicts between different types of users, different purposes of use and different operating conditions.*

Conceptual Use Case Map (CUCM)

The CUCM helps provide details on what information will be needed from the user and the system to accomplish a task. Steps to create CUCM are:

- Partition the use cases: *Consider only the use cases with human actors.*
- Create Task Model: *Decompose use cases into tasks and subtasks.*
- Define components of the use case: *Tasks can be bounded to components. This will help in determining the entities and objects that compose the system*
- Create Dialog Model: *Tasks represent decision points in the use case. At any decision point, a dialog will take place between the system and the user.*
- Analysis of the consistency: *Two properties of consistency are examined. The consistency among a set of use cases where the requirements captured by a set of use cases are not in conflict with each other. The consistency of a use case with respect to a requirements model where the information contained in the use case does not conflict with the requirements model and the information contained in the use case is a subset of the information contained in the model. Both properties can be validated in our proposed use case model. If the use cases are inconsistent, it will backtrack to the information elicitation and scenario description (phase 1) to resolve the conflicts*
- Analysis of completeness. *Completeness is the property where all the information contained in a requirement model is covered by at least one use case and does not contain any information that cannot be inferred from the information contained in the use cases. If incompleteness is discovered, it will backtrack to the scenario identification phase to identify additional scenarios that will cover the missing situation.*

Physical Use Case Map (PUCM)

The extended UCMs not only describes the sequence of tasks and dialogs that can take place between the user and the system but also help to understand and reason about the requirements of the user interface, including usability aspects. The PUCM represents the space within the user interface of a system where the user interacts with all the functions, containers, and information needed for carrying out some particular task or set of interrelated tasks. Moreover, successive display of different screens and interactive objects are presented. The PUCM can greatly benefit from the graphical representation of use cases. Steps to create PUCM:

- *Identify the objects that make of the user interface.*
- *Convert the use case maps into an active prototype using a graphical user interface development tool.*
- *Measure the usability of the screens*

Further information on validating CUCM and PUCM can be found in (Seffah and J.A Poll, 2003). Once the CUCM and PUCM are constructed and validated, further requirement analysis and specification activities can start to produce formal requirement specifications according to the requirement models. More research and investigations is required in this area.

3 A Case Study- Movie Recommender Software (MRS)

This section illustrates the description and analysis of the UIUR model using a case study that simulates Movie Recommender Software (MRS) for a PDA. The software should provide recommendations for movies based on preferences of other users.

The movie Recommender software allows the user to carry out the following scenarios: (1) View a list of the latest movies by selecting (Any Genre, Action, Comedy, Drama), (2) Search for movies of (any genre, For Action, For comedy, For drama), (3) View or add a movie to the (To See) list; a list of movies to be seen in the future, (4) Rate a movie. On choosing the options 1 or 2, the user can get detailed information of a movie generated by the Recommender system

As shown in Figure 2, the MRS system can be decomposed into five main tasks, (1) view latest movies, (2) search for movie recommendations, (3) create user to see list, (4) rate a movie, and (5) view movie detailed information. Different scenarios can be drawn, and details are delayed to sub-UCMs. Figure 3 depicts the second level of the requirement model when a user rates a movie. A *dialog* notation illustrates that a conversation between the user and the system is taking place. The CUCM should be validated against the scenarios by analyzing whether the use cases are consistent with each other. Examples of such validation are shown in (Seffah and J.A Pool, 2003).

