A Framework for Designing Constructionist Approaches to Community-Centered Messaging

by

Pascal Roger Chesnais

B.S., Hofstra University (1985)
S.M., Massachusetts Institute of Technology (1988)

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2000

Author ....
Program in Media Arts and Sciences, School of Architecture and Planning
September 15, 1999

Certified by ....
Walter Bender
Senior Research Scientist
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by ....
Stephen A. Benton
Chairman, Department Committee on Graduate Students
Program in Media Arts and Sciences

A Framework for Designing Constructionist Approaches to Community-Centered Messaging

by

Pascal Roger Chesnais

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
on September 15, 1999, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

Abstract

Social constructionism is a term used to describe systems where individuals take an active role in development that can enhance their community's social setting. This thesis asks,"What is the role of communication technologies in a social constructionalist approach to community-centered messaging?"

Towards answering this question, a framework is described for developing and assessing such a system. This framework accounts for the varying skills of the community members, the amount of support the network provider is willing to invest, and the effort needed to use tools. A well designed system can meet the ambitions of the community that employs them, while a poor design leads to frustration and apathy.

An experimental platform, Canard, has been built in order to explore this framework. It provides a unified communication representation and tools that allow rapid development of community-centered communication applications that do not require significant expertise to use them.

The Canard system has been used by diverse communities over the past two years with varying degrees of success. The degree of success is a function of the group's ambition and willingness to participate in the process of crafting communication solutions for personal and community benefit.

Thesis Supervisor: Walter Bender
Title: Senior Research Scientist
Program in Media Arts and Sciences

Thesis Committee

Thesis Supervisor ....

Walter Bender

Senior Research Scientist

Program in Media Arts and Sciences

Reader ....

Edwin Selker

IBM Fellow

International Business Machines

Reader ....

Christopher Schmandt

Principal Research Scientist

Media Laboratory

Acknowledgments

The Media Laboratory has been my "home" at MIT since 1986. During that time I have been fortunate to have worked alongside many talented people. Attempting to thank them all would result in an acknowledgment section far longer than the rest of the document. It is with regret that I trim the list to those directly contributing to this research.

Walter Bender has worn many hats, friend, colleague, supervisor, and most recently, advisor. He has seen me through many years of growth, much of it made possible with his steadfast support through even the most difficult of times. This document has gone through many reviews by Walter. His suggestions made this document more enjoyable to read. Outside of the Media Lab he has a terrific family: his wife, Wanda, and children, Elise and Daniel – make for a wonderful model for my family.

My other thesis committee members, Chris Schmandt and Ted Selker, have provided much guidance. Chris Schmandt has followed my progress at the lab since I first arrived at the lab and took refuge in one of his sound studios. His speech interface research inspired much of my own research in community messaging. Ted Selker, who seemingly appeared out of nowhere, added much excitement to my work. I will never forget the day when he bellowed, "Pascal, stop programming!" making me realize that I was close to being done. It was time to share in writing the many things I learned.

My general examination committee provided much direction for my research. Bill Mitchell introduced me to the joys of architectural design studio approach applied to software research. It often makes me regret having studied computer science instead of architecture as an undergraduate. Hiroshi Ishii opened my eyes to the wealth of research from the field of Computer Supported Cooperative Workgroups.

Nicholas, Elaine, and Dimitri Negroponte have been a second family to me while we were in Paris, France at the Centre Mondial Informatique et Resource Humaine in 1982. Elaine deserves special credit for that fateful dinner where she asked why I had not yet applied to MIT for graduate studies. The deadline was four days away. Needless to say, I made the application deadline.

It is well known that UROPs are what make the Media Lab tick. Their enthusiasm is unbounded. Throughout my time at MIT I have been fortunate to have the very best at my side. They work harder than anyone I know, and still have time to play. There are too many to name here, but two in particular pulled all the loose ends of Canard together and conspired to help me graduate "on time": Joshua Randall and Dana Spiegel. Although Matthew Mucklo graduated two years ago, he contributed to this project until he moved to the "other coast." David LaMacchia contributed to the early implementation of Canard, and also had time to write an instant conference system called iFlame for his MEng degree.

UROPs are not the only ones to contribute to the Media Lab. Special thanks go out to the Electronic Publishing Group. In particular: Felice Napolitano, Jack Driscoll, Douglas Koen, Jon Orwant, Vadim Gerasimov, Ingeborg Endter, and Dan Gruhl all contributed ideas to Canard. Doug has been a dear friend since he first showed up at the Media Laboratory as a freshman and had the misfortune of calling me a "gentleman". I guess he ought to know, because I know of no finer gentleman to have as a friend.

Two past collaborators who significantly changed how I program by making the world much simpler are Nathan Abramson, who gave us Dtypes, and Jonathan Sheena, who gave us the Glue libraries for *FishWrap*^{TM}. Both started at the Media Laboratory as UROPs. Both have the goal of making the world a little bit better than they found it.

Without Motorola's equipment grants this research could not have taken place. In particular, Jim Page got the ball rolling by believing this work to be important enough to get the network equipment at the lab before any formal agreements had been ironed out. Bob Growney, Frank Lloyd, and Sheila Griffin made sure that there was corporate and government support. Karen Henrick and Liz Altman made sure the logistics were all worked out between the multiple Motorola facilities. Lastly, Kim Goldinger filled some very large shoes at the Media Lab, and did a magnificant job of it. Kino Coursey is not a Motorola employee but ought to be one! Kino developed the paging system software we use to run our network.

The News in the Future consortium funded my reseach at the Media Laboratory for many years. The representatives of the 26 companies proved a good forum for me to articulate my ideas.

Although I moved from the Garden (E15-344) a year ago, it still holds a special place in my heart. It was my home for over a decade. The people there have an amazing group spirit. Andy Lippman was my advisor in 1986-1988, and knows how to keep people on their toes! I still rely on Henry Holtzman and Mike Bove for advice. The graduate students there have always been there to listen and argue about ideas. In particular Bill "Bubu" Butera and Stefan Agamanolis deserve special thanks.

The Speech Group, in particular Stefan Marti, was invaluable as willing and able research subjects. They provided some of the richest feedback during the field studies. What I can't understand is why they are still using Sun Sparc 1+ workstations!

John Watlington has been a cohort of mine for years. Some people thought we would never be able to go separate ways. In many ways we haven't, and I often regret not sharing an office with him anymore. His continual presence is sorely missed.

It is odd, but special thanks need to be extended to an informal institution known as "tetazoo." The students there long ago adopted me... well, maybe my ability to make cheesecakes! They are a great group to work with, and make for terrific lifelong friends.

My Family! I do not just mean Mom and Dad! It still amazes me the major changes that have happened in my life since entering the PhD program – a dog (Gracie) joined me at home, I got married to the most wonderful person (Heather), we had a daughter (Alexandra) and then another (Christine). Heather has made me very happy person.

During my doctoral studies two friends passed away. This document is dedicated to them:

Gert-Jan Zwart – my officemate for the briefest of time, renewed my passion for the lab.

David Morgan – was Motorola's in-house representative. He gave a very personal face to Motorola. To many at the lab, myself included, he was a mentor and friend.

Contents

1 Introduction **11**
1.1 Framework for Community-Centered Messaging . . . 15
1.2 Document Organization . . . 15

2 Design Framework **17**
2.1 Related Fields . . . 18
2.2 ASE Framework . . . 20
2.2.1 Ability . . . 21
2.2.2 Support . . . 22
2.2.3 Effort . . . 23
2.3 ASE Framework Applied to Prior Work . . . 23
2.3.1 Chat Programs . . . 24
2.3.2 MOOs – MediaMOO and MOOSE Crossing . . . 27
2.3.3 *FishWrap* . . . 29
2.3.4 Silver Stringers . . . 33
2.3.5 Multi User System In Community . . . 35
2.3.6 Wearable Computing Community . . . 37
2.3.7 Summary . . . 39

3 Canard's Experimental Platform **41**
3.1 Motivation . . . 41
3.2 Communication Model . . . 42
3.2.1 Source Layer . . . 43
3.2.2 Message Evaluation Layer . . . 45
3.2.3 Message Transport Layer . . . 45
3.3 Software Implementation . . . 46
3.3.1 Dtypes . . . 46
3.3.2 *FishWrap* Glue Utilities . . . 49
3.3.3 Canard Objects . . . 50
3.3.4 Any Tools . . . 51
3.4 System Databases and Servers . . . 52
3.4.1 Observer Server . . . 53
3.4.2 Timer . . . 54
3.4.3 Access Control . . . 54
3.5 Personal Databases . . . 54
3.5.1 Address Book . . . 55
3.5.2 Calendar and Events . . . 55

3.5.3 Messages . . . 56
3.5.4 Archive . . . 57
3.6 Communication Channels . . . 57
3.6.1 Settings for Communication . . . 57
3.6.2 Physical Size . . . 58
3.6.3 Two-way Paging System Interface . . . 58
3.6.4 Electronic Mail Interface . . . 60
3.6.5 Instant Messaging Interface . . . 60
3.6.6 Telephone . . . 61
3.6.7 World Wide Web Interface . . . 62
3.6.8 Electronic Whiteboard Interface . . . 62
3.6.9 LED Sign Interface . . . 63
3.6.10 Family Radio Service Interface . . . 65
3.7 Summary . . . 66

4 Field Studies **69**
4.1 Experimental Protocol . . . 70
4.2 Third East Undergraduate Living Group . . . 72
4.2.1 Overview . . . 72
4.2.2 Ability . . . 76
4.2.3 Support . . . 76
4.2.4 Effort . . . 77
4.2.5 Constructions . . . 78
4.2.6 Other Observations . . . 80
4.2.7 Reflections . . . 84
4.3 Junior Summit . . . 84
4.3.1 Overview . . . 84
4.3.2 Ability . . . 85
4.3.3 Support . . . 85
4.3.4 Effort . . . 92
4.3.5 Constructions . . . 94
4.3.6 Reflections . . . 96
4.4 today@mit.edu: Freshman Campus Calendar Project . . . 99
4.4.1 Overview . . . 99
4.4.2 Ability . . . 100
4.4.3 Support . . . 100
4.4.4 Effort . . . 101
4.4.5 Constructions . . . 101
4.4.6 Reflections . . . 101
4.5 Speech Group . . . 103
4.5.1 Overview . . . 103
4.5.2 Ability . . . 103
4.5.3 Support . . . 104
4.5.4 Effort . . . 104
4.5.5 Constructions . . . 105
4.5.6 Reflections . . . 106
4.6 The Cube . . . 106
4.6.1 Overview . . . 106

4.6.2 Ability 107
4.6.3 Support 107
4.6.4 Effort 108
4.6.5 Constructions 110
4.6.6 Horror Stories 111
4.6.7 Reflections 112
4.7 Summary 112

5 Conclusions **115**
5.1 Future Work 117

A Third East's Proposal to Participate **119**
A.1 Hall Profile 119
A.2 Hall Communal Activities 119
A.3 What We Expect 120
A.4 What You Can Expect 120
A.5 Time Commitment 121
A.6 Life With Canard 121

B Junior Summit Charts **123**
B.1 Outbound Message Destinations 123
B.2 Messages to Teams and Leaders 124
B.3 Salutations and Parties 124
B.4 CheckIn and Find Applications 124
B.5 Messages to LED displays 125

Chapter 1

Introduction

When I arrived at MIT in 1986, I worked in an area of the Media Laboratory called "The Garden." It was a large common space surrounded by graduate student offices. At the time, most of the research equipment was housed in the common area. Consequently, everyone worked there. The student offices were mainly used for studying or telephone conversations. The telephones at the time did not have distinctive rings; as a result, one ringing phone among eleven offices caused much confusion and anxiety. However, the creative minds of The Garden had a solution.

The students connected lamps in parallel with each phone line and hung the lamps from their office windowsills. When a phone rang, students merely needed to glance up to see if their lamp was flashing. As a newcomer to The Garden, I was impressed by the individual touches given to the lamps. Some were fashioned as chandeliers; others were integrated into disk drive parts. No two were the same.

Then the system evolved yet again. Two students put together a system that scanned all the phones for a ring signal and announced the incoming calls with a speech synthesizer. The result: instead of looking up for a flashing lamp, people simply listened for a synthesized voice speaking their number. Then, because it was possible to program the system to speak any arbitrary phrase, individual creativity struck again. One student even had the speech synthesizer sing the first line to the television sitcom, *Green Acres.*

This wonderful, community-built, system fell into disuse when MIT converted its telephone system from an analog one to Integrated Services Digital Network (ISDN). There was no practical way of transferring the analog system to the new digital phones. While the new system offered a partial solution to the call announcement problem, the new phones had a distinctive ring feature, no one bothered to read the user manual to learn how to reconfigure their phones. Even if they had, there were only eight ring styles for eleven offices. A decade later, The Garden still has not replaced the call announcement system.

This early experience provided me with a glimpse of social constructionism at work and how assumptions made by system designers can have both positive and negative effects on a community's ability to participate in the development of its communication environment.

Prior to the switch, MIT had two separate telephone systems, one for the student dormitories and one for the rest of campus. The student phone system suffered from reliability problems and the main campus phone system was experiencing a high demand for telephone lines dedicated to computer modems and facsimile machines. It was time to upgrade the telephone system. In preparing for the switch to ISDN, MIT Telecommunications Office conducted a survey of how phones were used across campus. From that survey was distilled a set of features that MIT believed its community needed. This requirement-gathering process seemed like a good idea for the school as a whole, but it failed to pick up on the specialized needs of small groups such as The Garden, who were used to adapting technology to their needs.

It was not only The Garden's specialized needs that were trampled on, but also the mainstream users. MIT's requirement-gathering design process determined that there was a need for a specific number of computer modems and answering services. In the ISDN phone plan was a centralized modem bank and a voice mail system designed to meet the theoretical communication needs of the MIT community. If these requirements were satisfied, there was no apparent need for MIT to provide analog phone lines to all the offices. Hence, digital phone units were installed with the promise that all telecommunication needs would be satisfied centrally. For those people who did not have an answering machine or modem, the new system did provide new services. However, anyone with an exisiting answering machine or modem was either forced to use the new system or to get an analog telephone line for his or her original equipment.

