Knowledge Management for Requirements Engineering

Daniela E. Herlea, Mildred L. G. Shaw and Brian R. Gaines
University of Calgary
Calgary, Alberta, Canada T2N 1N4
danah@cpsc.ucalgary.ca, mildred@cpsc.ucalgary.ca, gaines@cpsc.ucalgary.ca
http://ksi.cpsc.ucalgary.ca/SERN/, http://ksi.cpsc.ucalgary.ca/KSI/

Abstract

The support of the software engineering life cycle from the initial negotiation of requirements, through analysis, design, coding, testing, integration, trial, use, enhancement, maintenance and replacement, involves the management of diverse knowledge sources and their dependencies. Requirements negotiation, in particular, involves the management of a heterogeneous collection of materials in such a way that a community with widely varying computer skills can create them and understand their relationships. This article describes some experience in applying a groupware knowledge management tool to management of the software engineering life cycle.

1 Introduction

Software requirements negotiation is the process where the customers' needs in a software project are identified. This process is regarded as one of the most important parts of building a software system because during this stage it is decided precisely what will be built. Requirements negotiation is an iterative process where, through reflection and experience, users become familiar with the technology and developers become familiar with the user needs. For example, scenarios, prototypes or mock-ups provide the opportunity for the users to "experience" the new technology and for the developers to "experience" the work practice.

Cooperation between participants in the process is quite difficult, even if they meet in a physical meeting room. The process involves a social network of people with different professional backgrounds and different views over the system that must be built. If the participants in the process are in different organizations or different cities, meetings can be costly, inconvenient and infrequent. This leads to problems of communication, which can greatly impact the quality of the elicited requirements.

There are two major approaches to what has come to be called *requirements engineering*, that of *participatory design* (Schuler and Namioka, 1993) and that of *joint application design* (August, 1991), and the system supports both approaches. The overall problem can be treated as one of system development within the framework of Checkland's (1981) *soft systems methodology*.

Within this methodology the general roles defined for the requirements negotiation task are:

- the *facilitator*, who acts as a chairperson of the meeting, has a critical role in organizing the work of the requirements negotiation team.
- the *users*, who are people involved in using the system, are the "owners" of the problem.
- the *analyst*, who is a representative of the design team, has a key role in the transfer of the requirements from the "problem owners" to the design team.
- the *design team*, who are the system implementors, are responsible to meet the elicited requirements.

Requirements negotiation generates a heterogeneous collection of related data and where knowledge management is a major problem. Our focus on distributed groups required knowledge management tools that explicitly supported structured collaboration involving a wide range of knowledge representations and chose to use the TeamRooms system (Roseman and Greenberg, 1996) which is implemented as an Internet or intranet application using the groupware toolkit GroupKit (Roseman and Greenberg, 1992).

TeamRooms combines shared whiteboards, chat rooms and other customizable groupware applets (integrated mini-applications) in a persistent work environment that lets participants easily and flexibly collaborate and share information with colleagues across a network. The model for TeamRooms comes from real-life team meeting rooms, which provide a shared space for a workgroup. TeamRooms works the same way, except that the room and its objects are now a fully graphical, persistent groupware interface.

3 TeamRooms as a Negotiation Place for Requirements

The first step in customizing TeamRooms to support requirements negotiation was to analyze the roles of the major participants using Checkland's soft systems methodology as shown in the concept map in Figure 1.

