Facilitating Communication in Software Development

Michael E. Atwood, Bart Burns, Dieter Gairing, Andreas Girgensohn, Alison Lee, Thea Turner, Sabina Alteras-Webb, and Beatrix Zimmermann

NYNEX Science and Technology
500 Westchester Avenue
White Plains, NY 10604, USA
E-mail: {atwood, bart, gairing, andreasg, alee, thea, webb, bz}@nynexst.com

ABSTRACT
Effective communication is critical to the success of a software development project. It factors into the productivity of individuals and organizations, and has particular impact when change occurs. Yet communication is generally left unsupported by the software development process and by the communication infrastructure. We address this issue in the context of two software development projects at NYNEX through a conceptual framework called *Design Intent*. There are three innovations in our approach. Design Intent encourages stakeholders to engage in active listening, enables stakeholders to collaboratively construct a consistent understanding of the development effort, and provides a communication infrastructure for stakeholders to share ideas and participate in discussions.

INTRODUCTION
Effective communication among the stakeholders of a software development project is crucial to its success. The importance of this communication has been well documented by Curtis, Krasner, and Iscoe [8], who noted frequent, recurring problems related to the lack of adequate communication among those involved in the development effort. In order to improve the communication among members of a software development team, an effective process and the infrastructure to support it must be provided.

As designers, we have a new role of "designing experiences" or ways for people within our corporation to appreciate new ideas. This role is "*designing the boundary objects that facilitate communication and the interpretative moves [leaping] of overlapping communities of practice*" [18]. We need to build not only "prototypes of need or use" and "prototype systems" [2] — but also the infrastructures that support relationships, work practices, and social intercourse in communities of learners and knowledge workers.

We feel that three main activities are essential for producing good software systems:

1. *Active listening* and *interpretive leaping*: understanding the problem and how to solve it in a significant way — offering a model of transcendence.
2. *Designing boundary objects* that help people *experience* the power and possibilities of new ideas.
3. *Facilitating the communication* of ideas and innovations by building the infrastructures.

We refer to them as the three dimensions of *Design Intent*. We will discuss our experiences along these dimensions using two projects at NYNEX — SPARX and DADAS. SPARX (Spatial Analysis Resource for NYNEX) is a system meant to support network design engineers. We will use work on this project to discuss problem understanding and transcendence. DADAS (Direct Access to Directory Assistance Service) is a system to enable third party access to NYNEX's Directory Assistance Listing Services Database (LSDB). Here we will present the communication infrastructure that we built for the project along with issues and experiences with building and deploying the communication infrastructure. In each case, boundary objects were produced. In the course of working on these projects, it became clear that if these three dimensions of Design Intent were not in balance, then the overall success was jeopardized.

TWO PROJECTS EXPLORING DESIGN INTENT
Design Intent took root within two projects: SPARX and DADAS. Our research approach is to embed the development of Design Intent system within real development projects. This approach grew out of the recognition that only by being part of a development process could the Design Intent approach react, support and evolve along with the very development process that it was to improve.

SPARX Project
NYNEX has about 1200 wire centers, each with 300-600 paper records of outside plant schematics (known as plats). SPARX was an effort to develop a digital database mapping *outside plant* items to their geographical reference points. *Outside plant* is basically the piece of the phone network between a Central Office (CO) and the customer locations. When some change needs to occur in outside plant network (either through maintenance, network planning for growth, or specific customer requests), the design engineer is responsible for taking these facility requirements and designing an unambiguous plan that will work in the current state of the real world ("the field"). This plan is then given to the construction personnel, who implement it. The SPARX tool would be used by the outside plant design engineers who are designing broadband networks. Similar domain-oriented systems could be built for planners, management, forecasters, and so forth.

DADAS Project

Directory Assistance Direct Access Service (DADAS) provides inter-exchange carriers, certified local exchange carriers, and other telephone service providers with access to NYNEX's Directory Assistance Listing Services Database (LSDB). In addition to facilitating access by outside carriers, the service must also safeguard and ensure that requests neither degrade system performance nor permit proprietary information to be accessed from the outside. The DADAS Bridge is the software and hardware that provide outside carriers with managed access to the NYNEX LSDB. The system used to develop the Design Intent system was the web browser from Netscape Communications.

DESIGN INTENT

Design Intent (see Figure 1) provides an information repository whose contents are derived from project information and from communications among individuals and groups [3, 5]. Its contents are structured as a hypermedia document. It contains sharable knowledge of persistent value to the development process and the stakeholder community (e.g., folklore). It is an artifact embedded within a development process which evolves in response to additions and updates during the process [20]. Its content seeds communication and forms a medium of communication for integrating stakeholders and information. This communication supports and focuses stakeholders on problem understanding, mutual education, and collaboration so that the history and rationale of the system as well as the system itself are ultimately constructed as end products.

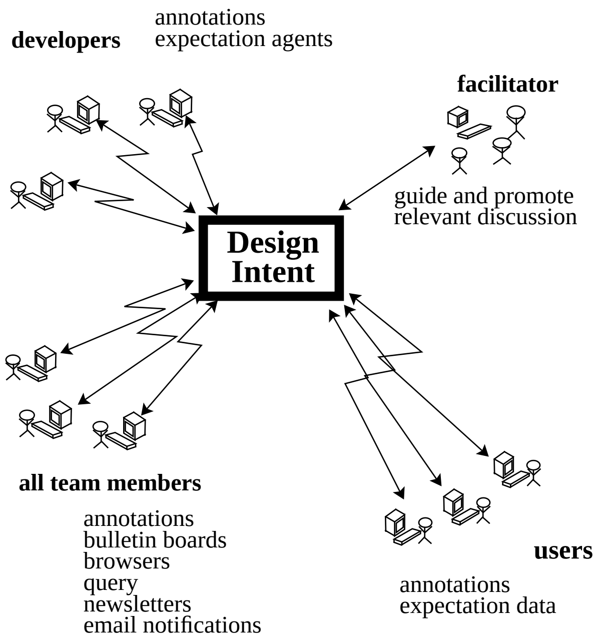


Figure 1: Design Intent.