MIT might have benefited by taking a constructionist approach to building the system that would have been easier to adapt to the various micro-cultures present on campus. "Constructionism" is a term first coined by Seymour Papert as an expansion of Jean Piaget's "constructivism." Papert writes:

> *We understand "constructionism" as including, but going beyond, what Piaget would call "constructivism." The word with the v expresses the theory that knowledge is built by the learner, not supplied by the teacher. The word with the n expresses the further idea that this happens especially felicitously when the learner is engaged in the construction of something external or at least shake-able... a sand castle, a machine, a computer program, a book. This leads us to a model of using a cycle of internalization of what is outside, then an externalization of what is inside. ([Pap91])*

Mitchel Resnick expands on the definition of constructionism to include people surrounding the individual:

> *They might be constructing sand castles, LEGO machines, or computer programs. What's important is that they are actively engaged in creating something that is meaningful to themselves or to others around them. ([Res94])*

MIT's phone system planners had the community's best interest in mind. However, given the size and diversity of MIT's campus, there was no way that a single, one-size-fits-all plan could satisfy the needs of all the smaller sub-communities. Was a vehicle for individual and community initiative what was lacking in the MIT phone system?

I hypothesize that the constructionist approach to telecommunication systems enhances the understanding of underlying technology by system providers and consumers. Further, individuals and communities learn more about these systems and are better able to exploit them to their fullest potential by manipulating them to suit their needs. This helps explain why there were so many rich adaptations to The Garden's analog telephone system versus the ISDN system where individual exploration was not encouraged.

In The Garden the students had the necessary skills to adapt the phones to their needs and there was an intellectual climate that encouraged such activities. The adaptation of the telephone system became a group activity, in part because the improvements were there for everyone to see. The lights drew attention to themselves, causing those not as skilled, or self-motivated, to seek the advice of others on how to duplicate the system. However, there was more going on in The Garden than constructionism as defined by Papert and Resnick. The students who did the synthesizer work were motivated by the desire to benefit the community as a whole, not to learn about phone technology. This benevolent constructionism is called "Social Constructionism" [Sha95].

Bender et al. describe social constructionism:

> *In social and developmental psychology, constructivist models depict the individual as a builder of knowledge, not as a passive receptor. Constructivism argues that the active nature of the learning process in which individuals are engaged needs to be enhanced and facilitated. Constructionism places a critical emphasis on particular constructions of the individual that are external and shared[Pap80][Pap90].These external constructions involve both creative action and "recreative" reaction, leading to an interplay between internalized and external experiences in such a way as to promote further creative activity.*
>
> *Social culturalists, such as Vygotsky[Vyg78], argue that learning to communicate is based upon internalized intellectual structures that allow messages to take on meaning. They argue further that these structures are initiated by external social and cultural relations.*
>
> *Social constructionism[Sha95] combines the sociocultural and constructivist views: social settings can be enhanced by the developmental activity of the individual. Likewise, enhancing the social setting by introducing activities that are socially constructive can enhance individual developmental activities.*
>
> *An individual's theories about his or her own development of knowledge profoundly affect that person's connections to and interactions with the world. How individuals believe they learn about the world around them, or become known by that world, helps to determine many of their goals and ambitions. These theories are the individual's epistemologies, and they help the individual determine how to develop and use technology.*
>
> *Social constructionism is an epistemological paradigm that suggests that becoming acquainted with one's neighbors is an act of extending one's self. Computer networking is a technical tool that can support this endeavor. However, the network is not the active force; the people are. The critical agency is based entirely upon the prerogatives of the people involved. ([BCE+96])*

Social constructionism provides a focal point for a number of projects that have been undertaken at the Media Lab, including Multi User System in Community (MUSIC), *FishWrap*^{TM}, *India Journal*, and Silver Stringers [BCE+96] [DE98]. These projects share the characteristic that the individual is an active participant in the messaging system, rather than a passive recipient.

- MUSIC[Sha95] is a bulletin-board style system designed with a particular neighborhood in mind. Computers were donated to families within the community and the system served as a focal point for community dialogue about neighborhood issues.

- The *India Journal* project created a specialized press for an Indian community in Jersey City, New Jersey. *The Jersey Journal*, a local paper, provided a staff editor and the technology for publishing on-demand papers in a local delicatessen and community center. The paper provided wire stories from India and the editor collected stories from local residents, enlisting their aid as "stringers" for the community publication. The role of community members was as story contributors. They were not engaged in the editorial or production aspects of the publication.

- *Fish Wrap*[CMS95] is a personalized, on-demand automated news system that provides geographically and topically relevant news stories for the MIT community. It was designed by a group of MIT freshmen for their peers. The news is organized by academic interests rather than traditional broad news industry categories (e.g., sports, financial, international, etc.) in hopes that it enables MIT freshmen to make more informed decisions about their academic pursuits. Students customize the publication to reflect their academic and personal interests. *Fish Wrap* readers are also editors of *Fish Wrap's* front page called "Page One", a page viewed by all readers. Any reader can recommend an article for placement on Page One and the presentation of the article varies based on the article's popularity. Readers are encouraged to add their personal comments about the articles on Page One. The readers are active participants in the publication.

- Silver Stringers is a software package for authoring on-line community news publications by senior citizens. The software exposes the process of writing, editing, and publishing step-by-step. As important as this automated function is, it is the weekly face-to-face meeting that makes the publication a community activity where they strengthen their skills and express their desires.

In order to develop a communication system that supports community participation, it is necessary to account for the varying skills of the individuals that will be using the systems, the amount of support the network provider is willing to invest in helping users, and the efforts needed to use each of the tools. This suggests that the system may be best supported by a collection of tools rather than a single tool. The collection of tools should be designed to adjust to the different levels of expertise of the users. The tools should not be so specific as to limit the novice. The tools should introduce the necessary skills (e.g., database manipulation, rule processing, and presentation) that enable individuals to advance to more powerful tools. Network providers are well situated to provide the support to help the individual learn more complicated tools. However, more often than not, it is the individual's willingness to participate that determines the amount of effort he or she is willing to invest in learning and using tools. Willing individuals often assist or teach others in their communities. It is this collective ambition that increases the system's richness and satisfaction for the community.

The Canard Messaging System [Che97c] is a platform for exploring the dimensions of skill, support, and ambition in the domain of communication technology. It, like other emerging systems, addresses interoperability problems for multi-modal communication (e.g., paging, email, and web) by providing a uniform message model. The model is a three-layer approach such as one used by image-processing developers. A source format is transformed to a uniform representation, then series of manipulations are performed, and ultimately the result is converted into a final vendor-specific format. The implementation is based on Abramson's Dtype library [Abr92][Abr93], which offers developers multiple representations, both ASCII and binary for network distribution. It allows many different development languages to be used (e.g., C, Perl, and Java), matching the skills of the programmer employing them. This approach allows rapid development of community-centered application components without requiring substantial community expertise. Canard also allows the integration of personal databases (e.g., calendar and address book) for use in message evaluation. Querying databases of communication traffic and making the databases available to the individual or community in many different forms facilitates the development of community-awareness applications.

1.1 Framework for Community-Centered Messaging

The objective of this thesis is to provide a framework for designing and evaluating tools that enhance a community's social well-being. With Canard, I have examined communication technologies, both wired and wireless, that can support active and passive users of community-centered messaging. Canard was deployed within a diverse set of communities in a multitude of settings over the course of two years. What these settings have in common is that in each case the community members share a physical space and have communication needs that exist beyond that space. The measure of success of Canard is a function of community ambition and participation in the process of crafting communication solutions for personal and community benefit.

The groups that used Canard were: an MIT undergraduate residence in support of their group activities; a group of freshmen as part of a joint class project; the Media Lab's Speech Research Group in as a means of extending the capabilities of their existing tools; three Media Lab research groups sharing a common space as part of a wireless environment; and 100 children attending an international conference at MIT in order to coordinate their activities with each other and with their online collaborators.

The result of this research is the development of a new platform that affords a community the means for a greater participation in the development of the communications systems they use. It gives community members the ability to construct and adapt such systems to their needs. This platform has been evaluated within an original framework that provides the means to predict the success of systems intended to serve new communities. This framework is useful as both a design and evaluation tool and for predicting the potential impact of existing technology in new community settings as well as new technology in pre-existing settings.

1.2 Document Organization

The rest of this document is organized as follows:

Chapter 2 describes a design framework for community-centered messaging that accounts for the community's skill level, extent of external support, and willingness to exert effort to adopt the technology. This design framework has underpinnings in ethnography, computer-human interaction, and computer-supported cooperative work. The framework is called "ASE Framework" representing the three major attributes: Ability, Support and Effort.

Chapter 3 describes the experimental platform used to explore the design framework and the specific off-the-shelf communication devices that were integrated in the field.

Chapter 4 includes the detailed field studies which used parts of the experimental platform to support various communities.

Chapter 5 contains concluding remarks and a description of future research directions.

Chapter 2

Design Framework

"Technology is anything invented after you were born" - *Alan Kay*

A critical point has been reached in the convergence of telecommunication and computer technologies. In 1999, computer adoption by household in the United States has surpassed 25% (in 1998 it was 42.1%[Irv99]), a point at which the technology can be deemed to be accepted by society[All99]. Access to the Internet has also surpassed the 25% mark with 26.2% of U.S. households having some form of access. This shift is an important one because it marks the transition of the use of advanced technologies from a formal setting (office) to an informal one (home). With this transition is the shift of training in use of the technology from formal (classrooms and workshops) to informal (reading manuals at home, or learning from friends). Telecommunication technology is also rapidly saturating U.S. society, and with it comes a transition from analog/voice transmissions to digital/short messaging that can be managed by personal computers. Although technically there is a potential for using personal computers to manage digital messaging, there are problems exploiting this potential. Most telecommunication systems are closed systems, not designed to be easily adapted by users.

Constructionists recognized the ability of individuals to learn from the manipulation of technology. This was illustrated early on with computer technologies such as the LOGO programming language that engaged children in the process of programming as a tool with which to think[PH91]. More recently, Programmable Bricks[RMSS96] provide physical constructions with computers embedded into Lego^{TM} bricks. This tendency to construct and consequently understand has been found in other technological areas such as automotive, aeronautic, and electronic (e.g., early radio technology had considerable participation from hobbyists). This trend continues with community-centered messaging systems, where individuals participate actively at all levels.

If as Dewey and others have argued [Dew38] that we learn through doing, then if we want to do more learning we should facilitate more doing. If we want this learning to be in support of community, then we should design platforms for social construction. Social settings can be enhanced by the developmental activities of the individual.

Experiences at The Computer Clubhouse[RR96] and Projeto Aprendiz[apr99] illustrated how an individual's learning can be paired with a desire to help the greater community. At both the Clubhouse and Aprendiz, children used computers to solve problems, and in the process they learned how to use new technologies. At Aprendiz, these activities occur within the context of civic action.

The usual path to helping one's community is through formal channels (e.g., local political or civic organizations). Informal initiatives by groups of individuals are another route to the same goal. A number of examples of community-centered messaging systems are presented in this chapter to illustrate this point. A design framework for social-constructionist approaches to community-centered messaging systems frames the discussion. The framework establishes guidelines that lead to a system's successful integration into an established community[BH93].

2.1 Related Fields

In order to develop a design framework for social constructionism, it helps to look at how other fields have approached the design process.

Industry developers make use of the classic "waterfall" design process for software development[Gru91] where the task of creating and implementing a system is passed down from group to group. The process is as follows: A vision for a product is proposed and a strategy is developed. The strategy is passed to a planning department. A product plan is drafted and is passed to the marketing department where a market for the product is determined. The marketing department then passes the project to a project manager. The project manager goes through a requirement-gathering phase. Once the requirements are determined, a software architecture for the product is developed. The software architecture is passed to a product development team that will be responsible for getting a group of programmers together to code the product. The programmers write the code and the software is tested using the target community.

There are some problems with the waterfall design process that make it ill suited for social constructionism. The most fundamental problem is that it distances the end users from the developers. It also separates those who come up with the product concept from those responsible for implementing it. Donald Norman uses modern digital telephone systems[Nor90], full of desirable features, as an example of where distancing the design of a system from its place of use has detrimental consequences. The users of such telephone systems are often baffled by their complexity and find it extremely difficult to understand how to employ the most basic features. The requirement-gathering phase in such a system happens without seeing how people actually use phones on-site.

The Human Computer Interaction (HCI) community focuses on understanding the efficiencies of man-machine interfaces and making the mechanics of the user interface better. Typically, computer interfaces are analyzed in terms of how the interactions benefit the user experience and the task being performed (e.g., does a delay between a simple database request and receiving the actual document impact a person's perception about the accuracy of the document retrieved[RBP98][Roa98]).

The participatory design method[Mul91] is a common technique employed in the HCI field. Users of technology are invited to participate with developers in designing a system. They employ tools such as Post-it Notes^{TM} to mock up user interface objects and represent processes as plastic icons (i.e., physical items that represent a database query). The session between system developers and users is videotaped in order to have a record of the design evolution. The participants are told that they are experts in a particular domain (e.g., the developers are experts in the systems used and the users are experts in the task). The advantage of such an approach is that it allows the users of the system to "co-own" the design.

A general criticism from within HCI is that its methods are less than adequate for the design of effective collaborative tools. Bannon and Hughes [BH93] list the areas of weaknesses as:

- too much focus on the individual and not the group,
- lack of contact with the 'real world' situations of use,
- tendency to confine itself to small experiments,
- use of novices rather than 'real world' users,
- neglect of the ways in which users can acquire expertise through experience.

Given these deficiencies in HCI and traditional waterfall design, it is clear that neither one is well suited for community-centered systems. A closer relationship between developer and end-user is needed. It would be more appropriate to take an iterative design approach (e.g., rapid prototyping methods[ABB+93]), where the software is incrementally improved in reaction to feedback from the field. However, even this approach is deficient since in traditionally closed systems, iterative design fails to help a community once the developers have left. Typically, the community is not given the opportunity or does not have the expertise to change the system.

An alternative approach is to design an open, extensible system that allow communities to be self-sustaining with little or no outside intervention. Open-source endeavors, such as, GNU, Perl and Linux, fit this description and usually have a strong user community associated with them[Ray98]. This approach may not generalize since these communities most often attract people who are already proficient in using their tools and who already have the necessary skill to make changes that extend the tool's functionality.