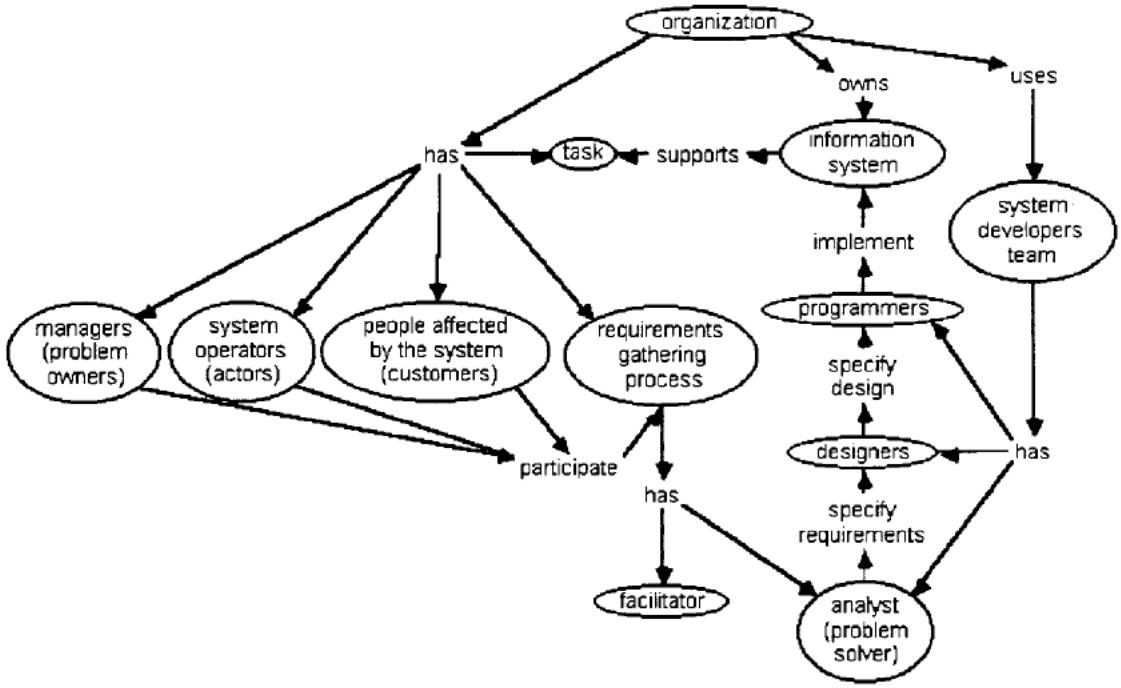

Figure 1 Concept map showing the roles played by the participants within the organization and the requirements negotiation process, according to Checkland's soft systems methodology

Then a set of virtual rooms was defined to support the types of interaction that would be required by participants having these roles:

- **Wish List Room**, used to store the initial list of requirements.
- **Meeting Room**, used as a general working space that gathers the participants in an informal environment.
- **Brainstorming Room**, used to record all "quick ideas" generated throughout the meeting.
- **Work Agenda Room**, used to keep an agenda of the meetings, used mainly by the facilitator.
- **Documentation Room**, used to record the elicited requirements.
- **Scenarios Room**, used to build scenarios of future situations.
- **Future Consideration Room**, used to record the intermediary results of the process.
- **Final Consideration Room**, used to store the final document of requirements, used by the design team in implementing the system.
- **Reports Room**, used to communicate and inform about the design team progress.
- **Worth Proceeding? Room**, used to test the prototypes against the requirements.
- **Read Me Room**, used to leave notes for other participants in the process.
- **Personal Rooms**, used to store users' own artifacts.

The requirements negotiation process is about navigating this electronic workspace. There is no defined order to navigate the rooms. Requirements elicitation is an iterative process, where the users of the system participate in meetings and discussions held in different rooms, at different times; they enter the same room many times in the process life cycle.

4 Collaboration in the Electronic Workspace

Awareness of other users in the work space is very important in simulating the real physical space. Figure 2 shows the list of users currently connected to the group's server, as well as the room each user is currently working in. This facilitates locating the other users and making contact with them. The images are still pictures taken off-line or on-line snapshots taken one per minute.