Design Intent provides the infrastructure to support a development process that must adapt to change. In order to do this, Design Intent must assist stakeholders in:

1. active listening and interpretive leaping,
2. designing boundary objects, and
3. facilitating communication

Throughout this paper, we interchangeably refer to this infrastructure as the Design Intent system.

ACTIVE LISTENING AND INTERPRETATIVE LEAPING

There is a lot of aggravation, time, and expense involved with applying technology. It has to be worth the trouble. We do not want our customers to have to wonder if they are happy or not with what is delivered. So, our basic intention of design is to build something *irresistibly* useful. This means achieving significant transcendence of the current practices of the problem domain. In order to do this, we have to not only understand the problem we are trying to solve but we also have to understand how current technology can be applied to achieve this transcendence. In other words, design needs to be a dialogue between the problem domain and available technology ("a dialectic between tradition and transcendence" [11]). This dialogue is the *active listening* and *interpretive leaping* aspect of *Design Intent*. It is the basis for a shared problem understanding across problem stakeholders which is the source of the system requirements. This dialogue is not just a one-shot deal, but needs to be continuous.

The problems we are trying to tackle are complicated, not well defined or understood, and frequently changing. The technology we apply to solve these problems is on a nonstop exponential growth path. We have to deal simultaneously with continuously shifting problem *and* solution spaces. Achieving closure or coming up with an ultimate, complete solution is not possible. Guessing about requirements and making delivery commitments well in advance of knowing what really needs to be done is a recipe for design disaster. We are constantly faced with *information overload, impossibility of coverage, and unavoidable obsolescence* [10].

The systems we build have to be useful, usable, and evolvable. While *usable* and *evolvable* may be enabled by technology, *usefulness* has to come from understanding the problem. We need to take an active role in "coalescing points-of-view around the nexus of the problem" [18]. In order for the mutual education necessary for shared problem understanding to take place, we need an environment of genuine collaboration — where everyone involved benefits [10]. This requires mutual trust. How do we build this mutual trust? "Tell the truth" [1].

In SPARX, we found expert practitioners in design, planning, and management of outside plant engineering. They worked in different NYNEX regions — Brooklyn, Manhattan, Rhode Island and Massachusetts — and in urban and suburban areas. By actively engaging these practitioners we developed an understanding of their current situation beyond what is articulated in the "Bell System Practices" for outside plant engineering (see Figure 2). We found that outside plant engineers are bombarded with bureaucracy while given very little support for design. About 50% of their efforts are involved in managing the technical bureaucracy already in place. Technology was actually taking them away from their job of design.

We definitely do not want to emulate this way of doing things. Given their current dilemma and available technology we offered a model of transcendence. Figure 3 illustrates how SPARX should work toward supporting the design engineer along the critical path of design, with design as the locus of interaction and technology doing the grunt work.

Our SPARX involvement highlighted the fact that it is not enough to understand the problem and then achieve theoretical transcendence. We have to be able to implement our model of transcen-

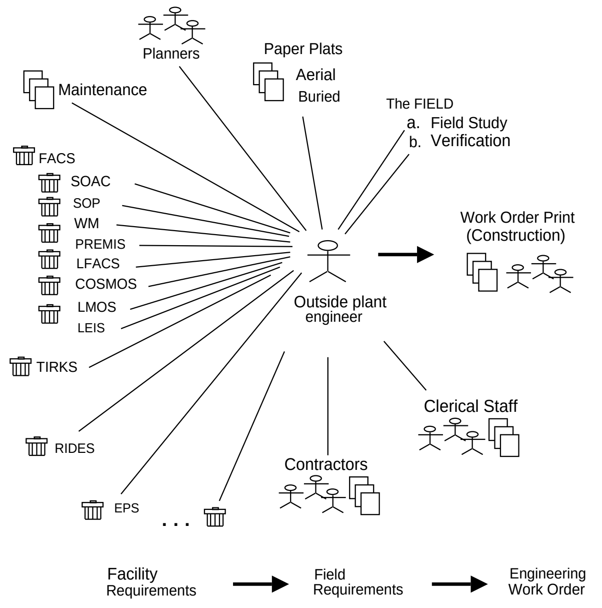


Figure 2: Outside Plant Engineer's job.

dence in a way that is usable and evolvable. In order to do this, we have to be able to deal with the complexity of problem knowledge as well as the complexity inherent in the computational technology.

DESIGNING BOUNDARY OBJECTS

Many types of knowledge are needed to build complex systems successfully. They are distributed among many different experts and organizational processes, across different parts of the organization, and among such artifacts as documents, old systems, or regulations governing business operations. In order for projects to succeed, this knowledge must be captured so that they can be made available to project members when they need it.

We began exploring this approach in the SPARX project and extended it further in DADAS. We will use a concrete scenario of information exchange from this project based on an information repository that we helped to create. It was crucial that the SPARX developers communicated with and understood the special requirements of an outside plant engineer. In an effort to facilitate this communication and to build the necessary collaborative setting between the engineers and the SPARX software developers, we worked with several outside plant engineers from Brooklyn and Rhode Island on establishing a Design Intent repository that would allow them to communicate their requirements to the software developers.

With the engineers we created workflow documents that they felt were descriptive of what their job actually entails when the design is done with paper and pencil. Then, we identified areas in the workflow that the engineers envisioned SPARX would support. After each of these meetings, the information was composed into a Framemaker hypermedia document. The document made use of the actual information collected in the meetings. For example, it contains a video clip of an outside plant tour given by one of the plant engineers and hand-drawn sketches developed in these meetings. Figure 4 illustrates the workflow information included for SPARX.

