Interactive Use Case

Diego Calzà
Lab.of Informatic Engineering & Organisational Analysis
Dept. of Sociology and Social Research,Univ. of Trento
via Zeni, 8 – I-38068 Rovereto – Italy
+39 (0464) 443131
diego@lii.unitn.it

Vincenzo D'Andrea
Dept. of Information and Communication Technology
University of Trento
Via Sommarive 14 – I-38050 Povo (TN) – Italy
+39 (0461) 882084
dandrea@dit.unitn.it

Arthur B. Baskin
Intelligent Information Technologies
9801 Fall Creek Road, #414
Indianapolis, IN 46256 – USA
+1 (317) 546 6337
abaskin@intelligent-it.com

Gianni Jacucci
Lab.of Informatic Engineering & Organisational Analysis
Dept of Sociology and Social Research, Univ. of Trento
via Zeni, 8 – I-38068 Rovereto – Italy
+39 (0464) 443131
gianni@lii.unitn.it

ABSTRACT

In this paper we present a proposal for a pictorial and interactive tool positioned in between Use Cases and Mock-up. We call the tool "Interactive Use Case" (IUC), underlining the participatory nature of system definition. The IUC can provide a common place to structure and record the dialogue between system developers and system users. We describe a case study, in which IUC were used in the analysis of a complex information system where the number of factors to be taken into account and their variability are extremely high. The information system was used to support the definition of meta-models, which described cost analysis, management and reduction for maintenance, repair and operations. Because the information system is used for a variety of customers, it must be able to adapt to different maintenance and repair practices. Interactive Use Cases played a critical role during the participatory definition of this system.

Categories and Subject Descriptors

K.6.1 **[Project and People Management]**: *Systems analysis and design, Life cycle.*

General Terms

Management, Documentation, Design, Human Factors.

Keywords

Participatory Methodologies, Use Cases.

1. INTRODUCTION

Our Laboratory espouses the Scandinavian Participatory Design (PD) approach to the development of information infrastructures for organizations. It focuses on design for change, on design for accountability, and on design for end-user design in use. Accountability and end-user design are strongly related concepts.

In PDC-04 Proceedings of the Participatory Design Conference, Vol 2, Toronto, Canada, July 27-31, 2004, under a Creative Commons license. CPSR, P.O. Box 717, Palo Alto, CA 94302. http://www.cpsr.org ISBN 0-9667818-3-X

In both, users and technology must thoroughly understand each other and be able to communicate effectively.

Mock-Ups and Use Cases have been used as boundary objects for communication between designers and users. In this paper we explore the hybrid concept of Interactive Use Cases. Structurally, IUC can be viewed as something between traditional Use Cases and Mock-Ups, and they can function as a boundary object between the users and the technology itself.

Participatory methodologies represent a modern approach to Information Systems (IS) design; in this approach the future users of the system play a crucial role in its design [1, 4, 5]. Participatory methodologies try to provide the worker with better tools to perform his or her work – not replace the human worker. The goal is not limited to improving the design of IS, but extends to the social aspects of a project, with special attention to the future users of the system. The final users are of primary importance both from a social and operational point of. They work with designers that are improving work practices as well as of building a product.

There are two fundamentals concepts governing the implementation of participatory methodologies: reciprocal learning and design-by-doing. Reciprocal learning reflects the need for users and developers teach each other work practices and technological potentialities, sharing a common experience. In design-by-doing, hands-on design is a way to represent reality by means of an abstraction and modeling and by interactive experimental testing of emerging system elements.

According to Kensing and Munk-Madsen [9] there is no failsafe technique for the active involvement of users. They maintain that the real problem is communication. The difficulty is to find a common language for dialogue and for sharing the design context. The interchange between actors sets the stage for the so called "language game". Without a shared formal language, the users and designers may not be able to effectively communicate.

In a large number of cases, software development is based on formal or informal specifications of requirements. The usage of specifications is bound to an analytical operating mode, used by developers, trying to reduce complexity by means of abstractions.

Without mockups or prototypes, users often have difficulties bridging the gap between an abstract description of use, and the knowledge and professional competencies characterizing the use itself [7]. Participatory processes involve human resources, technology, and the exchange of knowledge. There are several actors that need to communicate by means of media that have to be equally understandable on all sides of the communication. As mentioned above, each actor has its own language, way to reason or express concepts.

