SPOT: Using Collaborative Technologies for Developing Collaborative Technologies

James L. Jacobs, Michael C. Dorneich, Patricia M. Jones, Barbara J. O'Keefe, and Noshir S. Contractor

Department of Mechanical and Industrial Engineering
Department of Speech Communication
University of Illinois at Urbana – Champaign
1206 W. Green St.; Urbana, IL 61801

ABSTRACT

The SPOT project applies emerging collaborative tools and techniques to assist in the development of such tools. The Software Development Group (SDG) of National Center for Supercomputing Applications (NCSA) is developing several frameworks for deploying tools as collaborative applications. SPOT exercises SDG's collaborative frameworks while creating a collaborative project management tool. This tool makes visible the goals, activities, and the status of the various projects within the organization. This includes explicating the various competencies and responsibilities of the people within the organization. To be effective, collaborative technologies need to support fluidly defined groups by making visible the activities and structure of the group in an accessible manner. SPOT attempts to achieve this by bringing together emerging concepts from computer science, cognitive systems engineering, and social sciences. We describe our use of a task–focused approach that makes heavy use of end–user participation from work–study surveys through early paper prototype evaluation to iterative prototyping. SPOT is an interdisciplinary project between Cognitive Systems Engineering and Speech Communications. Our combined perspectives enable our team to combine organizational modeling techniques with cognitive engineering principles to mutually inform the design and enhance the practical utility of the evolving software product.

I. INTRODUCTION

The Software Development Group (SDG) of the National Center for Supercomputing Applications (NCSA) researches and develops advanced collaborative software technologies in a variety of projects. Managing these efforts and fostering collaborations within and between projects in a dynamic, research–oriented environment is complex and difficult task. Modelling the organization and designing an effective tool to increase the communication and collaboration within the organization requires a multi–disciplinary approach. The SPOT project is a collaborative research effort to study the SDG, develop a technical intervention to increase the degree of internal collaboration within their group, and to evaluate the resulting changes in the organization. As SDG is an organization that develops technology to support collaboration, from the start of the project we intended to use collaboration technology from SDG and other sources to support our own research and development activities.

From a cognitive engineering perspective, the major issues in the project are:

- Cognitive System Engineering Tools and Techniques.
- Collaboration with Cognitive Systems Engineering and Organizational Modeling.
- Domain Issues.
- Implementation Tools and Techniques.

From a social science perspective, the major issues in the project are:

- Modelling the communication patterns within the organization.
- Studying the task and work patterns.

Using modeling techniques from both perspectives, we have developed a technical intervention in the form a project management software tool. The functional focus of this tool is to make visible the activities, resources, and goals of the organization in order to facilitate better awareness and communication between organization members.

0-7803-4053-1/97/$10.00 © 1997 IEEE

II. BACKGROUND

SDG has been focused on developing software tools to help people collaborate. They have a strong desire to improve the ways and extent to which they collaborate effectively between project groups. Many of their new projects leverage off existing projects, and so there is a great need for visibility between projects, at all levels. Additionally, they must ensure that projects do not duplicate efforts. Finally, SDG is interested in exploring more human-centered approaches to design. The initial goals of the SPOT project are:

- Identify key coordination and communication requirements for work in the SDG.
- Implement a set of tools designed to improve SDG communication and coordination.
- Deploy and evaluate these tools as they are used by SDG.
- Explicate a design methodology for a collaborative design project such as SPOT.

Thus, we are engaged in activities to learn about the SDG as an organization and to build a collaborative project management tool in Java in order to help SDG functions more effectively, exercise the SDG technologies in interesting and useful ways, and to serve as an example of methods for "human–centered design".

A detailed discussion of the SPOT project is beyond the scope of this paper, but it is important to sketch out our activities. As mentioned earlier, the SPOT group itself is composed of an engineering component and a social sciences component. The hope is that these two disciplines can mutually inform each other as they go about their usual research activities. Social scientists typically focus on modeling organizations via surveys, interviews, and statistical modeling. A typical product of such studies might be a communication network (who talks to whom and how much). On the engineering side, we are interested in the process of building *effective* tools. To build effective tools, we must assess what the user community's requirements and capabilities are, develop a model of the domain in which they work, and design an intervention that supports the user's ability to accomplish tasks. Figure 1 details the collaboration and inter–dependencies of the two perspectives within the design team.