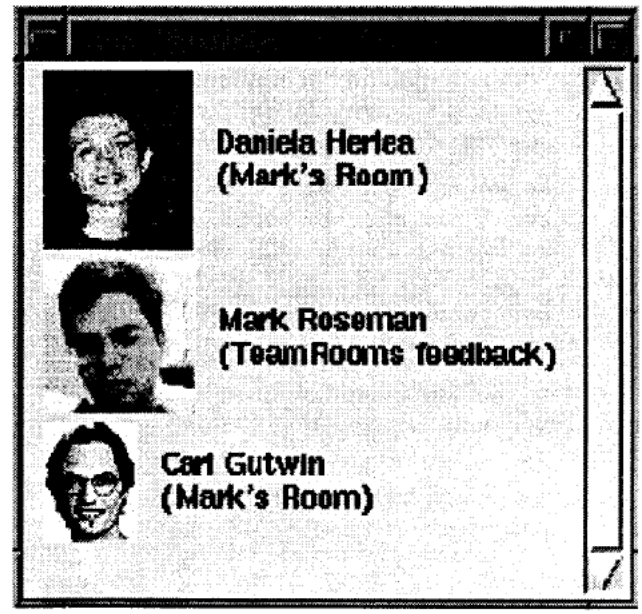

Figure 2 TeamRooms participants awareness

The facilitator, who acts as a chairperson of the meeting, has a critical role in organizing the work of the requirements negotiation team. She enables effective communication, works as a project manager for the elicitation team, can have the knowledge of the room and action of each participant in the working space. She can use the Work Agenda Room shown in Figure 3 to keep an agenda of the meetings, lists of activities and organize the work through an agenda of the meetings and lists of activities, using the Outliner or the Note Organizer applets to create a hierarchical activity structure, or a Concept Map applet to identify and set priorities. Awareness widgets help the facilitator in keeping the team on the right track. Under her direction, the team works to meet specific objectives, as stated in the structured Activities list shown in Figure 3.

As shown at the top left of figure 3, each room has a room radar overview that provides a miniature overview of the entire room when the user changes the location of his or her view onto the workspace to suit the needs of the immediate task. The room can be larger than the participant's display. The radar shows the position of each user's viewport in the room, telepointers indicating the location of their mouse cursor. Each time when a user navigates the room and manipulates applets, the radar tracks his actions.

Figure 4 illustrates the Future Consideration Room, a "storage and future retrieval" area. Exploring the users' environment, the capture team collectively investigates the target users' organizational setting and identifies and describes what they do. They produce documents recording the shared view of the users' environment. They create objects reflecting the users' workspace and save them in a database. Records in this database are files containing information that describes these specific objects. Any user of the system may retrieve these files and discuss the list of the users' needs. Later in the process, members of the team review and evaluate information they have previously generated in order to arrive to common decisions about the value, relationships, and structure of that information. They update this "user document", according to their understanding of the user environment.