In our approach, we use Interactive Use Cases as Boundary Objects (BO) to facilitate the communication between users and designers. According to Leigh Star and Griesemer ([10] - page 393), boundary objects can be real or abstract objects, with different meaning in different social worlds, with structures close enough to the different worlds to be recognizable. Their creation and management becomes a key process in developing and maintaining the coherence between intersecting and changing social worlds. It is of paramount importance that persons belonging to different communities of practice be able to use a BO in terms of what Chrisman [3] calls a "common point of reference" in conversations. The main property characterizing a BO is the capability of being understood on both sides. Each actor has to be certain that the boundary object expresses the concept they have in mind. People are use BO as "coordination and alignment media" [6]. The set of people with which it is possible to interact by means of a BO has different competences and goals, the BO must then satisfy all the different worries that these people could have. Finally, BO should be flexible, so that they are adaptable to the evolution of their needs.

2. INTERACTIVE USE CASES

2.1 Use Case

The traditional role of Use Cases is for describing the behavior of a system in response to requests of system users [8]. They are used during the phase of requirements gathering, in the lifecycle of a system, in the Object-Oriented Analysis and Design methodology. In our participatory methodology, we propose a role for Use Cases and Interactive Use Case as boundary objects between users and developers of an information system.

To create a Use Case it is necessary to identify the human and non-human actors that will use the system, their goals in the system usage, and the scenarios for achieving these goals with the system. Scenarios are defined in terms of a precise sequence of steps, identifying input from the actors and the corresponding system's response.

Reasoning with the users in terms of Use Cases helps the developers in the task of identifying and clarifying the system requirements. Their narrative nature helps the user to understand the system roles and actions in terms of concrete contexts of usage. Instead of focusing on the internal structure of the system and its constituents (the design view), the Use Case takes a "black box" approach, where only the system behavior is important.

The tools generally used for representing the Use Cases are both textual (usually a sequence of steps) and diagrammatical (a visual synthesis of the system usage). The Unified Modeling Language (UML, [11]) assigns a precise and relevant role to Use Cases, precisely describing the tools and methodologies for creating them. Use Cases are generated starting from one or more user requirements (what the users ask in terms of system functions and behaviors) and have a central role in clarifying the requirements as well as in creating more requirements. It is also common to measure the complexity of an information systems in terms of the total number of Use Cases addressed by the system and to refer to Use Cases also in later stages of the system lifecycle, to describe how the system behavior relates with the users' requests ("this screen is related to these Use Cases ...").

2.2 Mock-up

A mock-up is an artifact representing a subset of the functionalities of an Information System. The mock-up is built before the actual implementation of these features in the system or in a prototype. While there could be several reasons for building a mock-up, in the perspective of our work their main role is as vehicles of user involvement. Revising the system with the user, with the help of a mock-up, is an important part of our participatory methodology.

There are several benefits of using a mock-up. The existence of the artifact helps the user to visualize the context of use of the system and to point out inconsistencies or revisions in the requirements. The user can situate the mock-up in her work practice and compare it with the current or planned activities.

A mock-up is usually presented with an interface design close to the one being planned for the system. Graphical as well as logical inconsistencies can thus be identified. While a mock-up is generally built with software tools, it could actually be realized with any kind of material.

In the view of Bödker and Grönbaek [2], the mock-up is considered as a tool for an exploratory approach, used for simulation of the system or as a prototype that will be thrown away after use. An exploratory approach proposed by these authors is Cooperative Prototyping – users and designers cooperate actively in the design, using prototyping as a way to interact. In the Participatory Design discourse this is a situation of reciprocal learning, where the users and the designers can learn from each other [12].

2.3 Interactive Use Cases

Interactive Use Cases (IUC) are derived from descriptive and diagrammatic use cases, as a tool for the analysis of an information system. The goal is to provide the users with a way to express their approach to work with the future system. This is done during analysis sessions, including users and developers, in order to create a shared, reciprocal knowledge. IUC are created from scratch during these sessions: the discourse between users and developers is then visualized by IUC.

IUC incorporate the system logic and a description of the parts that will compose the system. IUC use a graphical tool that resembles the window of an application program. The window is showing the grouping of data categories (text fields) and action starters (buttons) that belong to a use case or to a subpart of it. Notes and comments can be superimposed to the window's elements. A minimal functionality is provided; that is moving from one use case to another (or moving between subparts of a use case) by means of buttons in the window.

The difference with the descriptive, textual use cases is of course the addition of a complete pictorial representation of the use case, with an interface familiar to the user experience. A long description of input data or of possible departing actions is easier to grasp, according to our experience, if it is presented in terms of a (fake) application window. IUC are produced and modified during sessions with the users.