Another field which holds potential benefits is the Computer-Supported Cooperative Work (CSCW) community[Gre88][Gru94]. Here studies are conducted in understanding how computers can aid a group cooperating on tasks either formally or informally. Gavers [Gav91] writes that "shared work involves the fluid transitions among focussed collaboration, division of labour, serendipitous communication, and general awareness." The transitions among awareness, communication and collaboration increase in both formality and planning required to cooperate on a task. It is implicit that shared work is not a single system but is many interconnected systems that can occur at different locations and levels of involvement. CSCW relies heavily on ethnographic methods to observe people using technology in their own settings[KL98]. From these observations, formal systems are designed and deployed into the community.

CSCW focuses on "work" settings and is not particularly focused on technologies that enable communities to be self-sufficient in informal social settings. Social constructionists focus on communities that may not have specific group goals, as was the case with The Computer Clubhouse and Project Aprendiz both of which nurtured the use of technology as a process for creating things. The individuals participating in The Clubhouse have their own agendas for the use the technology and they create artifacts of personal importance. At Aprendiz, the individuals share a common goal and they create artifacts that are of importance to the community as well as themselves.

2.2 ASE Framework

What are the design criteria for successful social-constructist systems? It is important that the developers have a closer relationship with the people using the system. Tools should be well suited to the abilities of the individuals using the systems. Ideally, the community takes on the role of the system developer and community members are afforded the possibility to extend and create new communities.

In order to answer these questions, a community-centered messaging system, Canard, was created and subsequently deployed in five communities. Community-centered messaging was chosen in order to engage the community within systems that traditionally have been seen as closed to adaptation. An experimental platform was used to create a superstructure that allowed the integration of closed telecommunication systems. These systems worked in concert with one another and allowed applications to be developed that exploited a unified messaging system. From the field study experiences, a design framework was developed to help explain why some of these pilot groups succeeded and why some seemingly did not.

By using the same experimental platform with multiple groups, it was possible to observe different levels of engagement by each group. The levels of participation included those who wanted to use the system, those who tried to extend the system, and those who created systems on their own. Most users fell into the first category. They did not have the desire to extend the capabilities of the system. This may be because they found the existing system adequate for their needs or they were not sufficiently motivated to engage in the complexities involved in fully understanding how the system works. Those who did wish to extend the system by construction were motivated by a number of factors, including curiosity and a desire to accomplish a particular task. There were a few community members who saw value in the system and wished to apply it to new communities and settings.

Different levels of participation are characterized by three attributes that reflect the abilities of the group, the level of support provided, and the effort communities were willing to invest in using the technologies to solve their problems. The attributes are selected in order to help understand how well the system is matched to the ambitions and goals of the community. Regarding ability, it is important to ask: How well suited are the tools to meet the ambitions of the community? How easily can a person participate in the system? Can they extend it? Can they create new communities with these tools? The support attribute is concerned with how much external support is required to sustain the system. Can the community use the technology without external intervention? If external intervention is required, is there a path for the community to take control and be self-supporting? Effort is assessed by asking questions such as: How will the community overcome the complexities associated with adopting a new technology? What factors motivate community members to put sustained effort into learning something new?

Throughout this document we use a three-dimensional chart to illustrate the ASE Framework (see Figure 2-1) where each axis represents an attribute of the framework. The three attributes are ability, support, and effort. Although there are qualitative distinctions between axes there is no attempt to quantify any of the axes nor is there an attempt to establish a formal relationship between them. Instead, the graphs are meant to stimulate discussion about the communities being studied and steer the discussions towards an understanding of how sustainable the technology will be for that particular community.

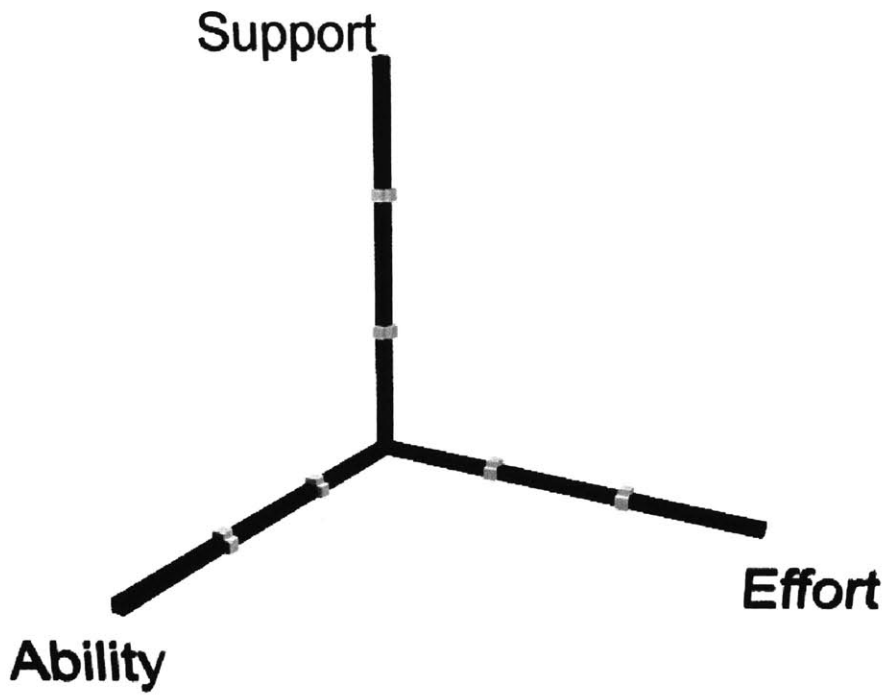


Figure 2-1: ASE Framework for designing community-centered messaging systems accounts for: ability of the individual or group, support structures, and effort of the users. Ability ranges from novice (at the origin) through "power user" to expert. Support ranges from self-supporting (at the origin) to spot intervention to formal support. Effort ranges from personal curiosity (at the origin) to altruism through effort driven by necessity.

2.2.1 Ability

The first attribute of the ASE Framework examines the ability of the individual or the community using the system. In the case of social-constructionist settings, where a diverse community of the users may be employing the system, a comprehensive set of tools must account for the varying abilities and desires of users participating in the well being of the community. In any community there will be three groups of users: those who will be content to participate casually in the system; those will wish to extend its capabilities; and those who will wish to create new systems for the benefit of new communities. In a well balanced system, members of each group should feel as if they are participating fully. So when analyzing a community, it is important to consider how to satisfy these three distinct groups.

Three categories of tools were created based on user skills:

Tools for novices These tools are well suited for the layperson with minimal skills. The tools can be fashioned to match the user's existing skills without requiring the acquisition of new specialized knowledge.

Tools for "power users" These tools are ideal for those who wish to learn more about the system and tailor it to their individual needs. These individuals have the skills to learn about detailed features of the system.

Tools for experts These tools are for the benefit of those who wish to completely immerse themselves in the underlying technology and have the ability to create new systems.

When designing a community-centered system, the first step is to survey the skills and abilities of its members. The second step is to consider what opportunities will be available for community construction and tool adaptation within the system. The designer must also account for the group's potential to acquire new skills. Communities are rarely static in their abilities. It is worthwhile creating opportunities for the community to grow, by providing opportunities for progressing from novice tools to "power user" tools. This is particularly important when implementing different genres of tools for a community.

2.2.2 Support

The second attribute of the ASE Framework concerns the autonomy of the community. Ideally, a system would be entirely self-supporting. Self-support is necessary when there are no means of external support (as illustrated with the Wearable Computing community described later in the chapter). Some systems are designed to be community supported from the start, as is often the case with some chat programs. However, in practice, some form of external support is usually helpful in order to assure a smooth adoption of the system by the community. The degree of external support can impact the sustainability of the system (e.g., if external funding runs out and the community cannot cover the costs of running the system without it, the system may fold). Spot intervention is less costly to the community in terms of reliance on external resources. It presupposes that the community is generally capable of running the system themselves, but acknowledges that assistance will be required from time to time to facilitate adaptation and augmentation of functionality.

Three levels of support were made evident during our field tests and are useful to consider when designing systems. When surveying a community, it is important to identify which level of support the community will require:

Full-time external support An outside institution provides full-time support to the community.

Spot external intervention An outside institution provides support only as needed by the community.

Self-supporting The community is solely responsible for all aspects of the system. By "self-supporting system", we mean a system that allows individuals to utilize and expand the capabilities of the system without outside intervention.

A self-supporting approach is preferable, not only because it reduces the burden of support on the system provider, but also because it is often difficult for community members to reach a consensus or to formally articulate their needs to an outside agency. This approach also recognizes that a community's needs and expertise are not static. A self-supporting system grows at a pace in step with its community.

When a self-supporting system is not feasible, outside intervention is necessary. This intervention can be beneficial to the community by providing the expertise that is needed as necessary. Providing a mechanism for sustained support that follows the community's own pace of growth facilitates adoption of new technologies. This implies that one can transition from a system requiring full-time external support to a self-supporting system by enabling the community to learn how the system works and providing them with the necessary tools for exploring and expanding it.

2.2.3 Effort

In his book *Diffusion of Innovations*, Everett Rogers proposes:

The complexity of an innovation, as perceived by members of a social system, is negatively related to its rate of adoption.[Rog95]

He further defines *complexity* as the degree to which an innovation is perceived as difficult to understand and use. He cites the adoption of personal computers (PC) in the 1980s as an example of how new users spent six to eight weeks learning to use their machines. These individuals were willing to work hard at overcoming complexities in order to get access to the benefits of the technology. The normal channels for finding information on how to use the PC - asking sales people, calling technical support lines, and reading the manuals - were daunting to the early PC owners. These channels employed overly complex technical explanations that the typical PC owner could not understand. Instead, these early personal-computer owners learned how to use their machines from local user groups that would meet in social settings. These user groups provided a self-help network that addressed the complexities of using personal computers in terms they could understand.

The final axis of the ASE Framework examines the effort an individual or group is willing to make towards using a new technology or participating in the community's well being. Many factors drive individual and group effort, making it seem to be a catch-all for factors that cannot be attributed to ability or support issues. However, the field studies suggest that a few distinct categories of personal and group motivation, each requiring more effort, did emerge to allow a better understanding of the target community:

Curiosity There seems to be a fundamental desire by some people to explore new technologies. This is the case with many early adopters of a new technology who have overcome the complexities of a system in order to use them. If a technology is too complex, it can be abandoned without great consequence.

Altruism The desire to help others is another driving factor of the willingness of a person or group to adopt new technologies. The effort invested by individuals parallels their motivation to help their community. Abandoning a technology in this case might not have much personal consequence.

Necessity In many cases, new technologies are adopted because a particular personal or community need has to be satisfied. In the case of personal needs, a unique technological solution to a problem is sufficient motivation to overcome the complexities of a system. In order to participate fully in a community, an individual may be compelled to adopt a technology.

In redundant systems, where the needs are met by multiple technologies, individuals are less willing to invest the effort in adopting a new technology. There is little motivation to try to overcome the challenges of a new system when an old one suffices.

2.3 ASE Framework Applied to Prior Work

Six community-centered messaging systems are used to illustrate the application of the ASE Framework. In each of the following examples, a community is either formed around a technology or exploits it to strengthen the ties between members. Each example has a description of a technology and the community that uses it. The ASE Framework is used to evaluate the match between the technology and the target community. This survey of communities was compiled from published reports rather than direct meetings with the users.

There are a few basic questions to ask about how limited an individual's role is within each of the community-centered messaging systems surveyed:

- Does the messaging system limit the community to simply communicating with one another?
- To what degree does the system allow individuals to adapt it to the community's need?
- Is it possible for individuals to use the technology for creating new communities?

Once the participation level permitted by the system is understood, then the ASE Framework attributes can be examined to see if there is a good match between the community and the messaging system.

2.3.1 Chat Programs

Chat programs, such as Internet Relay Chat (IRC) [OR93], are a good example of an easy-to-use community-centered messaging system. Chat programs generally consist of a simple user interface that allows two or more people to write messages to one another in real time. Depending on the sophistication of the chat program, graphics and hypertext links can be included in the messages. Special interest communities, which connect at the same time on the same server, can be easily organized. IRC organizes such communities into "Channels" that are topical in nature. Anyone can create an IRC channel. Discovering channels is accomplished by browsing directory servers that list all the available IRC communities.

An early example of an IRC community can be found in the political arena. IRC was used during Ross Perot's 1992 and 1996 presidential campaigns [Koe96] to create an electronic town hall. Members of United We Stand, a grass roots political group, which later became the Reform Party, would gather in living rooms around the country to meet on a special IRC channel. Each node on the electronic town hall represented a physical location at which more than one person was present. This allowed both physical participation on a local level and an electronic presence nationally. Participants with home computers were encouraged to share the experience with friends without computers. Consequently, a large number of people without computers were introduced to the benefits of Internet chat. Some of the new participants subsequently purchased their first personal computers so they could establish their own node in the electronic town hall meetings.

An unique feature of the electronic town hall system was the use of an automated moderator that maintained the IRC channel when no human activity was present. This allowed newcomers to the IRC channel to interact and learn something about the community without having to wait for a regular participant to help. The automated moderator was also a familiar presence for the regulars while they were waiting for others to join the channel for meetings.