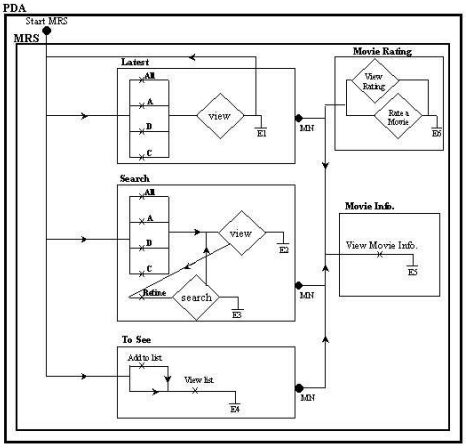

Figure 2: **CUCM for the movie Recommender**

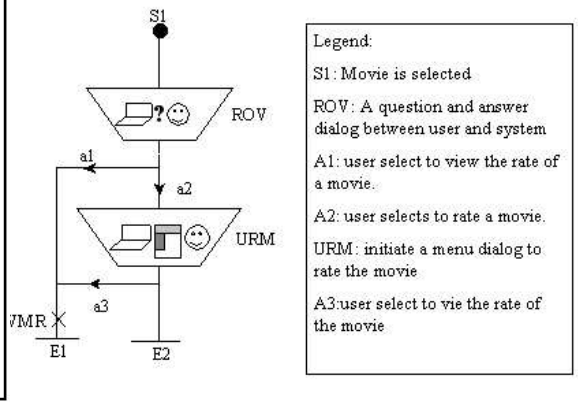

Figure 3: **UCM for movie rating**

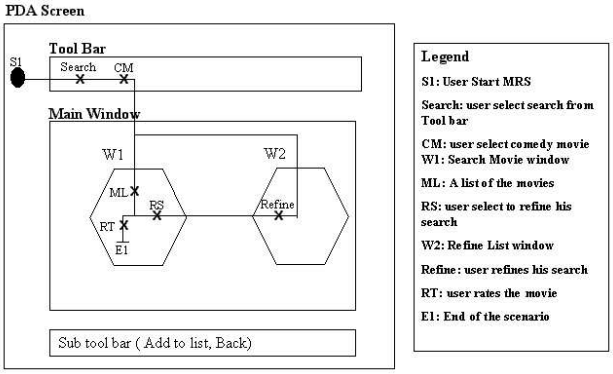

Figure 4: **PUCM for movie Recommender software**

The PUCM in Figure 4 describes the scenario where a user searches for a certain comedy movie and rates the movie. The user interface consists of a tool bar, Main Menu Window, and a sub-

toolbar. Next, the PUCM can be converted into a prototype. Both the PUCM and this prototype can be used by usability expert to predict the usability of the interface.

4 Conclusions

Our research effort leads to the development and analysis of a complete model for user interface and usability requirements (UIUR). This model ensure that: (1) a consistent and complete requirement specification can be captured using scenarios, (2) the specification is a valid reflection of user requirements, (3) the derivation of early design artifacts such as low fidelity prototypes is possible. The new model extends the UCMs to accommodate four dimensions of requirements including the task, the dialog, the structure, and the usability of the user interface. New notations are introduced to provide UCMs with more expressive power for interaction design. As a consequence, the extended UCM helps not only describe the sequence of tasks and dialogs that can take place between the user and the system but also assists to understand and reason about the requirements of the user interface, including usability aspects. Future work includes more investigation on formalizing the requirements according to the UIUR model.

References

Alsumait, A.; Seffah, A.; Radhakrishnan, T. (2002), Use Case Maps: A Roadmap for Usability and Software Integrated Specification, In: *Proceedings of IFIP World Computer Congress 2002*, August 25-30, Montreal, Canada.

Buhr, R. (1998), Use case maps as architectural entities for complex systems, *IEEE Transactions on Software Engineering*, 24(12), 1131-1155

Constantine, L.; Lockwood, A. (1999), Software for use: a practical guide to the models and methods of usage-centered design. Reading, Massachusetts: Addison-Wesley.

Forbrig, P. (1999), Task and object-oriented development of interactive systems - how many models are necessary? In: *Proceedings of Design Specification and Verification of Interactive Systems Workshop-DSVIS'99*, Braga, Portugal, 225-237.

Jarke, M. (1999), Scenarios for modeling. *Communications of the ACM*, 42(1), 47-48.

Seffah A., J.A, Pool Combining UCMs and Formal Methods for Representing and Checking the Validity of Scenarios as User Requirements, to appear in interact 2003. Human-Centered Software Engineering Technical Paper 01-0203, Concordia University

Sutcliffe A.G., Maiden N.A., Minocha S. & Manual D. (1998), Supporting scenario-based requirements engineering. *IEEE Transactions on Software Engineering,* 24(12), 1072–1088.

Weidenhaupt K., Pohl K., Jarke M. & Haumer P. (1998), Scenario usage in system development: a report on current practice, *in Proceedings of the International Requirements Engineering Conference (ICRE '98).*