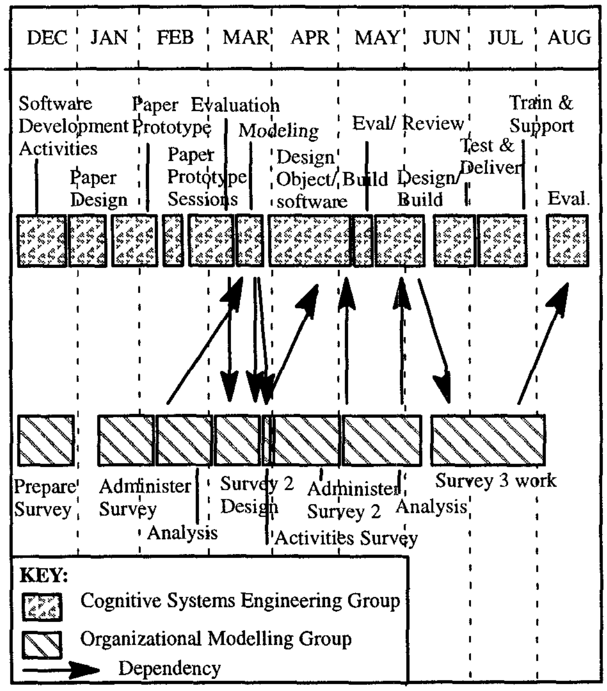

Figure 1. The collaboration and dependencies of the design team.

III. ANALYSIS AND MODELING: TOOLS AND TECHNIQUES

Our design approach: Task centered, user–driven

The main philosophy driving our design approach is the notion that the needs of the task itself are of preeminent importance. Conjointly, the users who are and will be performing the tasks are the greatest source of information and understanding of the tasks themselves. Consequently, in the early stages of the development effort, much of our focus is on constructing an understanding of the requirements of the tasks. As a part of this, we developed models of the organization (i.e.. communication networks, activity models). The early modelling of the domain made it clear that a tool was needed to make visible the inter-connections among work processes, products, and people. Thus, our design focused on building a project management and an organizational information system. The primary components of the tool display the organizational information in several modes, each emphasizing a slightly different perspective: people, projects, and temporal information.

Participation in design: informal prototyping

From early discussions and explorations of the task, we constructed an informal paper prototype of the software tool. An example of one screen of the paper prototype is illustrated in Figure 2. At

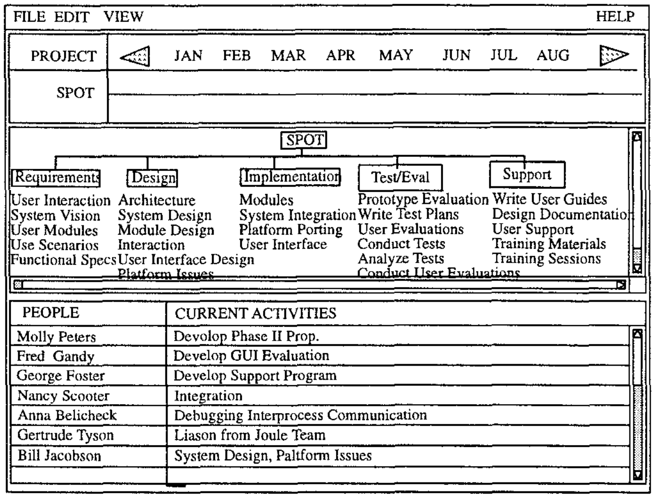

Figure 2. Paper prototype of the Project Window.

the same time, the organizational modeling researchers were building and conducting exploratory surveys of the entire organization and in the process developing a rich base of both formal and informal knowledge about its operation and work practices. Through our internal discussions with them, the system designers were able to tap into that growing knowledge base during our initial concept development.

In this concept development phase, we selected a heterogeneous mix of fifteen users to individual review and provide feedback on our design notions as embodied in a paper prototype. As we folded our early ideas into the paper prototype, we deliberately left the form a bit rough and unpolished. We chose not to present a web–based prototype for the same reason: we wished to convey to the users the extreme malleability and plasticity of the design state. We did not wish them to "hold back" their comments in any sense due to the assumption that the system had already been largely designed. Our approach helped pull them into the major design decisions to a greater extent, and made the user community a part of the design team without letting individual idiosyncrasies distract from the task focus.

System and Task Modeling: Using Rasmussen's Abstraction Hierarchy

As result of the paper prototype sessions, we had over 20 hours of video, dozens of pages of comments, notes, and questionnaire responses. We tabulated, examined, summarized, and thoroughly immersed ourselves in this data. The next step was to build a map of the domain and the key elements within it. For this purpose, we chose to use an "abstraction hierarchy" model.

Developed by Jens Rasmussen[1], an Abstraction Hierarchy is a multi–level representation of a system from different scales of analysis (perspectives). One critical feature of an AH is that the different levels are linked by causal (i.e. means–end) relationships. Properties at one level lead to a satisfaction of functions at the level immediately above. Additionally, properties at a level are the reason for the functions in the level immediately below.