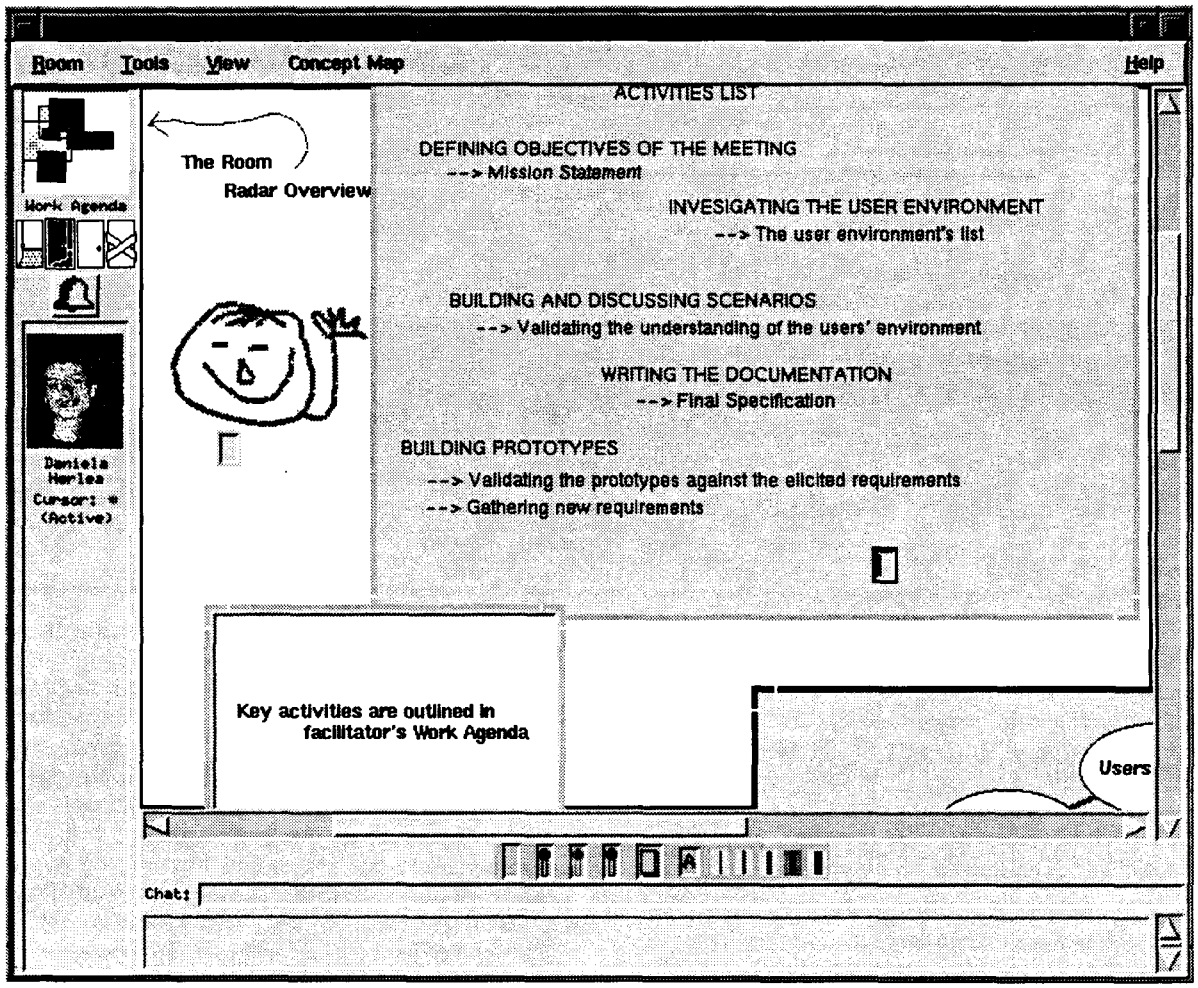

Figure 3 Facilitator's work agenda

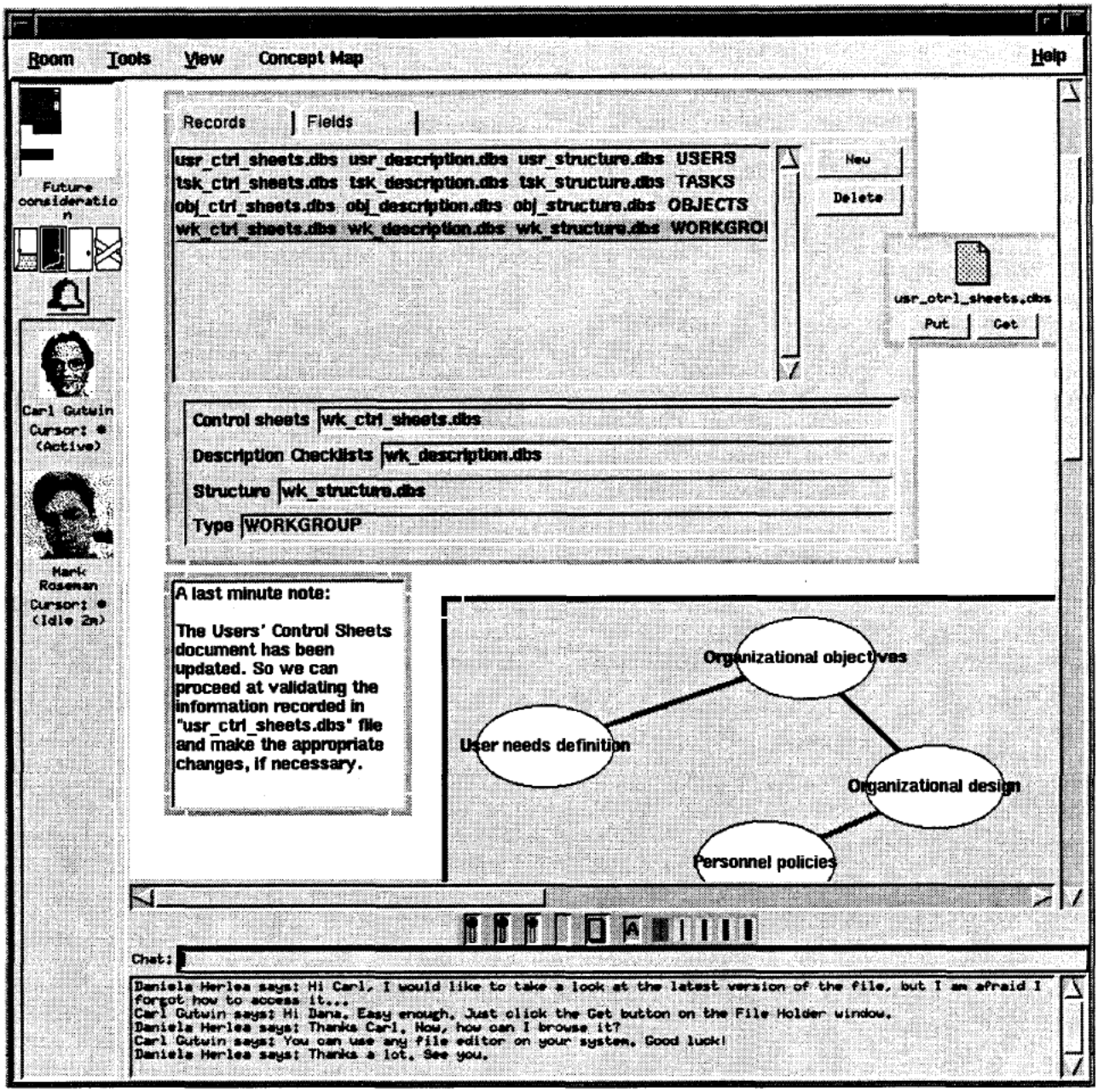

Figure 4 A "storage and retrieval" space for the user environment's objects

Participants can click on the records of the database—objects in the users' environment—and find out what files contain the description of those objects. The File Holder tool enables participants to exchange files in real time. Any file editor would give the participants the opportunity to read and modify the content of the files, after they discussed the changes with the others in the room.

The Documentation Room may be used by the analyst to record the elicited requirements. Documenting the elicitation process may be a time consuming task in usual physical meetings. Using the Documentation Room in the electronic space, the analyst records all information that participants generated, all the analysis conducted, all of the decisions reached. The analyst may decide to use files to store draft plans of the requirements document. This way the documentation files have a higher quality, they are easier and more quickly produced. Validating understanding of the user environment with the system customers and end users, and formulating requirements based on this existing "picture" may be done in the general Meeting Room. Any tool may be used to manipulate the existing information. The Meeting Room is used as a general working space that gathers the participants in an informal environment.