The purpose of this document was to present this information about the outside plant engineer's world in a form that would be easily understood by the software developers in SPARX. Developers were able to view comments made by the engineers as well as to respond to them. Thus, our efforts were an attempt to facilitate communication between the parties by communicating what the current work system was like. This information was intended to provoke discussions between users and developers about how the new system will fit into and interact within the work system in

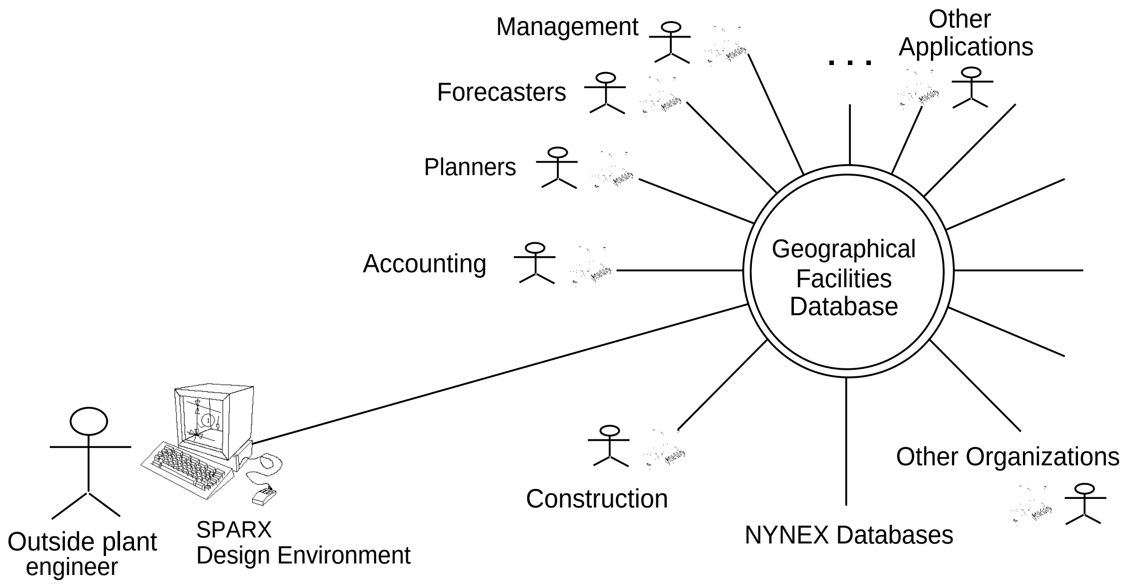


Figure 3: Overview of the SPARX system.