The graphical quality of the window is extremely rough and simple on purpose, in order to avoid a premature definition of the user interface. IUC windows should be considered as "conceptual containers" rather than representation of interactive systems. The role of such a window is to collect pictorial representations relevant to a textual use case. IUC have the procedural aspects of narrative use cases and the presentation aspects of mock-ups. They are an hybrid tool, blending the creation of appropriate representations and the elicitation of the action sequence.

Users actively participate in creating windows that are used as logical grouping of system features and data, putting in context and revising the application business rules. Users are also involved in selection between different options in the system development, seeing them "live in action" during the analysis sessions with the developers. Non-functional requirements can be easily integrated as notes in the pictorial representation of IUC.

IUC can be archived, along with session notes, providing a way to reconstruct the path bringing to a specific decision on the system. Developers' analysis of that data can allow the early identification of the need for system extensions or the potential directions for system evolution.

IUC can be prepared with a variety of tools, from programs to prepare presentations, to HTML, to development environments. The relevant factor is the ability to quickly modify and annotate the IUC during the sessions with the users. The use of IUC as the center of the discussion and the clarifications between developers and users was a powerful tool in the case study described in the following sections of this paper.

3. THE CASE STUDY

CMN (not the real name) is a consulting company, working on the costs analysis, management and reduction for maintenance, repair and operations (RMO) CMN consultants define the most convenient purchasing scenarios for a customer. In CMN business model, their income is depending on customers savings.

Today, CMN consultants are the owner of all the knowledge needed for defining the best purchasing scenario. The scenario depends on measurable variables (product categorization, customer purchasing policies, market trends, temporary promotions, ...) as well as on non measurable ones (a change in the management, a new actor entering the market, ...).

CMN is currently trying to consolidate the current business process and model, and to exploit the potentialities of the Internet for new business models. They need an information system to support the definition of meta-models, making it possible to operate with a new business model. Using meta-models, customers can create and maintain the models for their own processes, shifting the role of CMN consultants to certifiers for products and processes.

The company requires a knowledge management and decision support system centered on purchasing scenario, that in turn contain products, variables affecting their price, and business rules for identifying the best purchasing conditions.

Previous attempts toward the adoption of information systems were not successful. Both off-the-shelf applications, web based solutions and custom made systems failed to meet CMN goals.

Between the reasons for abandoning previous approaches to CMN information systems a relevant role is played by the need for an adaptable and continuously evolving system.

Our approach to better address CMN needs was to adopt participatory methodologies in the design of the system.

The workgroup was built using special attention in choosing the participants. CMN senior analysts, accounting managers, sales managers where involved in the team, each with his or her expectations and doubts with respect to the new technology for managing CMN information.

In the first sessions, the workgroup reconstructed in terms of textual use cases the main work processes of CMN. Our staff, after introducing the methodology in the beginning, operated as mentors for the following meetings. The main goal for this phase was to better understand the processes within CMN, using a tool close to the users' experience: the narrative text of use cases.

In the subsequent sessions, IUC were introduced, the focus was then shifted from the representation of the work processes made by the users. With IUC the focus is on the processes as they could be addressed by the information system. The participation of the users in the definition of the IUC underlines the connection with the textual use cases and then with the work processes. The pictorial and interactive nature of the IUC promote a better understanding of the developer intentions.

4. RESULTS

IUC freeze CMN needs, while also highlighting the area of fluctuating requirements. CMN can experience a snapshot of its potential information system. In addition, it is possible to identify which parts of the system are defined with sufficient detail and which one are still evolving. In response to this fact the work group decided to separate future designs in two different phases; separating the design of components manifesting themselves as stable. For these ones, design and implementation activities that may start immediately with low risk, freeing resources for a deeper exploration of unstable components (those affecting the still fluctuating portion of the IUC).

It should be noted that the fluctuation of one portion of the IUC is not due to insufficient requirement gathering, or to incomplete analysis; CMN has understood that this portion of the information system has to be designed and built so that the models of knowledge representation on buying benchmarks may be adapted by the user (the analyst) in use, depending on the specific cases (tailoring). These components will then have to be designed for change rather than stability. This is the most critical issue that in earlier cases has hindered the identification of solutions acceptable to CMN.

As results of the application of IUC the group reached a shared vision of the problem under attack; both parties have become

completely aware of the most relevant issues and of where one should invest more in order to develop a system covering the needs of CMN. IUC had a central role in the mutual learning phase of participatory design. Continuous displacements have been observed with respect to initial ideas (drift). These events, regarded as breakdowns, have been translated inside IUC to be later able to reconstruct the path leading to the final structure of the information system

In their comments, CMN personnel today has a better understanding of the company own needs for information systems and of the related problems. They are also conscious of what should be done in order to face and solve them.