A typical template of the levels of abstraction of a domain would be the following:

- **Purposes and Constraints.** (categories in terms referring to properties of environment, properties necessary for relations between performance of system and the reason for its design)
- **Abstract Functions and Priority measures.** (categories in abstract terms, referring to neither system nor environment, representation of concepts necessary for setting priorities and allocating resources)
- **General functions.** (categories comprises representations and activities having general and familiar input–output terms for particular domain, denotation of activity or function is independent of underlying processes or implementation)

- **Physical Processes and Activities.** (underlying physical processes and equipment)
- **Physical Form and Configuration.** (categories in terms of objects, their appearance and location, inventory of material resources).

The process of building this domain map helped to frame our entire project development, and help make our various contributions fit into a coherent (and interacting) whole. Not only did it develop a list of functions we hoped the Project Management Tool could be built to support, but it also helped to generate a list of issues and questions for the next organizational survey. This was another key place were the two perspectives and two halves of the project team were able to collaborate.

Iterative Prototype development

After analyzing the results of the paper–prototyping sessions, we began to code. Adhering strongly to an iterative, staged approach, we developed an aggressive, tightly focused prototyping schedule. The designed functionality was broken into related functional areas, and we assigned progressively more complex functions to each of a series of three prototypes. We allocated about two to three weeks for the implementation of each prototype, with each cycle building on the earlier version. Importantly, the cyclic pattern allowed opportunities for evaluation and revision of the design.

The software was written in the Java language (JDK 1.1) making full use of its object–oriented approach and platform–independence. We reused object code for various elements of the system from seven different external sources, adapting it as needed for our use. By rough estimate, this saved about eight months of team development effort. Continuing this model, we developed the SPOT system as a collection of reusable class libraries with separate application code built on top.

Making Use of Multi–Disciplinary Research Approaches.

Due to their extensive interactions with the SDG people, the organizational modeling researchers developed a rich understanding of the organizational climate, issues, and work patterns. This knowledge was a great resource to the software development effort of the cognitive system engineers. Another result of their work that contributed directly to the engineered system was a structured list of work activities that was used in the software to build models of skills and activities. Throughout the project, we found that the multiple perspectives in the team strengthened the technological intervention. Along the way, it became clear that the particular research focus of each portion of the team was aided by the work of the other. Together, the combined resources of the team help create richer models of the domain, the organization, and the users.

IV. OPTS: DESCRIPTION AND STATUS

OPTS, the software system developed in the SPOT project, is a software management and organizational information tool built upon the SPOT software libraries. The goal of the OPTS System is to make visible the activities, resources, and goals of the organization in order to facilitate better awareness and communication between organization members. The SPOT libraries model the components of a research organization as a series of objects, e.g. projects, people, events, skills, activities, roles, deliverables, etc. OPTS builds upon these descriptions an environment where these objects can be visualized, manipulated, and edited. An important element of the environment is the ability to define and visualize the dependencies and commitments that bind these objects together in both inter- and intra-project ways. Finally, by incorporating the OPTS system into JOULE (developed by SDG), an asynchronous virtual collaborative work environment[2], OPTS will be able to leverage off several features to create a persistent, collaborative tool. Features of JOULE that can be incorporated into OPTS will include persistent object database capabilities (such that information is served from a central database and thus the information in the tool is always current) and a notification system. The notification system allows users to register a level of interest in an item, and then the user will automatically be notified if the item of interest has been changed or has come due.

The OPTS software contains within its database a current description of the organization. The software has four principle ways of viewing that information, where each mode emphasizes a different perspective on the information. The Project Overview Window provides the user with an overview of all the projects in the organization, and the events associated with those activities. Figure 3 illustrates the Project Window that details a specific project, with an emphasis on displaying the structure of the activities, and the people associated with those activities. Figure 4

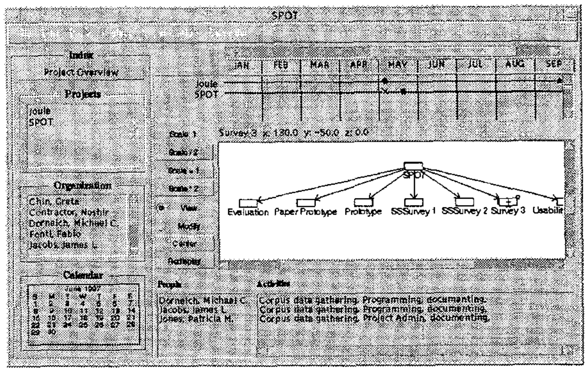

Figure 3. The Project Window.

illustrates the Organization Window, which shows the people in the organization and the relationship they have to each other and the projects they work on. Finally, Figure 5 illustrates the

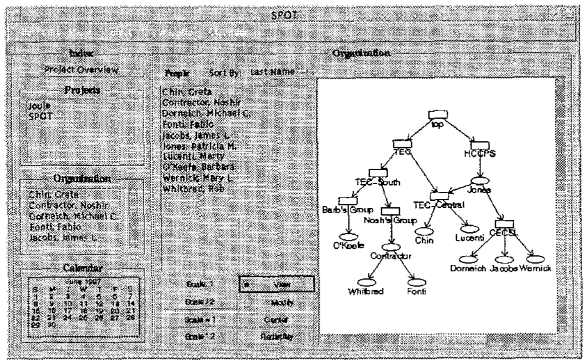

Figure 4. The Organization Window.