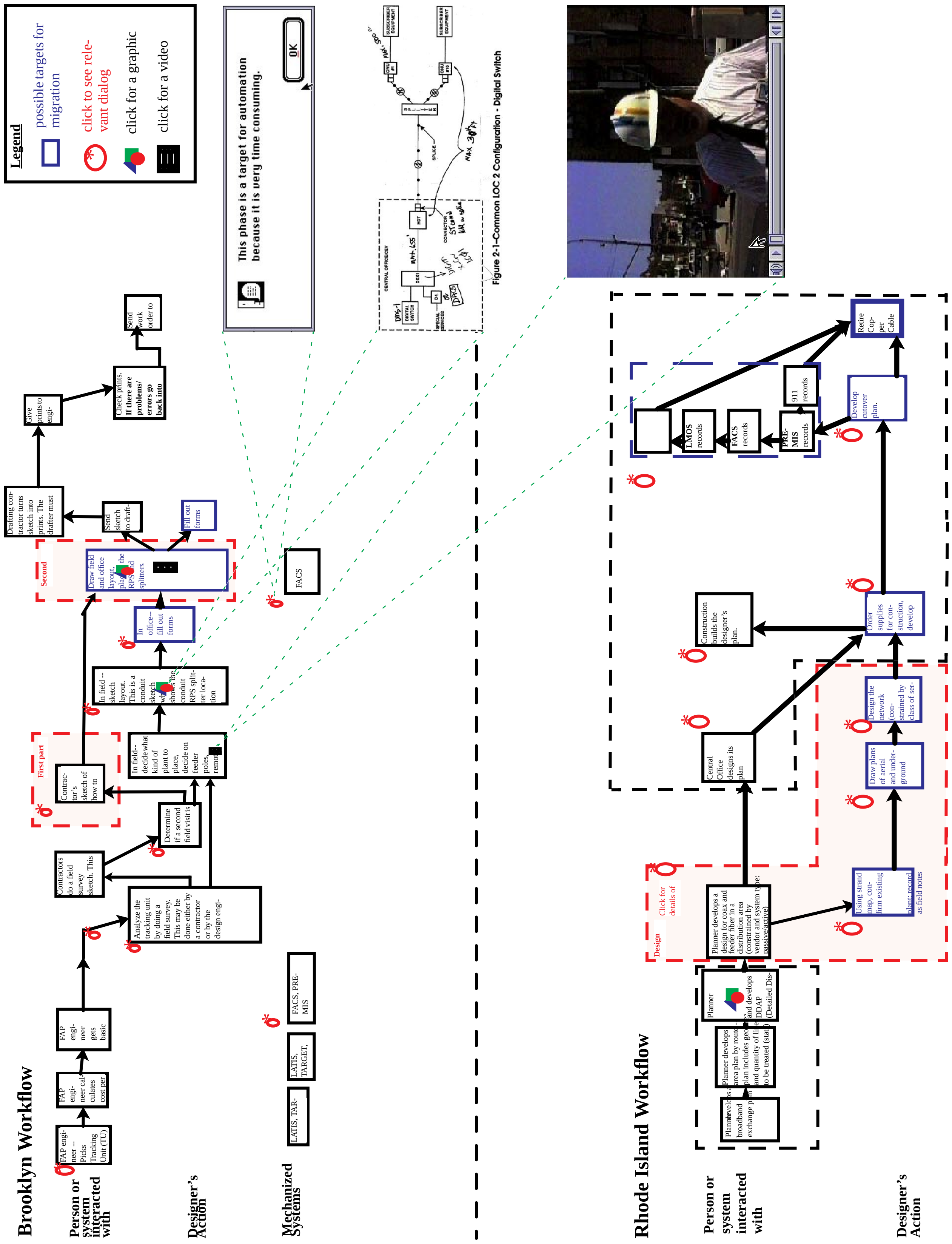

Brooklyn Workflow
Person or system interacted with
Designer's Action
Mechanized Systems
Legend
possible targets for migration
click to see relevant dialog
click for a graphic
click for a video
FAP engineer – Picks Tracking Unit (TU)
FAP engineer calculates cost per
FAP engineer gets basic
Analyze the tracking unit by doing a field survey. This may be done either by a contractor or by the design engineer
Contractors do a field survey sketch. This
Determine if a second field visit is
Contractor's sketch of how to
First part
In field--decide what kind of plant to place, decide on feeder poles, remove
In field -- sketch layout. This is a conduit sketch which shows the conduit RPS splitter location
In office--fill out forms
Second
Draw field and office layout, the placement of RPS and splitters
Fill out forms
Send sketch to drafting
Drafting contractor turns sketch into prints. The drafter must
Give prints to engineer
Check prints. If there are problems/errors go back into
Send work order to
LATIS, TARGET
LATIS, TARGET,
FACS, PREMIS
FACS
This phase is a target for automation because it is very time consuming.
OK
Figure 2-1–Common LOC 2 Configuration - Digital Switch
Rhode Island Workflow
Person or system interacted with
Designer's Action
Planner develops broadband exchange plan
Planner develops area plan by route – plan includes geography and quantity of lines to be treated (status)
Planner and develops DDAP (Detailed Distribution Area Plan)
Planner develops a design for coax and feeder fiber in a distribution area (constrained by vendor and system type: passive/active)
Design
Click for details of
Using strand map, confirm existing plant; record as field notes
Draw plans of aerial and underground
Design the network (constrained by class of service)
Order supplies for construction, develop
Central Office designs its plan
Construction builds the designer's plan.
Develop cutover plan.
PREMIS records
FACS records
LMOS records
911 records
Retire Copper Cable

Figure 4: Outside plant design engineer workflows for Brooklyn and Rhode Island. Note that the workflows are not identical.

which it is placed. As shown by the workflows, even the current premechanized workflow is very different in each office.

There are two innovative elements in our approach. First, we encouraged team members to determine the nature of the content of what they needed to communicate (i.e., actual communicative artifacts). This information is broad in scope. Second, there was neither a prescribed representation for the information nor a prescribed structure to the information being communicated. A purpose of capturing and presenting the information was to facilitate communication as part of the primary goal of developing an understanding of the problem among all project members. As such, it was important that the communicators determine the form and content of the communications. Consequently, our approach differs from previous design rationale approaches [6, 7, 17] by not prescribing the representation and structure to be used nor the nature of the information to be communicated. In both SPARX and DADAS, document authors are able to use the presentation media that is most effective for communicating their information.

FACILITATING COMMUNICATION

Our SPARX experience led us to realize that although an effective process and the design of boundary objects are necessary to system development efforts, they alone are not sufficient without an infrastructure to support collaboration and negotiation. This communication infrastructure represents the third dimension of *Design Intent*. It integrates the community of project members, the tools, and the information in an effort to foster an environment in which problem understanding, mutual education and collaboration and negotiation take place.

Our design process is common with approaches like design rationale and participatory design — developers talk to those in the know, listen to what they have to say, work on understanding the problems and offer real solutions. The innovation in our approach is to provide an infrastructure to support this process by creating mechanisms that facilitate and record this communication. The following four sets of collaboration and negotiation mechanisms were developed for the communication infrastructure in the DADAS project:

1. Awareness and cohesion mechanisms to foster socialization of stakeholder community.
2. Mechanisms that focus team members on artifacts of communication that is the basis for the construction and evolution of a mutual understanding.
3. Mechanisms that enable other team members to examine the communication artifacts.
4. Mechanisms to disseminate new additions and changes as they become available.

Each of these mechanisms are presented in turn in the following subsections. Then, we present our experiences in deploying these mechanism in the DADAS project.

Socialization of Stakeholder Community

More often than not, project members and groups differ culturally, are separated geographically, have different concerns, and may not have known each other previously. Facilitating communication involves not only lowering physical barriers related to temporal, spatial, and organizational distances, it involves lowering cultural and social barriers to communication.

The cultural and social dynamics of a project group influence how knowledge is shared and integrated by individuals and groups in the formulation of a shared vision [4, 8]. These dynamics affect the ability of a project team to translate individual talent into group talent that is critical to the success of the project. This group talent includes not only the team's ability to design and implement programs or to integrate crucial project information so that all parties share a consistent understanding of the development; it includes their ability to bring critical coordination, collaboration and negotiation skills to bear during the communication process. Such skills require cognitive and social abilities as well as the ability to use the interpersonal relationships that is developed in the course of the project (i.e., socialization). Consequently, beyond bringing team members together or providing them with the ability to communicate, the infrastructure must support the socialization of the group.

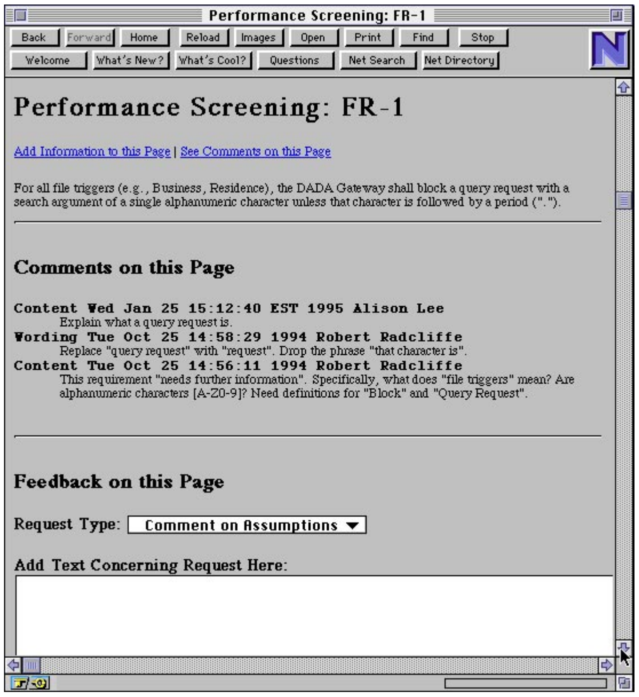
As part of this support, the infrastructure needs to assist individuals and groups to overcome physical, cultural, and social barriers that affect communication. The infrastructure lowers physical barriers by offering alternative means of communication and by making all artifacts produced in communication accessible to everyone (discussed in next three subsections). It attempts to lower cultural and social barriers by offering mechanisms that help team members become aware of others in their community and for the community to build a sense of cohesion [9, 13, 19].