Another example of a chat program is MIT's messaging system called Zephyr. Zephyr [DEF+88] was designed to allow students on the MIT network to message one another without having to know which machine they were connected to. On the Zephyr system, it is possible to send messages to an "instance," which is much like an IRC channel. An

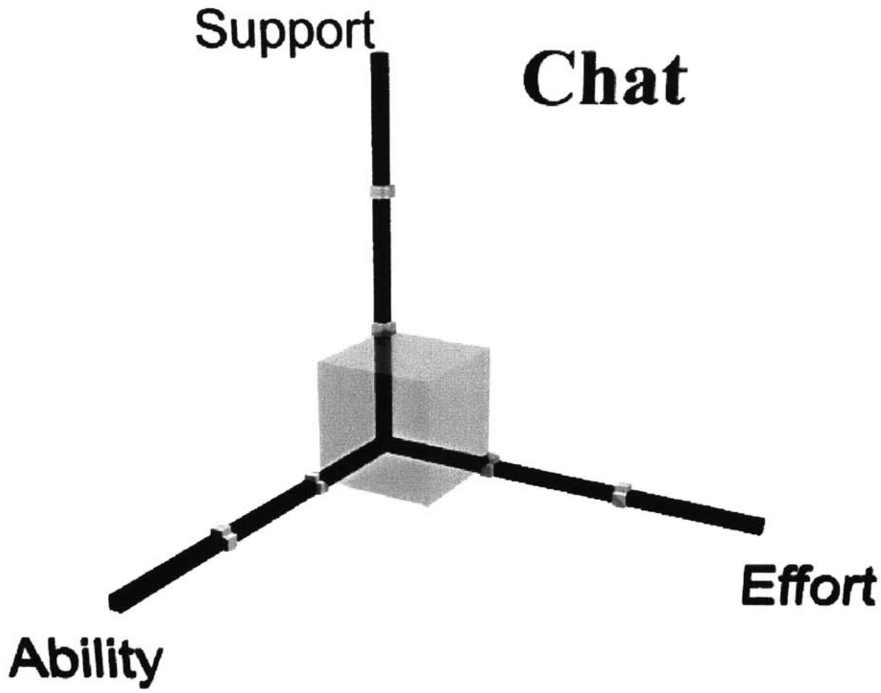


Figure 2-2: Chat Programs are easy to use and can be sustained with little outside intervention.

example of a community created around Zephyr is the Help instance. It is a forum in which people who have questions about almost any subject can ask these questions[AP96]. Groups of people regularly subscribe to this forum and take great pride in answering questions as quickly and accurately as possible. This instance provides a service to those needing help and provides an outlet for a group which has a desire to help others.

Ability

Using chat programs requires very few new skills beyond those already acquired to connect and use simple Internet services. The typical interface has both a composition element and a view of the community discussions.

Extending the capability of a chat program requires some programming ability. In the case of IRC, one can construct software agents, called "bots", to automatically handle requests on the channel. For example, an IRC bot can return a file containing the frequently asked questions for the forum in response to a user asking for help. In Zephyr, there is a scripting language to allow individuals to extend the functionality of the system. Community members are able to construct gateways and information services for the community.

Creating new communities with chat programs is, in most cases, easier than extending the program's capabilities. In the case of IRC, a step-by-step guide is available for individuals wishing to set up their own servers. In the Zephyr environment, one can create new discussion channels that can be private or public.

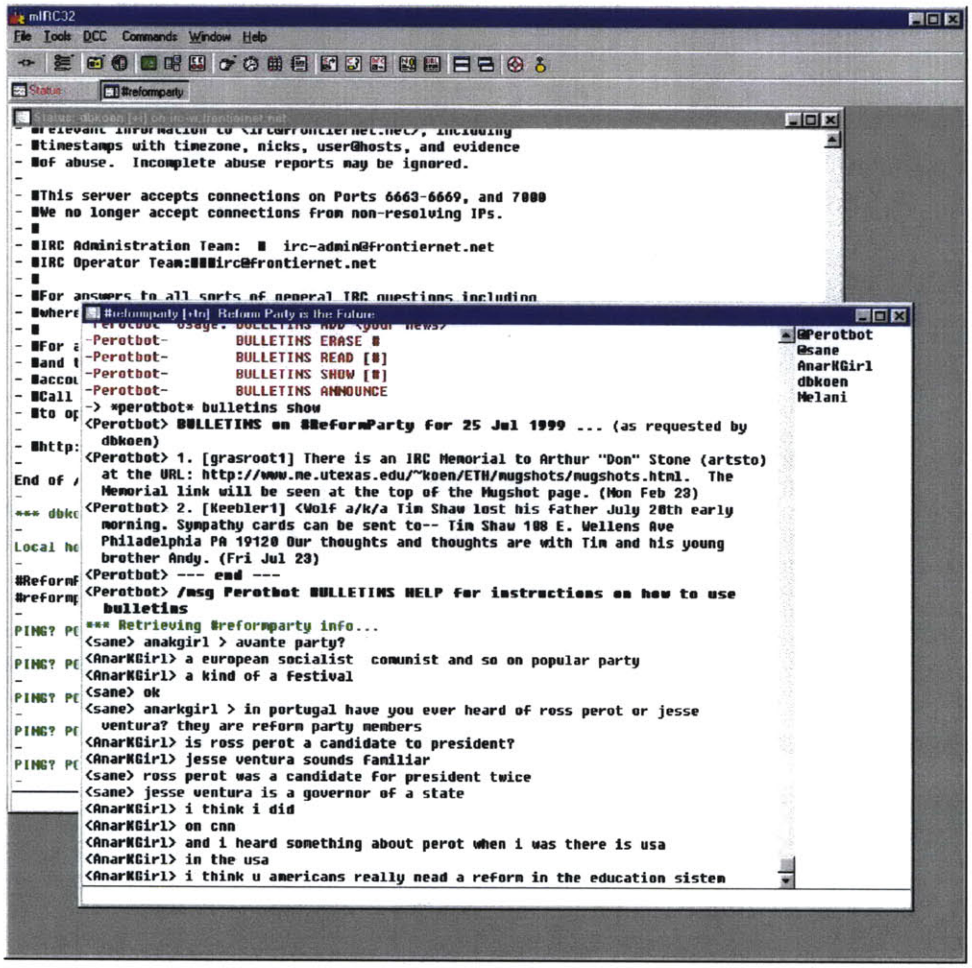


Figure 2-3: Reform Party IRC channel uses an automated "bot" to provide information about the community to new users.

Support

Chat systems are self-supporting communities. Once put into place, users have little need for formal intervention to support their usage of the system. It is common for participants to help new users out by introducing them to the etiquette of their systems (e.g., Zephyr Help Instance). IRC has many easy-to-find online help documents and getting help from online participants is equally as easy.

There are interfaces available for extending chat systems like IRC and Zephyr. Accomplishing this is left up to users by having them read complex online help documents.

The ease of creating new communities depends on the chat program. It is trivial to create new forums with IRC and this aspect of the system's design encourages the formation of new communities.

Effort

Chat programs are not very complex. It requires very little effort to participate in a chat systems, and they fulfill a user's fundamental desire to belong to a community.

2.3.2 MOOs – MediaMOO and MOOSE Crossing

MediaMOO and MOOSE Crossing [Bru97] are derivatives of Multi-User Dungeons (MUD) that are specifically tailored for their communities. Both MediaMOO and MOOSE Crossing encouraged interactive chats through the use of "rooms" and "objects." Rooms are roughly equivalent to chat channels. Objects are artifacts that can be transported from room to room by a user or left in a room for others to find. For example in MIRE[Kay92], newspaper objects that carry news from wire services can be created and rooms can be tailored to specific subjects. Unlike traditional chats, MUDs are designed to be extensible. Rooms are added and objects are created routinely by the users.

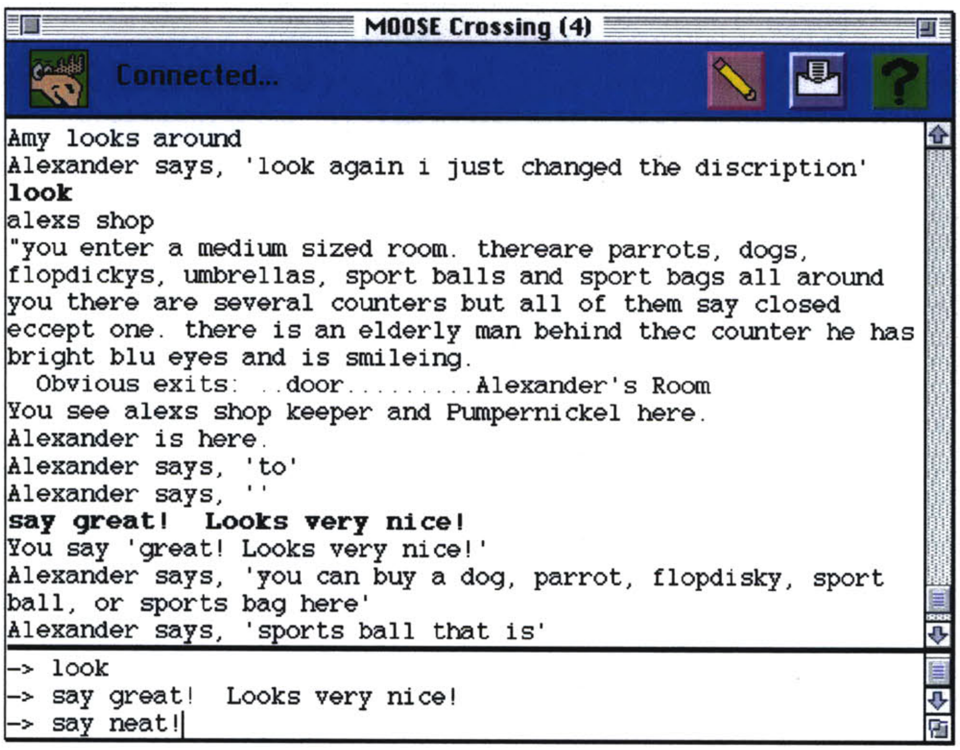


Figure 2-4: The MOOSE Crossing interface provides a chat-like interface for writing to other people in the system.

MediaMOO and MOOSE Crossing were created for specific communities. MediaMOO is a meeting place for people interested in the media technologies. MOOSE Crossing is an environment for children.

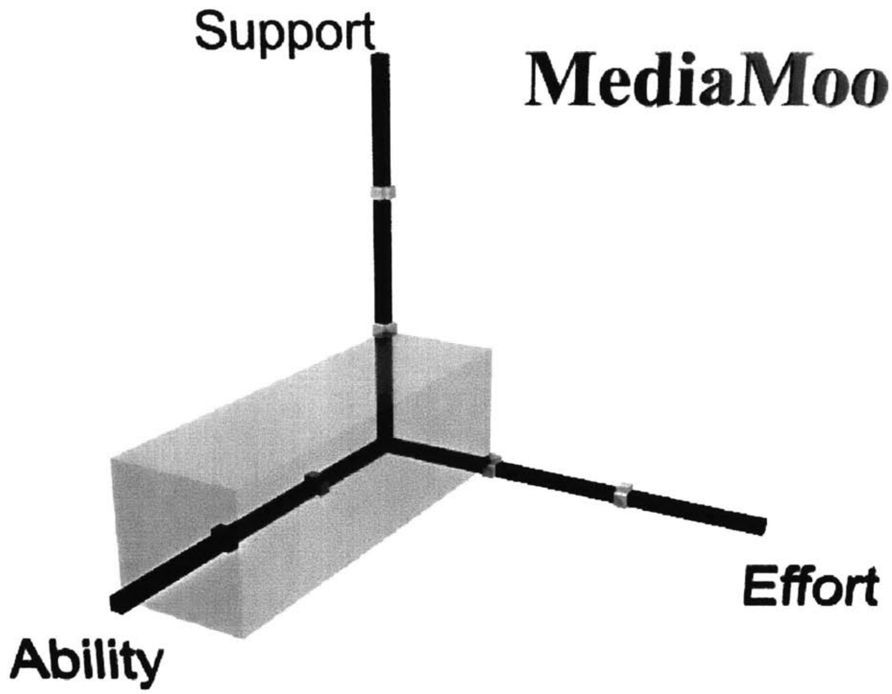


Figure 2-5: MediaMOO and MOOSE Crossing are examples of "open source chat" in which a broad spectrum of community expertise could be employed to extend the system.

Ability

Participation in MUDs requires few new skills. MUDs have an easy-to-use interface similar to chat systems (see Figure 2-4). A compose window is provided for the user to type messages, and the results are seen on a community view.

Extending a MUD can be done through the creation of new community rooms or the creation of new objects to be used in the MUD. Creating new rooms is easily accomplished through the use of simple administrative commands. Creating objects requires learning basic programming skills to use the object-authoring language. MOOSE Crossing demonstrated that children were able to create new objects in this environment and they took great pride in doing so.

Creating new MUDs requires computer administrative skills in order to set up the server.

Support

Minimal formal support is needed to run a MUD. It requires a dedicated server and a person to administer the account creations. Once the server is set up, the community can easily support the system.

Adding new objects in the system does not require formal intervention. Users are free to share their objects.

Effort

Like chats, the participants in a MUD are interested in interacting with their community. Participating in discussions in a MUD requires little effort to participate. Creating new objects in a MUD requires significant effort to learn how to use the scripting language. However, participants are willing to invest time in learning the scripting language to create interesting objects and show them off to others in the community. This was the case with the children in the MOOSE Crossings project.

2.3.3 *FishWrap*

FishWrap is a personalized news system put into operation in 1993. It was designed as a class project by a group of freshmen to help incoming freshmen transition into the MIT environment. Being familiar with the challenges faced by freshmen during their first few weeks on campus, they were able to identify some core features that would be beneficial. They designed *FishWrap* to have a personal focus, recognizing that students coming to MIT have stronger ties with their hometowns than their new home. Hometown news provided a stream of news from familiar locations back home. The designers recognized that freshmen have the daunting task of selecting a major at the end of the year. MIT is very good about informing students about what courses they need to take to get a degree, but it is not very good about informing them about the state of the industry. So *FishWrap* sorts news wire stories by academic offering at MIT. For example, an aerospace engineer can find news about the aerospace industry and public policy about aviation. The freshmen designing *FishWrap* also realized that their peers are seeking to learn about a new community and that exposure to topical issues debated on campus would be useful. *FishWrap* selects stories representing those topical issues (e.g., the abortion controversy) that reflect different points of view.

FishWrap is not limited solely to content provided by the news wire services. Students, staff, and faculty can also provide content for the system. Students act as reporters and columnists. The Dean of Students' office had an advice column that fielded many issues that were not easily handled by normal MIT channels. MIT's News Office contributed press releases promoting the accomplishments of the various departments.

Although *FishWrap* provides the reader with an egocentric view of the world by means of a personalized organization of news, it also features a shared community space called "Page One" (see Figure 2-8). All the readers view this page. It contains a collection of articles that are submitted by *FishWrap* readers. In this way, the task of article selection is not solely an automated one. Community members directly impact what the community at large sees. The organization of articles on the front page varies based on the popularity of individual articles. That is to say, the more a story is read, the more prominent a position it receives on Page One. In addition, readers are welcomed to provide comments about articles on the front page for others to read. This reflects the designer's view that the readers can add value to the community by being active participants in the editorial process of selecting stories.