5. CONCLUSIONS

IUC were approached, in the CMN case, after reaching a stable state in the narrative use cases. The rationale for IUC was the participatory exploration of possible areas for building CMN information systems.

The IUC session with users and developers were addressed to:

- WHAT: draw on the screen a pictorial representation of a textual use case, providing a rough idea of the possible human-computer interaction in the automation of work practices. The focus was on the input that the user needs to provide to the system and on the outputs provided by the system. There was no consideration in on the technological feasibility of these features.
- HOW: refine the design of interfaces by working on the logical sequences of steps in the human-computer interaction (starting from the event flow in the use cases). The focus was on the choice of interface elements better representing the work process.
- EVALUATION: using the "walking on the IUC" technique to simulate with the users the interaction with the system, addressing the possible improvements of the system. The interactive nature of IUC helps in this task.
- DOCUMENTATION: remove all possible ambiguities by documenting each graphical element in the interface.

Showing with a video-projector the development process of the IUC on the PC is central point in the methodology used: it allows the users to actively participate into the analysis process. CMN personnel were able to follow the development of IUC, and to correct all misunderstandings.

Several meetings have been filmed with a video-camera, so as to allow off-line observation on the interaction with the users.

The method presents also some difficulties:

- Each meeting saw the emergence on new objectives (a continuous drift): the project focus moved all the time, new and different things were continuously added.
- During the participative construction process of IUC using graphic design tools, the operator's attention can decay. Reviewing filmed sessions can help identifying these cases.
- A too early introduction of IUC may alter users' ideas, imposing on them the analysts' vision.
- IUC allow such a detailed representation of the application that the stakeholders may mistake it already for a prototype! We consciously choose to keep the graphical quality of IUC in an "unpolished" state.

6. ACKNOWLEDGMENTS

We thank Alessia Zumerle and Pierluigi Roberti for their useful contribution to this project.

7. REFERENCES

[1] Blomberg, J. and Kensing, F. (1998). Preface to the Special Issue on Participatory Design. Computer Supported Cooperative Work, 7, 163-165.

[2] Bödker, S. and Grönbaek, K., (1991). Design in Action: From Prototyping by Demonstration to Cooperative Prototyping. In Design at work: cooperative design of computer systems (Greenbaum, J. and Kyng, M., Editors), Lawrence Erlbaum Associates, Hillsdale, NJ.

[3] Chrisman, N. (1999). Trading Zones or Boundary Objects: Understanding Incomplete Translations of Technical Expertise. 4S, San Diego.

[4] Ehn, P. and Kyng, M. (1991). Computer: Mocking-it-up or Hands-on the Future. In Design at work: Cooperative design of computer systems (Greenbaum, J. and Kyng, M., Editors), Lawrence Erlbaum Associates, Hillsdale, NJ.

[5] Ehn, P. and Sjörgen, D. (1991). From System Description to Scripts for Action. In Design at work: Cooperative design of computer systems (Greenbaum, J. and Kyng, M., Editors), Lawrence Erlbaum Associates, Hillsdale, NJ.

[6] Fischer, G. and Reeves, B.N. (1995). Creating Success Models of Cooperative Problem Solving. In Readings in Human-Computer Interaction: Toward the Year 2000 (Baecker et. al. Editors),

[7] Grönbaek, K., Kyng, M., and Mogensen, P. (1993). CSCW Challenges: Cooperative Design in Engineering Projects. Communications of the ACM, 36 (4).

[8] Jacobson, I., Ericsson, M., Jacobson, A., and Overgaard, G. (1992). Object-Oriented Software Engineering: A Use case Driven approach. ACM Press, New York,

[9] Kensing, F. and Munk-Madsen, A. (1993). Participatory Design: Structure in the Toolbox. Communications of the ACM. 36 (4).

[10] Leigh Star, S. and Griesemer, J.R. (1989). Institutional Ecology, 'Translations', and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology 1907-39. Social Studies of Science, 19.

[11] Rumbaugh, J., Jacobson, I. and Booch, G. (1998). The Unified Modeling Language Reference Manual. Addison-Wesley.

[12] Trigg, R.H., Bödker, S., and Grönbäk, K., (1991). Open-Ended Interaction in Cooperative Prototyping. Scandinavian Journal on Information Systems, 3.