In DADAS, we created documents that enabled individuals, groups, and organizations in the development project to identify themselves and to share information about themselves. Some examples contained in documents on each team member include contact information, photo, project role, and message of the day. This was coupled with mechanisms (e.g., electronic mail, bulletin boards, availability) that enabled individuals and groups to create their own informal opportunities to communicate. Furthermore, individuals or groups who create documents for the Design Intent system (authors) are identified in the document by name and by a photo. This enables others to associate name and face to a document. We are currently implementing more sophisticated awareness mechanisms indicating availability of individuals and groups (like UNIX finger, or Xerox PARC-like Portholes images [9]).

Construction and Evolution of Understanding

As part of our SPARX efforts, we worked on having discussions going between users and developers so that they could collaboratively construct and evolve a mutual understanding of the project's problem. In an effort to enable DADAS project members to actively initiate and participate in these discussions themselves, we developed an *annotation* mechanism. This mechanism allows users to visit particular documents seeded with topics (e.g., functional requirements) and to add their updates, critiques, clarifications, or comments. Such annotations are embedded in the vicinity of the applicable document from the most recent to the least recent (see Figure 5). The resulting discussion thread, created through the series of annotations, resembles, in many ways, the discussions that occur on a *bulletin board*, *mailing list* or *newsgroup* service around a topic. The information and discussions like the ones on functional requirements are intended to evolve to become the basis of, for example, a specifications document, design document, or test plans.

There are two mechanisms, built into system prototypes or installed systems, that are intended to fold information and comments obtained from such systems into the Design Intent system: *expectation agents* and *prototype annotations*. *Expectation agents* [15], having noticed that users deviate from their expectations of how a system is to be used, can communicate directly with the users to collect further clarifications. *Prototype annotations* [12], like



Performance Screening: FR-1

Add Information to this Page | See Comments on this Page

For all file triggers (e.g., Business, Residence), the DADA Gateway shall block a query request with a search argument of a single alphanumeric character unless that character is followed by a period (".").

Comments on this Page

Content Wed Jan 25 15:12:40 EST 1995 Alison Lee
Explain what a query request is.
Wording Tue Oct 25 14:58:29 1994 Robert Radcliffe
Replace "query request" with "request". Drop the phrase "that character is".
Content Tue Oct 25 14:56:11 1994 Robert Radcliffe
This requirement "needs further information". Specifically, what does "file triggers" mean? Are alphanumeric characters [A-Z0-9]? Need definitions for "Block" and "Query Request".

Feedback on this Page

Request Type: Comment on Assumptions

Add Text Concerning Request Here:

Figure 5: An example of annotations.

the expectation agents, also retrieves feedback but differs in that people, rather than the system, initiate the dialog. Having identified some changing need while using the prototype or releasing new software, users or developers communicate this information via prototype annotations. The feedback, in both mechanisms, can be sent as reports to interested team members and/or related directly back into the design discussions occurring in the Design Intent system.

Examination of Information

Team members can examine the communication artifacts in one of two ways: *browsing* or *querying*. The basic browsing mechanism is provided by the hypermedia platform. Links in the document enable users to move freely from document to document. A link may bring up a media player (e.g., video, animation, movie) or a viewer (e.g., document). In addition to this basic mechanism, we provided additional browsing capabilities for some documents. For instance, the DADAS project timeline document includes a number of display widgets that allow the user to change a number of parameters that affect the display of the document's content and its detail (Figure 6).

The query capability is currently under construction and it allows users to retrieve relevant documents matching certain textual information. Furthermore, we are extending an application development environment (called Dynamic Forms [16]) to allow users to query the information repository within the application prototype. These queries may include, as parameters, the current system context. This query mechanism, embedded in the application prototype and development system, provide an external query function to the information repository in much the same way that expectation agents and prototype annotations provide an external annotation mechanism to this repository. Consequently, the communication infrastructure is integrated with the application development system so that information can be added to and re-

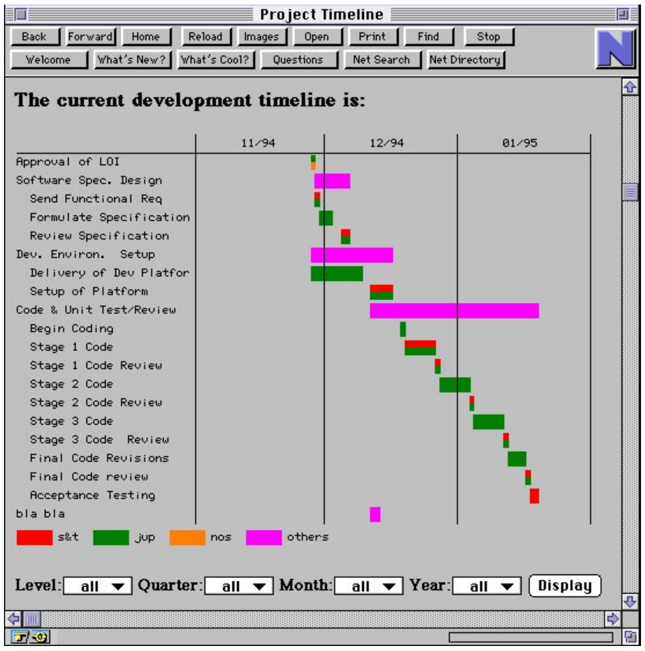


Figure 6: Project timeline page. Note the customizability.

trieved from the information repository from the application development system.