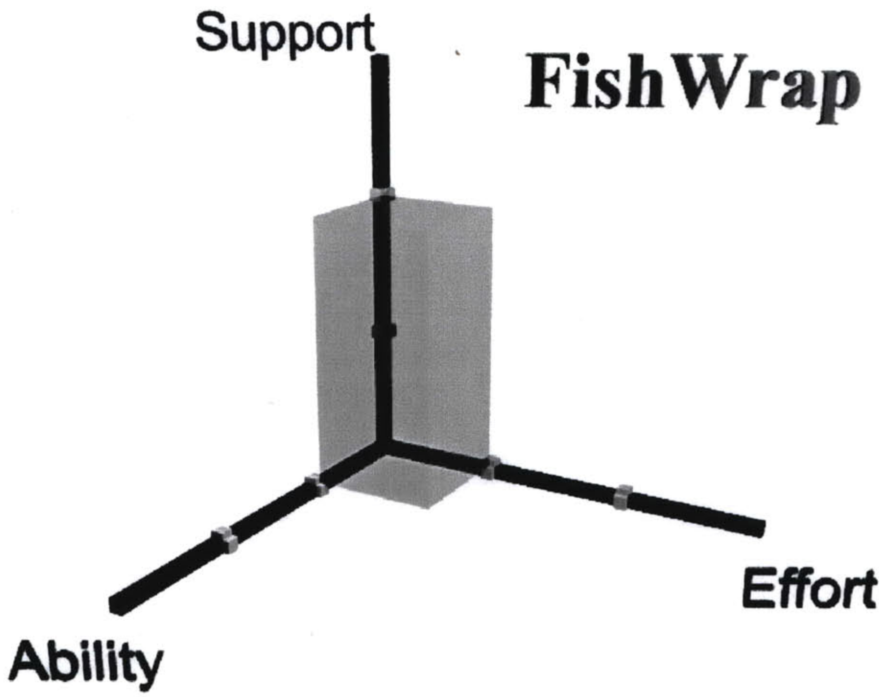


Figure 2-6: *FishWrap* is an example of a system that requires little external intervention.

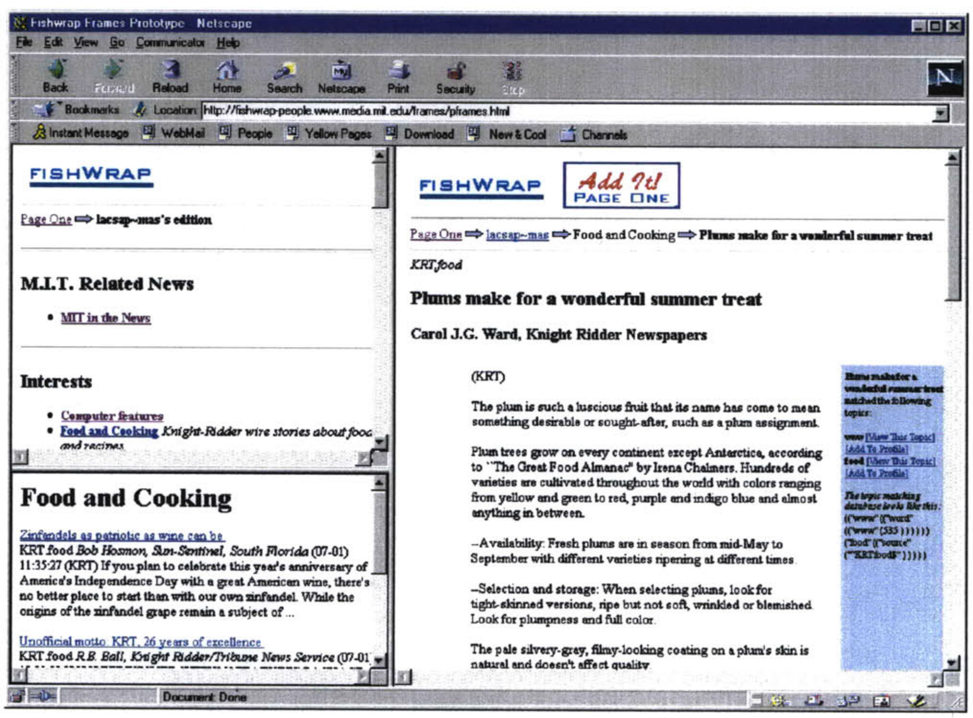


Figure 2-7: *FishWrap* provides the reader an ego-centric view of the world at large.

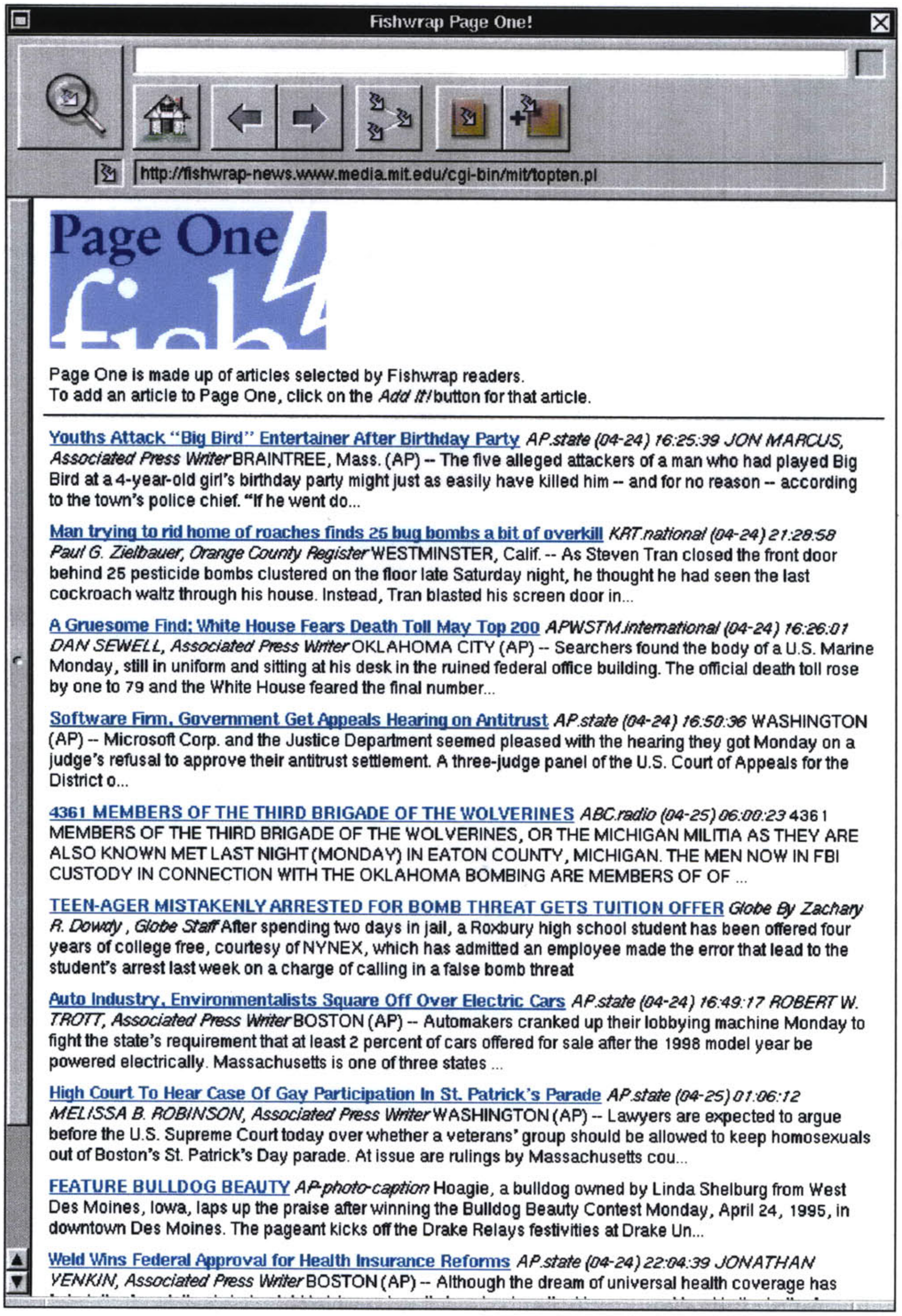


Figure 2-8: *FishWrap's* "Page One" consists of articles selected by its readers. Articles are then sorted by popularity.

Ability

Using *Fish Wrap* is relatively simple. The only skill required is the use of a Web browser. To create a skeleton profile, the user has to answer only three questions: "Where are you from?"; "What majors interest you?"; and "What is your MIT affiliation?" The first question establishes a "Hometown News" section. The second answer is used to create a selection of news related to their academic interests. The third question is asked in recognition that readers can be from any part of the MIT community. It allows the possibility of routing retirement news to employees and student government news to students. Extending an individual's news profile can happen in one of two ways. Every article displays how it relates to topics in the *Fish Wrap* system. The user can immediately add those topics to their profile. Alternatively they can use a profile editor, which is no more complex than selecting items from a list.

Participation in the extension of *Fish Wrap* can occur in many ways. Community members can add articles to the Page One section of *Fish Wrap* thereby bringing to light issues of the day. Sharing opinions about Page One articles is another means of active participation. Even passive participants have an impact. By merely reading articles on Page One, readers cause a change in the priority of articles. These forms of participation require no skills unique to the system.

Contributing content to *Fish Wrap* is relatively easy since *Fish Wrap* can accept submissions through a number of channels. Columnists usually submit their material through an electronic mail interface. News wires have their own interfaces, each requiring specialized knowledge of wire service formats [anp84] and how the news source codes stories.

Adding to the knowledge representation of the *Fish Wrap* system requires some expertise in keyword matching. However, the Web-based interface for creating new topics is easy to use.

Creating new *Fish Wrap*-like publications requires extensive UNIX administration knowledge and the ability to program in C in order to adapt the system.

Support

Once configured, *Fish Wrap* runs with minimal intervention. All the news wire feeds are designed to run unattended. Apart from adding new reader accounts, the system requires no administrative intervention.

Fish Wrap is intended to be a self-supporting system. However, in practice, spot expert intervention is needed to add new topics into the knowledge representation and to add and maintain news streams.

Effort

The MIT community was willing to use the *Fish Wrap* system in addition to other established news sources. When *Fish Wrap* was first launched, no other publication at MIT or anywhere on the World Wide Web had as extensive a news wire resource. For the MIT community, *Fish Wrap* filled a need for getting news from outside of MIT. In addition, *Fish Wrap* benefited from MIT student's interest in the novel use of technology.

Students were also willing to contribute columns to *Fish Wrap* as it provided a channel for their views. Past columns in *Fish Wrap* were general advice or help columns. Since the channel for submitting columns was made very easy, the students were able to focus on the writing rather than the formatting issues.

New topics and features were added to *FishWrap* by the student developers, because they wanted their peers to see the best possible news service. There was a fundamental desire on the part of the developers to provide a valuable service. There was no better motivation for innovation.

2.3.4 Silver Stringers

Silver Stringers is a community publishing project geared towards senior citizens[Las99]. The publishing activity occurs at senior centers in several communities in the United States and Finland. Unlike *FishWrap*, the community creates all of the content itself with a publishing tool called Pluto. Pluto is designed to assist a community in putting together an online publication by stepping the users systematically through a formal publishing process. The system is designed to focus the seniors' attention on the creative aspects of publishing. An important element of the Silver Stringer project is not what happens on the computer, but rather what happens in the community centers in which the publication is assembled. The publishing activity stimulates interaction among seniors, and it encourages the seniors to tell and share stories. Also, in participating in the project, seniors are learning how to manipulate computers and relate to the technologies their grandchildren are using.

Silver Stringers' Pluto system is used to publish a number of community online publications. Now in its fourth year, *The Melrose Mirror* is the longest running of the publications. The *The Melrose Mirror* is published by a group of seniors as an activity in their senior citizen community center.

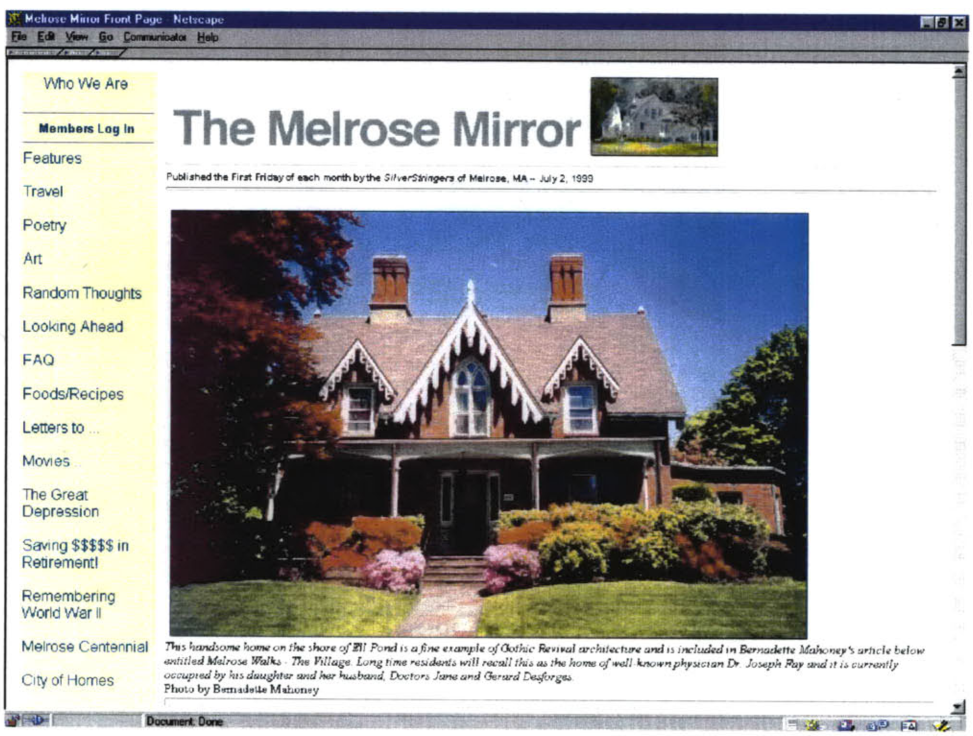


Figure 2-9: *The Melrose Mirror* is published using Silver Stringer's Pluto system.