Calendar Window, which allows the user to view the events (meetings, deadlines, decision points, etc.) within the organization, with the ability to filter the view such that only the events of interest are displayed. At all times, the Index Window at the left of the main screen displays some relevant information, as well as allowing the user to quickly toggle between main window screens.

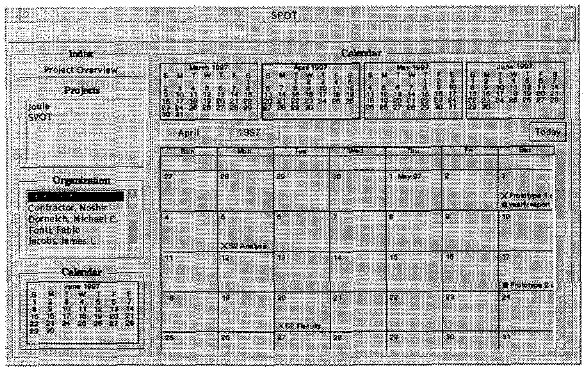

Figure 5. The Calendar Window.

V. CURRENT STATUS AND FUTURE DIRECTIONS

Field testing of implementation

Currently, the third internal prototype of the software tool is being released as a demonstration prototype and tested within the SDG by key users. A small number of SDG managers and engineers are evaluating the system for its utility and efficiency. This is helping us to find the inevitable problems, bugs, and minor software issues; in short—aiding in the verification of our code.

Refining and extending the prototype

The issue of *validation* refers to ensuring that the system does what it needs to do. For this we sought intensive participation from the user community. At the time of this writing, we are planning a phased insertion of the tool, gradually introducing it to users with hands–on training and focused user tutorials. A key part of this interaction, and perhaps the most important, is drawing out their comments and feedback on the design and implementation. This is the best opportunity for us to ascertain where our efforts have gone awry and make the needed course corrections with the technology development.

Re–examining the organization and observing changes

At about the same time, the organizational modeling portion of the team will be conducting another comprehensive survey of the organization. By using the initial survey suite as a baseline, it will be possible to directly observe changes in work patterns. A third comprehensive survey planned for six to eight months later will enable us to evaluate the impact of our technical intervention (the software tool) on the organization.

Workflow models

One of the next major domains we intend study and support is workflow modeling. We will be applying the same development and task domain modeling approach to issues of how work to partitioned and structured. Using the SPOT software and significant extensions to it, we hope to provide representations of the interactions between the various members of a work group, outside agencies, as well as an environment to envision and manipulate such representations.

Develop and document methodology

The SPOT project has drawn heavily from both social science research paradigms and cognitive system engineering techniques. The approach we have developed and used for this project shows promise for broad application. A significant upcoming task for us is the explication and codification of that process.

Build Abstraction Hierarchy tool

The abstraction hierarchy model played a significant role in our development approach. Unfortunately, we've not been able to find good tools tailored to the creation and editing of these models. We now intend to build a tool suite to provide this functionality.

Develop connections to support simultaneous collaboration environments

The design of the SPOT tool will support asynchronous collaboration. We wish also to tie SPOT into a synchronous collaboration environment. This would allow SPOT to act as a resource for group decision making in an immediate fashion.

VI. ACKNOWLEDGEMENTS

The authors would like to acknowledge Fabio Fonti, Dawn Pfeiffer, Heidi Saltenberger, Alex Sielicki, Mary Wernick, and Rob Whitbred as fellow contributing members of the SPOT project team. Thanks also to Carolyn Fiebig for lending her expertise and assistance in building the Abstraction hierarchy. Thanks again to the members of the Software Development Group of NCSA for their time, assistance, and participation during the course of the project. This project was supported in part by NCSA Grant no. 1–2–23961.

VII. REFERENCES

[1] J. Rasmussen, A. M. Pejtersen, and L. P. Goodstein, Cognitive Systems Engineering, New York: John Wiley & Sons, 1994.

[2] JOULE System TOC Webpage, http://skydive.ncsa.uiuc.edu:8003/joule_home, 13 Dec. 96.