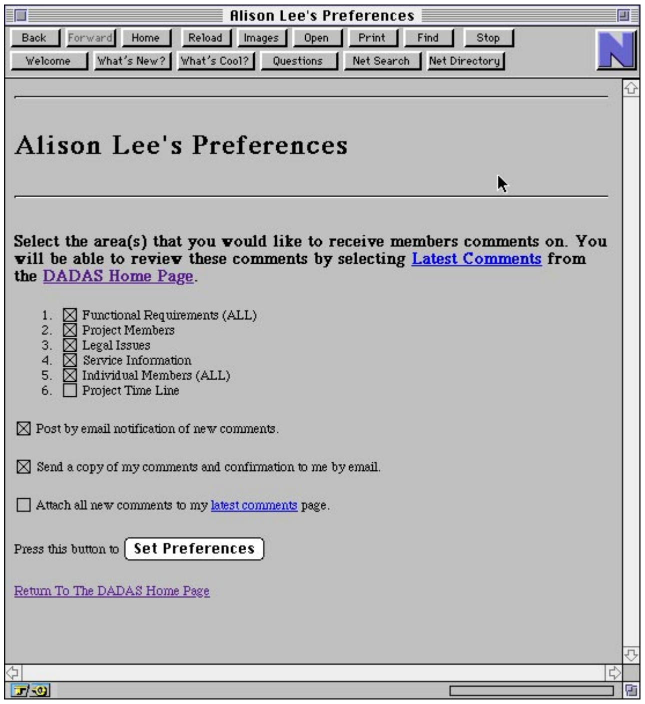
Dissemination of New Additions and Changes

Authors and readers of communication artifacts need to keep abreast of new additions and changes. Each individual has a different preference for how they are notified of comments that they added or comments other team members added. Furthermore, they need to indicate whether they are interested in all or a select set of discussions. In order to support these user needs, we developed a "Preferences" document in the DADAS project. This document is a form that allows the user to indicate their preferences and to change these preferences at anytime. Figure 7 shows a portion of the "Preferences" document.

In DADAS, users may be notified about additions and changes in several ways. First, they can examine their *personalized news page* (entitled "Latest Comment Selection for …" — see Figure 8) if they choose this notification option. It contains all comments made by anyone (including themselves) on those documents that they have expressed an interest. Within this document page, individuals may keep or remove each article item as they so choose. Links are provided from within each article to the document where the comments are located.

Second, they may have indicated a preference to get *electronic mail notifications* when new comments are made. This serves, largely, to remind them to check out the information in the Design Intent system. When the users use the Design Intent system, they can check out the "Latest Comment Selection for …" to see these new comments.

Finally, there is a *newsletter* page entitled "Latest Developments." It has the same representation as the personalized news page but its content is a compilation of all additions and changes made by any team member on any document within a recent period of time. This page includes comments on all documents, including those



Alison Lee's Preferences

Select the area(s) that you would like to receive members comments on. You will be able to review these comments by selecting Latest Comments from the DADAS Home Page.

1. ☒ Functional Requirements (ALL)
2. ☒ Project Members
3. ☒ Legal Issues
4. ☒ Service Information
5. ☒ Individual Members (ALL)
6. ☐ Project Time Line

☒ Post by email notification of new comments.

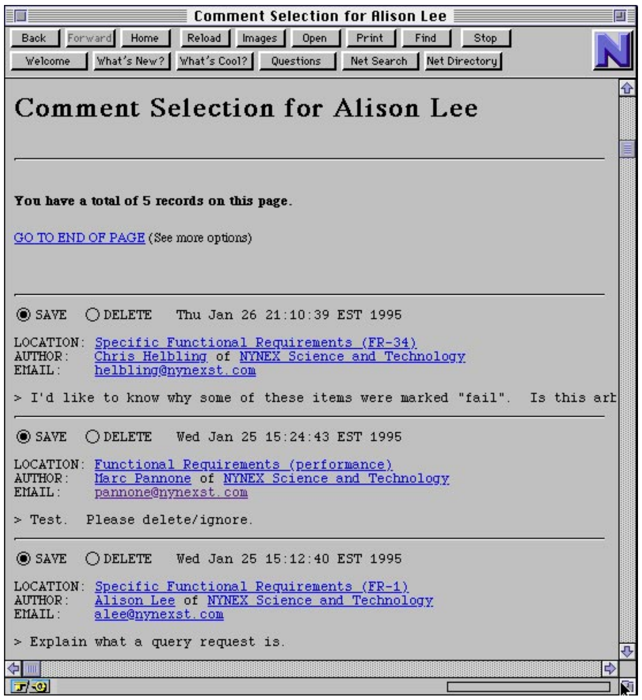
☒ Send a copy of my comments and confirmation to me by email.

☐ Attach all new comments to my latest comments page.

Press this button to Set Preferences

Return To The DADAS Home Page

Figure 7: An example of the Preference Document for Alison.



Comment Selection for Alison Lee

You have a total of 5 records on this page.

GO TO END OF PAGE (See more options)

```
[● SAVE] ○ DELETE   Thu Jan 26 21:10:39 EST 1995

LOCATION: Specific Functional Requirements (FR-34)
AUTHOR:   Chris Helbling of NYNEX Science and Technology
EMAIL:    helbling@nynexst.com

> I'd like to know why some of these items were marked "fail".  Is this arb

[● SAVE] ○ DELETE   Wed Jan 25 15:24:43 EST 1995

LOCATION: Functional Requirements (performance)
AUTHOR:   Marc Pannone of NYNEX Science and Technology
EMAIL:    pannone@nynexst.com

> Test.  Please delete/ignore.

[● SAVE] ○ DELETE   Wed Jan 25 15:12:40 EST 1995

LOCATION: Specific Functional Requirements (FR-1)
AUTHOR:   Alison Lee of NYNEX Science and Technology
EMAIL:    alee@nynexst.com

> Explain what a query request is.
```

Figure 8: Example of Alison's personalized news page.

that they are not interested in. The purpose of this page is to allow team members to check out discussions that they may want to begin actively participating in.