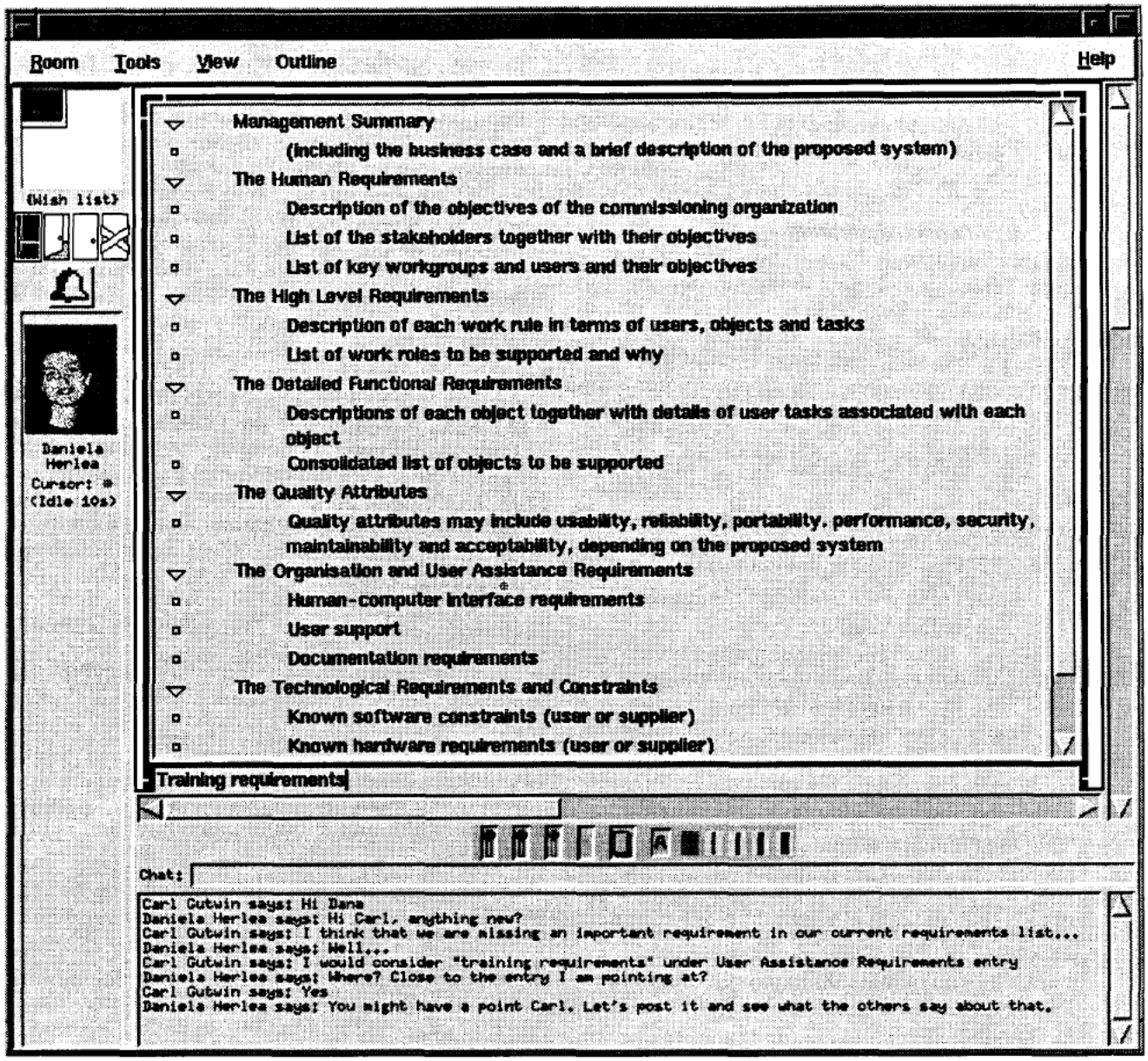

Figure 5 User requirements: list of contents

As a next step, an initial list of requirements may be proposed and stored in a generic "wish list", in the "Wish List" Room. Figure 5 shows a list of contents for a set of issues related to the requirements document. This list of contents is stored using the Outliner tool that helps in analyzing the information. It enables users to build upon a previously generated list and refine it into a hierarchical structure.

5 Conclusions

The support of the software engineering life cycle from the initial negotiation of requirements, through analysis, design, coding, testing, integration, trial, use, enhancement, maintenance and replacement, involves the management of diverse knowledge sources and their dependencies. Requirements negotiation, in particular, involves the management of a heterogeneous collection of materials in such a way that a community with widely varying computer skills can create them and understand their relationships. This article has described some experience in applying a groupware knowledge management tool to management of the software engineering life cycle.

The emphasis in the research described in this article has been on gathering informal requirements in an integrated fashion from a distributed group. In this respect the system may be viewed as a front-end to systems that track dependencies more formally in the software design phase such as CoMo-Kit (Maurer, 1997), and systems that manage testing through globally distributed teams such as GWSE (Global Working in Software Engineering) (Gorton and Motwani, 1996). It would also be simple to support more formal requirements methodologies such as those based on requirements being testable (Eickelmann and Richardson, 1996).

It will also be apparent from this article that TeamRooms itself is a valuable system for knowledge management that is readily customized for specific applications such as requirements negotiation, and that the system described could be used for general requirements negotiation not involving software engineering.

Acknowledgments

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada, and by the sponsors of Dr Shaw's Industrial Research Chair in Software Engineering, Motorola, Computing Devices Canada, Nortel and ACTC. We are grateful to our colleagues, Saul Greenberg and Mark Roseman, for access to the TeamRooms software.

URL's for Materials

TeamRooms:

http://www.cpsc.ucalgary.ca/Redirect/grouplab/

Related articles:

http://ksi.cpsc.ucalgary.ca/articles/

References

August, J.H. (1991). **Joint Application Design: The Group Session Approach to System Design**. EngleWood Cliffs, New Jersey, Yourdon Press.

Checkland, P. (1981). **Systems Thinking, Systems Practice**. Chichester, UK, Wiley.

Eickelmann, N.S. and Richardson, D.J. (1996). An evaluation of software test environment architectures. **Proceedings of the Eighteenth International Conference on Software Engineering**.

Gorton, I. and Motwani, S. (1996). Issues in co-operative software engineering using globally distributed teams. **Information and Software Technology 38** 647-655.

Maurer, F. (1997). CoMo-Kit: Knowledge Based Workflow Management. **Artificial Intelligence in Knowledge Management**. this volume. Menlo Park, AAAI.

Roseman, M. and Greenberg, S. (1992). GroupKit: a groupware toolkit for building real-time conferencing systems. **Proceedings of the Fourth Conference on Computer-Supported Cooperative Work**. pp.43-50. New York, Association for Computing Machinery.

Roseman, M. and Greenberg, S. (1996). TeamRooms: network places for collaboration. **Proceedings of the Sixth Conference on Computer-Supported Cooperative Work**. New York, Association for Computing Machinery.

Schuler, D. and Namioka, A., Ed. (1993). **Participatory Design**. Hillsdale, New Jersey, Erlbaum.