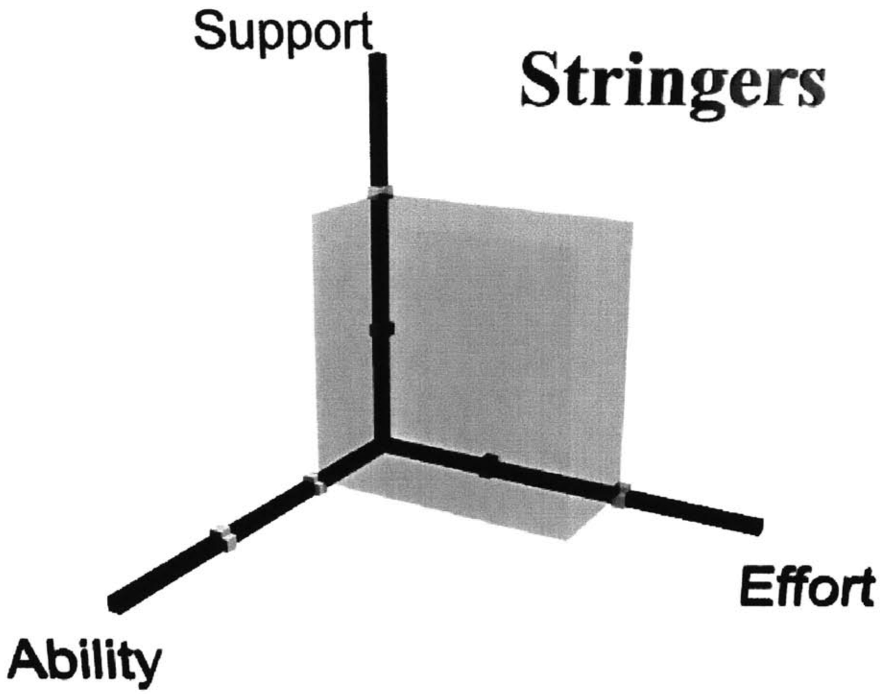


Figure 2-10: Silver Stringers is an example of a system that supports communities through spot intervention.

Ability

Reading Silver Stringers publications requires no skills other than using a web browser. Contributing articles is made easy by the Pluto software package. The system is intended for novice users. It guides the users step-by-step through the writing, editing, and publishing stages.

Extending a Silver Stringer publication is simplified by basic administrative commands in Pluto. It allows the creation of new sections and modification of stylistic properties. Although the package attempts to simplify the tasks, it does require an understanding of how the components interact with one another.

Creating new Silver Stringers sites is a straightforward task. It requires basic computer administration skills such as the ability to set up and configure a dedicated web server.

Support

Pluto is designed to require little external intervention. Like *FishWrap*, its server function is designed to run unattended for long periods of time.

There is little start-up work necessary to set up and configure a Pluto server by an external provider. The most daunting task for communities that have attempted to be self-supporting with Pluto is to find an Internet service provider that allows a Pluto server to be configured on their network. Most consumer networks offer ready support for clients but make it difficult for their customers to run their own servers.

Spot intervention is required in training the users on how to use the system and run a publication. However, being part of a weekly community activity facilitates this training. The majority of the support offered to the users of Pluto is in regard to issues of journalism rather than computing. Questions concerning writing and editing occupy most of the face-to-face and electronic discussions between the users and the system provider.

Effort

The seniors who participate in the Silver Stringers programs have a genuine desire to share stories. They also have both the desire and time to learn a new form of publishing. By making the Silver Stringers project a community activity, it is possible to mask some of the complexity of the system. Not every user need know how to accomplish every task associated with the publishing process. Some participants contribute without ever using the computer. Over time, the seniors have come to control all aspects of their online publication. Some are even learning to program so that they can directly impact the Pluto feature-set.

2.3.5 Multi User System In Community

Alan Shaw's Multi User System In Community (MUSIC)[Sha95] is a system designed for strengthening community ties through a custom-made bulletin board system that reflects various elements of a neighborhood. In the case of the original MUSIC site, in Dorchester's Four Corners neighborhood, computers were donated and distributed to various parts of the neighborhood. The willingness of a family to be a neighborhood "captain" and to share the resource with others helped determine which homes would have computers. Captains were trained to use the computer and the MUSIC bulletin board system. The MUSIC bulletin board system was used to discuss neighborhood issues. Community "facilitators" navigated the neighborhood to help make connections to other elements of the community system. The result was a mechanism that provided a sustained dialogue within the community.

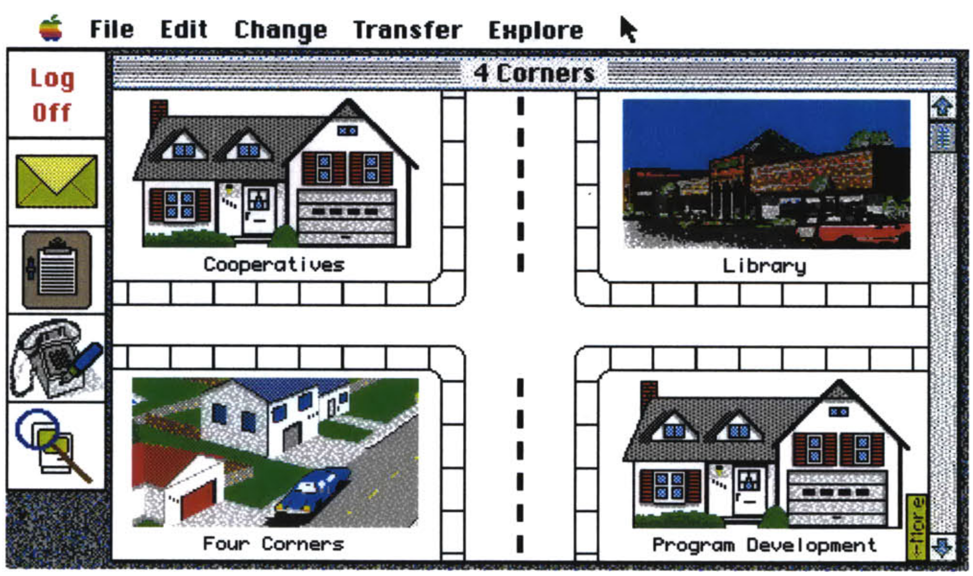


Figure 2-11: An example of the Multi User System In Community's (MUSIC) Four Corners graphical user interface.

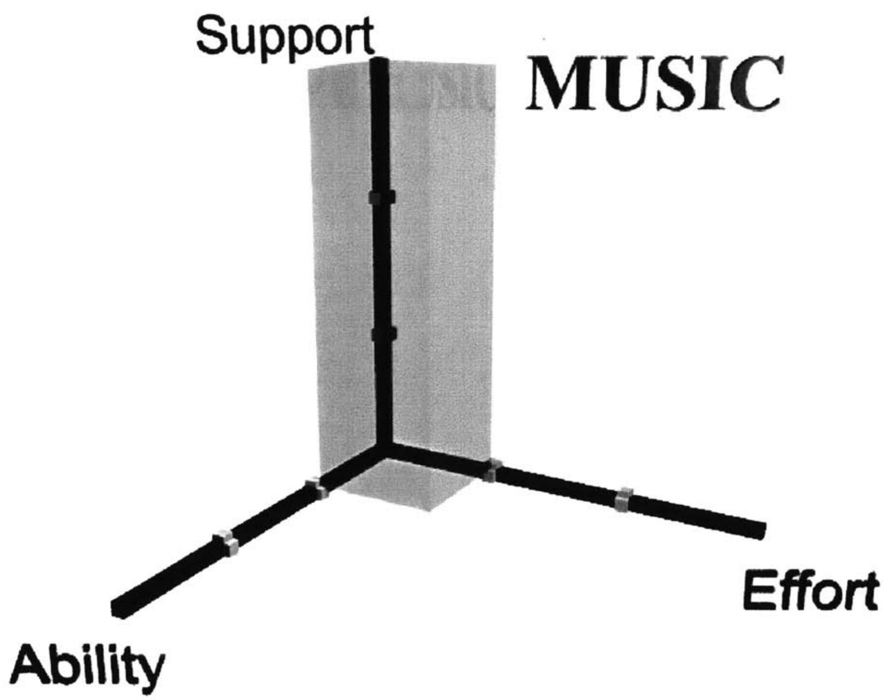


Figure 2-12: Multi User System In Community (MUSIC) requires extensive outside intervention, but little skills from within the community it serves.

Ability

Fundamental computer skills are required to operate the computers and use the bulletin board systems. These skills are typically not present in the target MUSIC communities and were taught to a seed group of neighborhood "captains" within the community. This seed group was then responsible for getting others within the community to participate.

Extending the system required specialized skills outside the scope of the target community. However, as in the case of chats, community members were able to create new discussion groups.

Creating new communities is extremely challenging since each system needs to be specifically designed for the target community.

Support

A significant amount of formal external support is required to make MUSIC work. It is labor intensive, making it a costly endeavor to sustain. A full-time facilitator is on-site to teach the community how to use the system and to encourage its use.

Effort

It requires a significant willingness on the part of the participants to learn new skills to operate the computers and navigate the bulletin-board system. However, it does help members of the community communicate with one another in an orderly manner, creating dialogue where it did not exist before. It also provides a unique means to learn about and create community resources.

2.3.6 Wearable Computing Community

Figure 2-13: Wearable Computing Community. (Photo copyright Sam Ogden)

The Wearable Computing community [Rho97] [SMRL97] [Sta96] at the MIT Media Laboratory is formed around a group's desire to explore opportunities offered by computers that are carried with the user at all times. The group is composed of students from many disciplines at the lab. Each uses his or her own specialty to extend this technology. For example, one student is focused on the application of vision technology to wearable computing. Another student is exploring the opportunities of an autonomous agent that retrieves documents as needed. The community formed itself to develop a common platform for all to use and to share their experiences with one another. They share information on integrating the latest display technologies and wearable computer peripherals. Unlike the personal computing user groups of the 1980s, this group does not have an external support group to help them. They are truly technology pioneers.

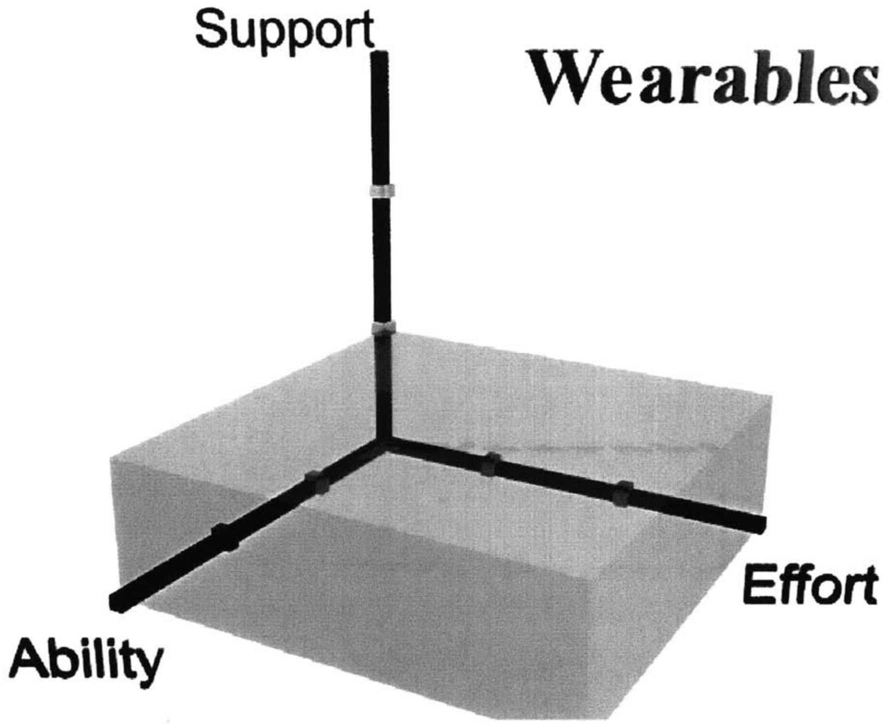


Figure 2-14: Wearable Computing Community at the Media Laboratory is a self-supporting community. Advance computer skills are required by the participants in order to participate.

Ability

Participation in the wearable community requires a significant set of sophisticated skills, both in computer hardware and software. Wearable computer users build their own computer systems and packaging. They also write the necessary software components to make their specialized devices work. Wearable computer users utilize their own specialties to extend the community's technologies.

Support

Being a pioneering community, the wearable community does not have the opportunity to leverage a formal external support system. They are completely self-supporting and have established a strong sense of community by sharing their discoveries with one another.

Effort

Significant willingness is required to overcome the technological complexities to participate in the wearable computing community. However, since there does not exist any other way of satisfying the desire of carrying one's computational equipment, the participants are willing to make the necessary sacrifices to participate. Not only are diverse skills required to participate in exploring the technology, but there is also a "gawk" factor of people startled by this novel use of computers.

2.3.7 Summary

Experiences of six communities using different technologies have been characterized using the ASE Framework. The table below highlights the degree to which ability, support, and effort contribute to the successful adoption of technology by each of these communities. The abilities attribute is perhaps best exemplified by the range exhibited in the Chat and Moo examples. The effort attribute is best exhibited in the Wearables example. The support attribute is a significant factor in all of the examples. When applied to these systems, the framework is useful in assessing their sustainability.

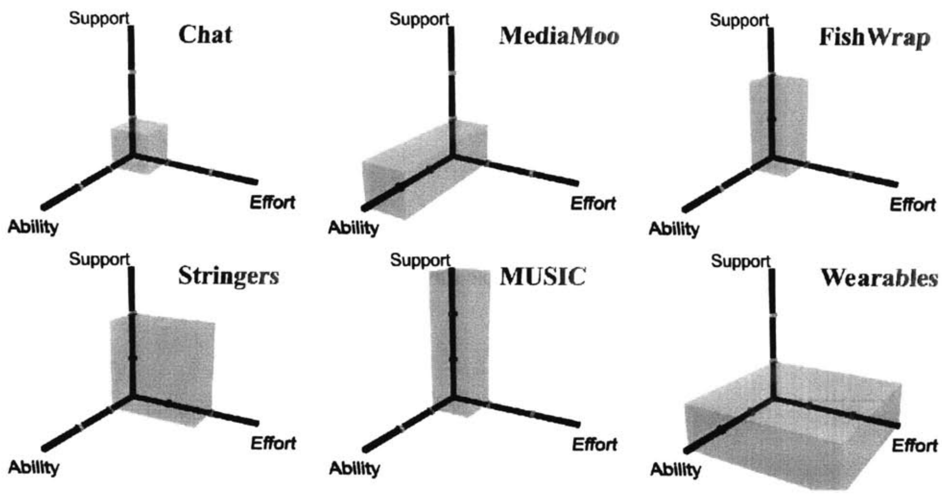


Figure 2-15: The ASE Framework applied to existing community-centered technologies.