EXPERIENCES WITH THE DADAS DESIGN INTENT SYSTEM

Our group is continuing to develop the Design Intent system. Based on our experiences with using the system in a real development project, we have identified a number of issues that affect the implementation, usability and usefulness of such a system.

Technical Issues

In our Design Intent system, the discussion seeds are early drafts of what will become project documents. It is important that different team members are able to seed discussions in a timely fashion and to do so without too much effort. Most of the time, we helped the authors to prepare these documents using the hypertext elements of the Design Intent deployment platform (e.g., Netscape, Symbolics Concordia, FrameMaker). Other times, the authors prepared the documents with a word processor that they are familiar with and used automatic converters to convert them into a hypermedia format. The converters do not make full and effective use of a hypermedia representation and did not permit the authors the flexibility to make unlimited changes. As yet, we have not found a suitable approach that allowed the authors to perform unlimited changes to the hypermedia structure of the document without having to learn to use the capabilities provided by the hypermedia system (i.e., basic functionality, scripting language, developer toolkit).

It is also important that many different team members can actively contribute to discussions. Our approach in the DADAS project was to provide a form-based interface (the CGI component of the World Wide Web tool kit) for adding comments (i.e., annotation mechanism) and the Design Intent system provides the appropriate formatting of the information. Discussants do not have the ability to control the formatting unless they include the formatting attributes (described using the Hypertext Markup Language — HTML) along with their comments.

We examined a number of different deployment platforms for developing the Design Intent system and we have focussed on two. Each of them has their strengths and weaknesses. FrameMaker has a rich set of document formatting capabilities and graphics elements. Animations can be included easily. The user interface is customizable and extensible with the developer's kit. Web browsers do not support very sophisticated document formatters and graphics are restricted to bitmaps. We were not able to change the basic behaviors of the web browser but we are able to customize and extend the Design Intent system behavior using the elements of the World Wide Web (WWW) development platform. Furthermore, we found that the WWW platform allowed us to rapidly prototype new functionalities for the Design Intent system as well as to quickly react to user feedback on the functionality of the Design Intent system [14].

The Design Intent system is used by team members who are geographically separated and who used different computing platforms. We found that once team members are set up to run the web browser software, there is no need to perform any system administration activities to access new versions of the Design Intent documents and the Design Intent software. There is no proliferation of copies or different versions of the document or software since users access the document from the web server and the functionality of the Design Intent software is provided by the web server. Network communication is provided by the WWW platform, so that no additional mechanism or procedure is needed to distribute new versions of the documents or software. In another paper [14], we discuss how the components of the World Wide Web actually support the effortless deployment and maintenance of the type of collaborative applications that the Design Intent system represent.

Usability Issues

There are basically three levels of "change interactions" that can be supported. Users can be restricted to just reading documents. This does not enable the users to engage in a dialog. At the same time, this approach puts the least burden on the user to learn new tools. In the next category, users can provide input through a restricted interface (e.g., annotations or bulletin board mechanism). Such an interface would most likely be form-based. It is easy to use and it does not put the users in the danger of inadvertently damaging the document. But it is also restrictive in what the users can add to a document. The last category gives the users the most freedom but might require users to learn a lot. For example, they would have to learn HTML (for web browsers) to create and modify documents. For more sophisticated uses, they need to learn how to create HTML form templates and how to process the form input using a programming language. During the limited time in which the Design Intent system was used in the DADAS project, users were quite satisfied with the use of the restricted interface.

To make the transition from a document reader to a document author easy, it is important that the interface for authors resembles the interface for readers. In this way, users only have to learn a few additional skills to make this transition. The web browsers presented problems because a completely different interface must be used to create documents (textual markup language called HTML) than for viewing the end result. Furthermore, users need to learn the markup language in order to create documents. To overcome some of this burden, automatic converters from other document formats plus a restricted interface to touch up the results might offer some relief to the document producers. On the other, this is not a problem when the Design Intent system is built on top of FrameMaker as it is a WYSIWYG system. However, web browsers do provide support for viewing a wide variety of different media formats as well as graceful handling of media formats on impoverished platforms.

As previously discussed, the Design Intent system needs a mechanism for allowing its users to include documents they produce using some other word processors. In the web browser implementation, automatic converters cannot produce the same output quality as the original document system. Hence, users need to keep the original around for making subsequent changes. In this case, a semi-automatic process that would require additional work after each conversion does not provide an acceptable approach because it requires the maintenance of two copies of the same document. A fully automated process is desirable.

Another problem with document converters is that a document is written to be read linearly whereas a hypermedia document, with embedded links to other material, is not necessarily read linearly. At best, a converter can attempt to suggest possible links around structural elements of a document (e.g., sections, subsections) but could not do an adequate job on converting it to be used in a hypertext format. The care and attention that is paid to author a linear document is also required for making that same document accessible in hypertext format.

People Issues

The Design Intent system users must see the potential benefits before they are willing to put in the extra effort in contributing their information. After all, from the perspective of those who have to do the bulk of the work, the system creates additional work for them with no immediate payoff. Such benefits can be the timely distribution of information, being able to keep up-to-date, and being able to retrieve relevant project information at a later time. Another important benefit that is not immediately apparent to users is that the documents can seed design discussions that lead to a better understanding of the problem and the development of a workable design. The benefits of the Design Intent system have to be apparent and real to the users and the extra effort for contributing to its construction and evolution has to be minimized.