Technology	*Ability*	*Support*	*Effort*
Chat (IRC)	Minimal skills	Self-supporting	⋆
MOOs	Basic programming	Self-supporting	⋆
FishWrap	Minimal skills	Spot intervention	⋆
Silver Stringers	Minimal skills	Spot intervention	⋆ ⋆
MUSIC	Minimal skills	Full-time support	⋆
Wearables	Advanced programming	Self-supporting	⋆ ⋆ ⋆

Chat (IRC) is an example of a system that enables a community to sustain itself over a long period with little or no outside intervention. This is evident by the sustained use of IRC by United We Stand as a means of maintaining a connection between its members. Many other online communities also use IRC and it continues to grow in popularity as a means for communities to stay connected. IRC occupies a small volume near the origin in the graphic representation of the ASE Framework. This small volume, representing minimal dependence upon community skills, support, and expertise, is a predictor of successful adoption of IRC by many communities.

The MOOs described earlier in this chapter are still in use. Support for Media Moo and MOOSE Crossing has moved, along with Amy Bruckman, from MIT to Georgia Institute of Technology. The ASE Framework illustrates some of the potential and shortcomings of the MOO design: (1) It is unlikely the children in MOOSE Crossing would have started a MOO on their own. It requires an level of expertise that is beyond the capabilities of the target group. Also, the community was created around the technology–the children did not know each other prior to joining the project; (2) However, once launched, children are willing participants. Within MOOs, they have the ability to construct new objects and they enjoy showing these objects off to others in the community; and (3) Running a MOO requires a sustained commitment that children are not in the position to make. External support is needed to keep MOOSE Crossings running for the benefit of the children it serves. The MOO examples reveal some of the complexities of the ASE Framework's attributes, particularly the ability attribute. The graphical form may be misleading in presenting the multiple abilities needed to participate as a continuum. While participating and extending a MOO are a natural progression for the typical user, these activities are discrete from creating new communities using the technology. In the case of MOOSE Crossings, spot intervention was needed to establish the community, both to administer the technology and to attract members to the community.

Although *FishWrap* technology is still running at MIT, it does not have the same level of community participation that it had when it was first launched. Once the developers graduated, the publication was left with no one to promote it. Had the developers used the ASE Framework, they might have formulated a plan to transfer the operations to an another group to provide long-term support (promoting *FishWrap's*, soliciting community participation, and providing spot-support).

Silver Stringers is an ongoing research project at the Media Laboratory. Although there are a dozen communities using the system, the designers have not yet developed a mechanism for a graceful transition between formal external support to a self-supporting system. Only when such a plan is in place will it be possible for the number of Silver Stringers groups to expand from the current handful of communities to a scale enjoyed by technologies such as IRC. The ASE Framework illustrates this attribute of the technology adoption well. The framework does not account for social forces that encouraged the community members to participate. Recent societal acceptance of computers has played an important role in motivating participation.

MUSIC has also been tested with a small number of communities. Although the system itself is simple to use, the surrounding reliance on formal external support make it difficult to sustain after the initial funding period. Given the necessity for formal support, the ASE Framework predicts that a community would have difficulty sustaining itself unless the designers of MUSIC develop a plan for the community to take control of the system past the formal intervention period.

The Wearables Community at the Media Laboratory is a small one and unfortunately getting smaller as the participating students graduate. This is in part because the community was formed around an emerging technology that was not well understood. When new, it engaged its members both as individuals and as a group in learning about its potential. Once it became a relatively mainstream technology, its appeal waned and the community dispersed. This suggests that technology cannot help keep a community together when it loses its raison d'etre.

Chapter 3

Canard's Experimental Platform

3.1 Motivation

The following hypothetical scenario helps illustrate the need for developing an experimental platform that unifies different communications streams.

> *Beep, beep, beep. I get a page from my wife. "Good news, the doctor says it's twins! Call me at the clinic." I pull the cell phone out of my pocket, prepare to dial, and suddenly realize I don't have the phone number, nor, in my panic, can I remember the name of the clinic. But I do remember that the phone number is in the Rolodex^{TM} on my desktop computer. If only I could figure out a way to transfer it from there to my phone. Or maybe my pager could have transferred it for me!*

As telecommunications systems increasingly become digital, opportunities arise for integrating phones, pagers, personal data assistants, answering machines, and other devices, into the personal computer environment. One aspect of this integration is the converge of multi-modal data streams into a single digital channel. Voice and data need no longer be segregated to different physical channels. While this convergence is leading to more efficient use of channel capacity, it has only an indirect impact on consumer services. One potential of integration that has direct impact on consumer services is the commingling the meta-data that are associated with various data streams.

Meta data integration provides numerous opportunities for efficient communication management and the provision for new services. Some commercial products, such as Microsoft's Outlook^{TM}, leverage databases maintained by competing products in order to help the consumer navigate with a unified interface. For example, Outlook will consult Netscape's Communicator^{TM} address book for entries. These products help an individual manage their desktop, but they are limited to a single vendor's vision and they don't enable or encourage the construction of services by the non-developer community.

Over the past decade, the user modeling community has broadened its agenda from that of modeling users within the context of a single application to examining the users and communities of users within the context of all their activities[Orw93][Orw96]. Specifically, through the monitoring of the communication streams and individual and community databases, user modeling techniques can help optimize communication within a community. One feature of these techniques is their ability to represent to the community relevant information about its communication activities. These representations bring a level of transparency to community communications that affords the opportunity for members to leverage telecommunications for personal and community benefit.

Central to this thesis is the hypothesis that open systems allow communities to adapt disparate telecommunications solutions to their needs. Such systems allow community members to create applications that benefit the community as a whole.

Most off-the-shelf communication devices available today are designed as closed systems, which preclude easy integration into user modeling systems. Cross-channel integration of telecommunication devices (e.g., telephone, fax, pager, etc.) is usually provided at a system provider layer. This is the case with Bellcore's approach to unified messaging[Sou98a] [Sou98b] [RT98] Ericsson's iPulse^{TM}[ipu99], and ATT's OneMail^{TM}[one96]. Such an approach to unified messaging is intended to offer complete solutions for the benefit of service providers. It is not intended that individuals create novel services. Industry focus has been to develop large networks that serve "everyone," not cater to the specific needs of an individual community. In general, these systems provide basic rule-based filtering to route messages from one communication device to another, but not much more. Example of such system include IBM's Alter Egos[alt99], ATT's OneMail^{TM} and PersonalLink^{TM}. Commercial software libraries, such as MAPI (an asynchronous messaging application programmer's interface (API)) and TAPI (a synchronous telephone API) [Amu96], suffer in that they are device- and vendor-specific.

Some recently designed telecommunication servers were built for controlling specific devices in a synchronous mode (e.g., a speech-based interface over the telephone) [Aro92][Sch93] and as gateways to specific Internet protocols (e.g., electronic mail). Extensibility by individuals was not a design objective. From the outside, these systems seem monolithic. Other open-source messaging systems look to provide improved Internet-based instant messaging (i.e., iFlame extends Zephyr by making MIME a primitive data type[LaM96]) but do not consider over-the-air transport. At the time this research was conducted, there was no off-the-shelf solution that could be used as an experimental platform for a constructionist framework.

3.2 Communication Model

The design goals for the experimental platform included a modular model which could be easily expanded. It was also important to use a model that was easy to comprehend by developers and users alike. The group of developers working on the project had diverse programming experiences as did the intended users. Some developers on the project were just learning how to program while others were already experienced programmers. The programming environments that were used were likewise diverse. Some programmers preferred working with Perl, others with C. Since the programming environment was modular, system components could be developed independently.

The supported operating system environments of the project included Microsoft Windows 95^{TM}, NT^{TM}, and various UNIX^{TM} derivatives. The main Canard message servers were running in Apple's OPENSTEP^{TM} environment. The main consideration for this choice of operating system was the availability of a Mach threading library that enables shared databases to be simultaneously accessed efficiently.

The Canard communication model was inspired by Jef Poskanzer's Extended Portable Bitmap Toolkit[Pos91] (PBM). In the PBM library, images in different file formats (e.g., GIF, TIFF, TARGA, etc.) are first converted to an intermediate representation. Successive

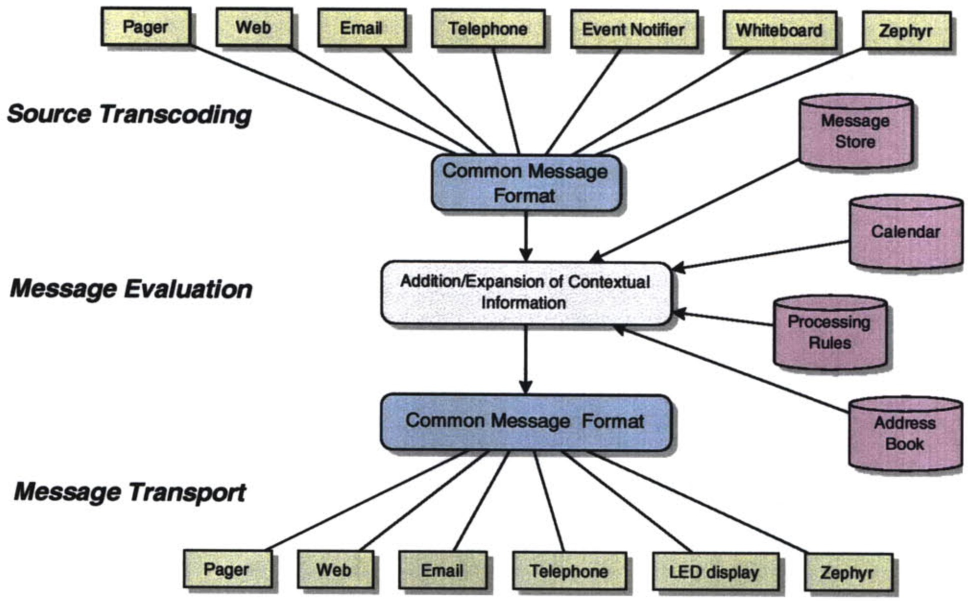


Figure 3-1: The Canard messaging model has three layers: representation, evaluation, and transport.

image transformations are performed (e.g., gamma correction, rotation, scaling, etc.) on the image in its intermediate representation. Finally, the transformed image is converted to its final file format. This allows developers to write image transformation functions that only need to interface to the common representation, rather than all possible file formats. New image formats can be incorporated by writing file converters to encode to, and decode from, the source representation.

The Canard system applies a similar philosophy to messaging: new formats are adapted by writing conversion functions which convert from a source representation to a common message format. Once in the common format, a message's importance is evaluated against personal databases. At delivery time, the message is converted into a format suitable for delivery through a specified channel. This allows for many developers to work on the project at the same time in an orderly, modular fashion.

Figure 3-2 is used as an example to illustrate how messages are handled by the Canard system. The example is drawn from an early implementation of Canard, when the system used a two-way pager with an extremely limited user interface (four buttons!). It illustrates how, by using a sophisticated server in concert with a device with limited capabilities, a user can have rich and rewarding interactions by maximizing the use of available contextual databases.

3.2.1 Source Layer

Messages entering the Canard system come in different formats. The goal is to extract the salient features of the message while preserving the richness of the original format. In the case of electronic mail format, the sender information, message subject, and body are extracted while the original headers are stored for later use. Similar features can be extracted from other streams (e.g., caller identification (caller-ID) from incoming phone

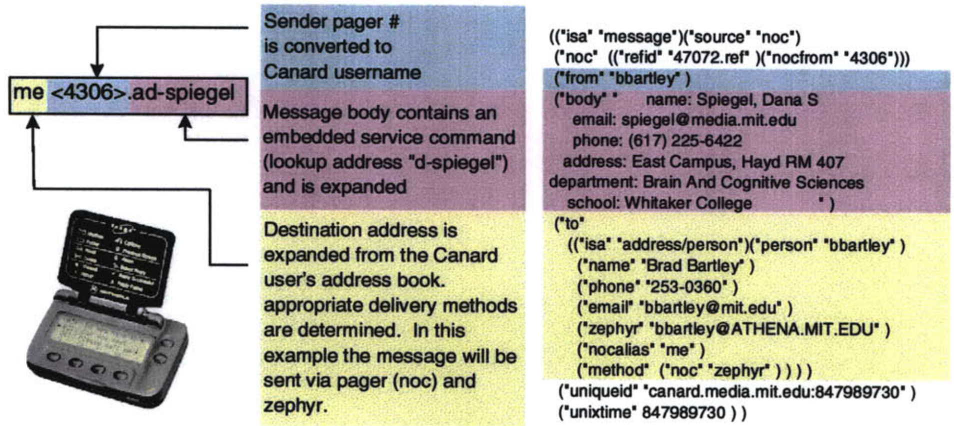


Figure 3-2: In this example, a message from a Motorola Tango two-way pager (on the left) is converted into a Canard message object (on the right). Salient features from the original message are preserved. Delivery mechanisms are computed by consulting personal databases (in this case, the address book).

calls.) Preserving all the original features of the message in the intermediate representation allows developers writing evaluation modules to utilize those features without having to go back to the original source conversion software. An additional benefit of this strategy is that all message data is made available in a familiar format (The original author of the converter may have written it in a programming language unfamiliar to the developer making modifications to the evaluation layer). Indeed, this proved to be the case more than once during the development of Canard. In a commercial environment, this has the benefit of allowing module providers to keep the specifics of their implementation hidden from those writing evaluators.