In our efforts to get team members in real projects to use the Design Intent system, we need to also involve people that are not team members of the project. Some of these people are network administrators who have to help with setting up the necessary infrastructure for distributed communication. Because there is no apparent benefit for these people, it is difficult to convince them to contribute to the success of the project.

The mechanisms and information that we built in the Design Intent system to help create better awareness within the team of the members of the team were well received. In fact, DADAS team members commented the usefulness and accessibility of the contact information (e.g., phone numbers, electronic mail address). It is unclear how useful the personal information about the team members (e.g., photos, job description) are in terms of building group cohesion during the limited period in which the system was used.

CONCLUSION

Our SPARX experience reinforces the need to target and understand problems that are not only relevant, but to which current technology can be applied successfully. We must take responsibility for making our understanding available to decision makers in such a way that they can readily see its innovative value. We believe that this understanding is constructed, evolved, and communicated through the design of boundary objects. We also realize that although an effective process and the design of boundary objects are necessary to the software development efforts, they alone are not sufficient without the communication infrastructure to foster an environment in which problem understanding, mutual education, and collaboration and negotiation take place. In DADAS, we began developing such an infrastructure.

ACKNOWLEDGMENT

Marc Pannone assisted with the development of several of the mechanisms in the DADAS Design Intent system.

REFERENCES

1. Atwood, M.E. *Opening Address*. In Breckenridge Workshop, Reengineering Companies, Universities, and the Relationship between Them. 1994. Breckenridge, CO.

2. Atwood, M.E., B. Burns, A. Girgensohn, A. Lee, T. Turner, and B. Zimmermann. *Prototyping Considered Dangerous*. In Proceedings of INTERACT'95: Fifth IFIP Conference on Human-Computer Interaction. 1995 (in press). Amsterdam: North-Holland.

3. Berlin, L.M., R. Jeffries, V.L. O'Day, A. Paepcke, and C. Wharton. *Where Did You Put It? Issues in the Design and Use of a Group Memory.* In Proceedings of INTERCHI 1993 Human Factors in Computing Systems. 1993. New York: ACM. pp. 23-30.

4. Boehm, B.W. *Software Engineering Economics*. 1981. Englewood Cliffs, N.J.: Prentice-Hall.

5. Carroll, J.M., S.R. Alpert, J. Karat, M.v. Deusen, and M.B. Rosson. *Raison d'Etre: Capturing Design History and Rationale in Multimedia Narratives*. In Human Factors in Comput-

ing Systems, CHI'94 Conference Proceedings (Boston, MA). 1994. New York: ACM. pp. 192-197.

6. Carroll, J.M. and T.P. Moran. *Introduction to this Special Issue on Design Rationale.* Human-Computer Interaction, 1991. **6**(3&4): pp. 197-200.

7. Conklin, E.J. and K.C.B. Yakemovic. *A Process-Oriented Approach to Design Rationale.* Human-Computer Interaction, 1991. **6**(3&4): pp. 357-391.

8. Curtis, B., H. Krasner, and N. Iscoe. *A Field Study of the Software Design Process for Large Systems.* Communications of the ACM, 1988. **31**(11): pp. 1268-1287.

9. Dourish, P. and S. Bly. *Supporting Awareness in a Distributed Workgroup*. In Proceedings of CHI 1992 Human Factors in Computing Systems. 1992. New York: ACM. pp. 541-547.

10. Fischer, G. *Rethinking Education and Computing Environments from a Lifelong Learning Perspective with Domain-Oriented Design Environments*. In Breckenridge Workshop, Reengineering Companies, Universities, and the Relationship between Them. 1994. Breckenridge, CO.

11. Fischer, G. *New Perspectives on Working, Learning, and Collaborating and Computational Artifacts in Their Support*. In Proceedings of Software-Ergonomie'95. 1995 (in press). Stuttgart, Germany: Teuber Verlag.

12. Fischer, G., S. Lindstaedt, J. Ostwald, M. Stolze, S. Sumner, and B. Zimmermann. *From Domain Modeling to Collaborative Domain Construction*. In Proceedings of DIS'95. 1995 (in press). New York: ACM.

13. Fish, R.S., R.E. Kraut, and B.L. Chalfonte. *The Video Window System in Informal Communication*. In Proceedings of CSCW'90, Conference on Computer-Supported Cooperative Work. 1990. New York: ACM. pp. 1-11.

14. Girgensohn, A., A. Lee, and K. Schlueter. *Developing Collaborative Applications Using the World Wide Web "Shell"*. In UIST'95 Conference Proceedings. 1995 (submitted). New York: ACM.

15. Girgensohn, A., D. Redmiles, and F. Shipman. *Agent-Based Support for Communication between Developers and Users in Software Design*. In Proceedings of the 9th Knowledge-Based Software Engineering (KBSE-94) Conference. 1994. Monterey, CA: IEEE Computer Society Press. pp. 22-29.

16. Girgensohn, A., B. Zimmermann, A. Lee, B. Burns, and M.E. Atwood. *Dynamic Forms: An Enhanced Interaction Abstraction Based on Forms*. In Proceedings of INTERACT'95: Fifth IFIP Conference on Human-Computer Interaction. 1995 (in press). Amsterdam: North-Holland.

17. MacLean, A., R. Young, V. Bellotti, and T. Moran. *Questions, Options, and Criteria: Elements of a Design Rationale for User Interfaces.* Human Computer Interaction, 1991. **6**(3&4): pp. 201-250.

18. Reinfrank, J.A. *Conversation with John Seely Brown.* Interactions, 1995. **II**(1): pp. 42-51.

19. Root, R.W. *Design of a Multi-Media System for Social Browsing*. In Proceedings of CSCW'88, Conference on Computer-Supported Cooperative Work. 1988. New York: ACM. pp. 25-38.

20. Terveen, L.G., P.G. Selfridge, and M.D. Long. *From "Folklore" to "Living Design Memory"*. In Proceedings of INTERCHI 1993 Human Factors in Computing Systems. 1993. New York: ACM. pp. 15-22.