Figure 3-2 is an example of a message originating from a two-way pager. The message received by the Canard system is "me <4306> .ad-spiegel", which, once in the system, is expanded into a uniform representation called a message object. The first feature of the message is "me", which is the destination of the message as typed in by the user. The second feature is "4306", which is the pager identification number (PIN) provided by the two-way paging system. The third feature, ".ad-spiegel", is the message body as composed by the user. The PIN is used to look up the Canard user name of the owner of the pager. Knowing which user name to use determines which personal databases to use (or in some cases which remote server to consult). It has the added benefit that the recipient of the message will be presented with a single user name rather than an unfamiliar PIN. (One goal of the system is to reduce the number of communication addresses an individual needs to remember.) The destination address provided by the user is looked up in the user's personal address book. In this example, "me" is an alias for the owner. Prior to message evaluation, the message object for this message looks like:

```
(("isa" "message")
 ("source" "noc")
 ("noc" (("nocfrom" "4306")))
```

```
("from" "bbartley")
("to" "bbartley")
("body" ".ad-spiegel"))
```

The message object representation is based on Dtypes (explained in greater detail in Section 3.3.1. Each line contains a keyword/value pair. The "isa" keyword in the first line determines the object type. In Canard, the most common objects used are "message" and "address". The "source" keyword in the second line refers to the source communication channel of the object. In this case, "noc" refers to the Network Operations Center, a fancy name for the two-way paging system. The keyword/value pair in the third line preserves all the information that was extracted from the original representation. The most important information is the PIN, which is located in the "nocfrom" environment (Note that a value can contain an embedded keyword/value pair). The fourth and fifth lines contain the sender and destination Canard user names. The mode of delivery will be decided later.

3.2.2 Message Evaluation Layer

Once a message is in a uniform representation, it can be processed by one or more filters. These filters are much like the PROCMAIL[Ber94] electronic mail filtering package, except rather than solely operating on electronic mail messages, they can be programmed to evaluate any field in the message object, and they can access personal databases external to the message. This evaluation determines which delivery mechanisms to use.

In the case of the example presented in figure 3-2, the message originated from the paging system. Consequently, a paging system-specific filter is invoked. The filter examines the body of the message and determines that it starts with a '.', indicating an embedded service request. Embedded services are structured messages [MGL+88] that are handled by external processes. In this case, ".a" indicates an address lookup. Following the command are arguments to be passed to the address lookup program. In this case, the name "d-spiegel" is looked up in the MIT online directory and is used to fill the body of the message. The program to do the directory lookup was written independently from the filter program.

The embedded service model is particularly useful for consulting databases that exist outside of the system. In addition to the MIT on-line directory address lookup, a direction assistance service is offered, which gives directions from one street address to another. In a personalized system like Canard, it can be assumed that the starting address may be omitted as it can be inferred as the recipient's last known location. The location can be extracted from communication meta-data or calendar.

In the above example, the message is addressed to the sender since the author needs the results of the directory search. In this example, the system determines the delivery method to be the same one used to send the message, which is the two-way paging system. The results could have easily been directed elsewhere by addressing the message to another person.

3.2.3 Message Transport Layer

At this stage, it is determined which message elements are to be delivered and in what form they are to be sent. For example, a phone message has a number of elements, including caller-ID and directory information, a sound file, and a description of the sound file (the length and time recorded). When delivering this message to a text pager, the sound file is omitted. A text message is delivered that contains a transcription of the meta-data that characterizes the sound, e.g., "Dana left a 36 second voice message at 2:46PM."

In the example of Figure 3-2, a message was received from a pager and run through a filter that expanded the body. The resulting message was a text-only message that required no special processing for presenting on a pager. Had it been a MIME-encoded[FB96] message, then a program would have been executed to convert the message into a form suitable for a pager. The delivery mechanism is determined at the evaluation layer. As a final step, the message is queued for delivery by the pager system.

3.3 Software Implementation

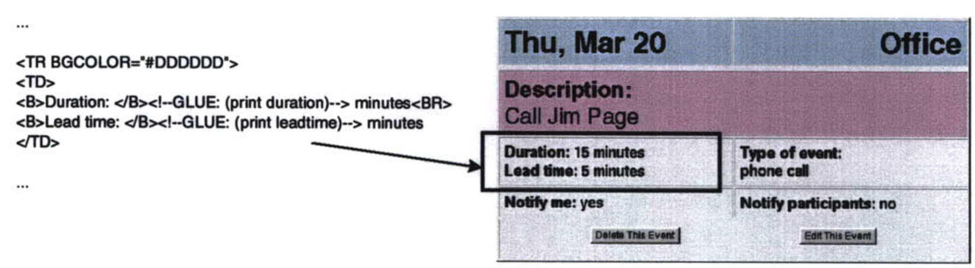


Figure 3-3: On the left side is an excerpt from the style sheet used to render calendar event objects. On the right side is the resulting WWW hypertext presentation. Dtype evaluator commands are embedded within "<!–GLUE: –>" mark–up tags. This allows for WWW browsers to ignore the embedded commands.

3.3.1 Dtypes

Dtypes [Abr92] define a standard at the Media Laboratory for the transmission and manipulation of complex data structures across computer networks. They are designed to make the task of developing complex, distributed applications easier. Dtypes accomplish this by providing an extremely flexible, platform- and programming language-independent representation for complex data structures such as integers, floating point numbers, strings, and lists. The Dtype standard has an external ASCII representation which allows it to be used in a variety of systems and networked environments. For this reason, Dtypes are widely used within the Media Laboratory. Dtypes are at the core of the FramerD [Haa96] knowledge representation system and have been used to build a number of distributed servers as well[Blo91][Abr93][LK96][Die95]. New Dtype implementations exist in C, C++, Scheme, LISP, Java, and Perl on a wide variety of platforms.

Dtypes are used extensively in the *FishWrap* personalized news system [CMS95] [BCE+96]. A modular system, with Dtypes as the interface, is used to link independently developed user-modeling, knowledge-representation, and database servers[She96]. By having a simple consistent way of transmitting complex data types over the network, module developers are free to concentrate on their respective tasks.

With Dtypes, structured database objects are easy to define. Simple database associations can be created, and fundamental manipulations can be performed. All Dtype implementations have an evaluator which allows platform–independent manipulations.

The following are the basic data types supported by Dtype[Abr92]:

DTYPE_NULL is a valueless data type which is often used for representing uninitialized values or for passing back error results.

DTYPE_INT is a 32 bit integer type.

DTYPE_REAL is an IEEE floating point type.

DTYPE_STRING is a zero-terminated string.

DTYPE_BANG is a Dtype which can best be described as an "impulse" signal - if, for example, you have a program which is waiting for any signal before it starts processing, a DTYPE_BANG is the acknowledgment message sent to it.

DTYPE_PACKET is a buffer of raw data which also has an associated header of keyword/value fields. An example of a packet is an image stored in binary form, with the header containing the dimensions of the image.

DTYPE_LIST is a linked list of Dtypes which may themselves be DTYPE_LISTs.

DTYPE_REF is a special Dtype which is used to "reference" lists. It is an easy means of picking out elements or portions of lists from other DTYPE_LISTs.

In the C programming language, a Dtype is represented as a simple structure, containing a type field, a data field, and a next field. The Dtype structure definition in C looks like:

```
typedef struct Dtype {
  long type;                  /* Data type */
  union {
    long i;
    float r;
    char *str;
    struct Dtype *list;
    struct OPacket *op;
    struct DtRef *dtr;
  } data;
  struct Dtype *next;
} Dtype;
```

The *type* field contains a value that indicates what type of data the Dtype contains. This value is one of the predefined macros (DTYPE_NULL, DTYPE_INT, DTYPE_REAL, etc.) representing the eight primitive data types. The *data* field is a union containing the actual value of the Dtype. The element of the union that should be used depends on what is in the type field. The *next* field is only used when the Dtype is part of a list, in which case the *next* field points to the next Dtype in the list.

The Dtype library provides functions for creating, accessing, and manipulating the data contained within the structure. The library has functions for transmitting Dtypes over computer networks. Means are also provided for storing and retrieving the data from files.

Manipulating Dtypes solely with the library functions can be a burden. Fortunately, the Dtype library provides a mechanism in which Dtypes can be represented as ASCII text strings and be parsed into Dtype structures. Although this facility is not as computationally efficient as a purely binary representation, it allows for rapid prototyping and reduces the time to debug programs by eliminating the complexity of creating Dtypes solely with C functions. Programming errors often occur when creating complex data structures.

The following C program segment reveals the complexities of creating a Dtype:

```
Dtype *newEvent=NULL;

newEvent = DtypeCreate(DTYPE_LIST, NULL);
dtypePush(newEvent,
    DtypeCreate(DTYPE_LIST,
    DtypeListCreate(DTYPE_STRING, "event_description",
    DTYPE_STRING, "Global Event", NULL)));
```

it can be replaced by a less complex line:

```
Dtype *newEvent=NULL;

newEvent=DtypeParseStr( "((\"event_description\" \"Global Event\"))");
```

The ASCII representation simplifies what the programmer sees and, as a result, novice programmers feel more comfortable lifting such examples from working programs to include in their own. The simpler the example, the less daunting the task seems. Dtypes enjoyed great success and longetevity at the Media Laboratory because they simplified the development process for many projects.

Dtype Lists can be used to store simple databases as lists of keyword and value pairs. A Dtype keyword and value combination is called a "binding." A collection of bindings is called an "environment." The value of a particular environment can be found with the Dtype library's DtypeEnvLookup function.

The Dtype library also provides a basic SCHEME-like[AS96] evaluator that allows fundamental operations to be performed by taking a Dtype list and treating the first item like a function name followed by a list of arguments:

```
("+" 3 4 5)
```

The evaluator looks up the function name in a table that maps the name to a function in the C language. This map in turn calls the C function with the argument list. The C function performs the necessary operation and returns a Dtype as the result of the evaluation. The Dtype library provides a basic set of functions which the Dtype programmer can easily add to. This ability to extend the functionality of the Dtype evaluator is an important one for application developers.

The evaluator provides the following functionality:

- Dtype list and environment manipulation
- Looping and branching functions (if, while, for, switch)

- Arithmetic functions
- Boolean logic functions
- Comparison function (==, !=, <, >, <=, >=)

The Dtype library provides a client/server facility that makes the creation of network-based applications rather simple. Dtypes are passed as lists where the first item is a command for the remote server to execute. The dispatching functions are then invoked with the rest of the Dtype as the variables. The Dtype manual[Abr92] describes an example server that maintains a database about people. This example provides a foundation for user-modeling developers.

For personalized news projects, a server protocol, called "Betty"[Blo91], was developed. The server is based on the Dtype server model with a few basic commands: "fetch", "put", and "get". The "fetch" command allows specified articles to be retrieved from the database. "Put" allows for storage of articles in the database and "get" allows particular environments (e.g., "headline", "body", etc.) to be retrieved. Articles to be processed are specified by unique identifiers (uniqueid) or evaluated (e.g., (= headline ".*Kennedy.*")). The use of regular expressions is supported by the server.

Storing Dtypes can be challenging since, in practice, they are used in ways that traditional database servers do not handle well. In a development environment such as the Media Laboratory, it is not always clear what features (or fields) of a database record are of interest. Fields are added and eliminated as a project progresses and the size of the records changes. To address this problem, a database system called Mini-Cell Database (MCDB) [She94a] was implemented. It is built upon the GNU Database Manager [Nel93]. MCDB allows storing of Dtypes and indexing of arbitrary Dtype environments. In most cases, it is string and integer values that are indexed (e.g., creation time of a Dtype based in seconds since 1970). This allows for easy retrieval of Dtypes that have been stored locally in a file or remotely on a network.

3.3.2 *FishWrap* Glue Utilities

The Canard system is built upon much of the infrastructure that was developed for the *FishWrap*[CMS95] personalized news server. *FishWrap* uses Dtypes extensively as a means of storing news wire articles.

FishWrap articles come from many sources and channels. By using Dtypes, a uniform representation of common article features (e.g., author, headlines, dateline, etc.) is maintained, while preserving source specific coding (e.g., story tags). Once an article is in the uniform representation, it is amenable to analysis and augmentation.

A library was written to facilitate the retrieval of news articles regardless of the method used to store them. A unique identification for an article was developed that consisted of a server "tag" and a unique name for the article. For example, fishwrap:ap39849343498 represents the article with the name "ap39849343498" on the server "fishwrap". The "fishwrap" server would be described in a database as:

```
( "fishwrap"
  ( ("protocol"  "betty" )
    ("host"      "fishwrap.mit.edu")
```

```
("port"       "bettyserver")
("language"   "ENGL") ) )
```

The Glue library provides a function to retrieve an article using the server tag and article identifier:

```
Dtype *glGetArticleFromServer(char *tag, char *uid);
```

Placing an article in Dtype representation into a server is made equally as easy with the function:

```
Dtype *glPutArticleInServer(char *tag, Dtype *dt);
```

In order to keep *FishWrap* article presentation current with emerging WWW standards, a parser was developed that allows an article Dtype to be applied against a contextual Dtype (i.e., a style sheet). This facility allows information about the presentation of articles to be encoded outside the core *FishWrap* software distribution. For example, a headline can be extracted from an article object and rendered in a style customized for each reader.

3.3.3 Canard Objects

Canard "objects" are Dtypes used to represent fundamental messaging data types. Messages, address books, and calendar events are all represented with Canard objects. These Dtypes have an "isa" environment that allow the Canard system to identify the messaging data type they represent.

Emulating an approach taken in another Media Laboratory project, The Media Bank, all Canard objects have associated online documentation in HTML format with additional embedded mark-up tags. The *FishWrap* parser is used to validate Canard database object with text from the documentation file. The documentation specifies what environments are required, and validation procedure verifies that the environment is present. If the required environments are not present the procedure logs an error so that the author can correct it. This procedure allows for quick detection of programming errors due to misinterpretation of documentation in a live environment. The example below illustrates a segment of the Canard documentation with instructions for verifying an object compliance. Dtype evaluator commands in the example below are enclosed within "<!–GLUE: –>" separators:

```
<li><b>person</b> is a required environment
<!--Check to see if there is one -->
<!--GLUE: (if ("==" PERSON NULL)
              (set invalid 1))) -->
```

The parser is also used for rendering data according to a "style sheet" (see Figure 3-3). The Canard system adapts to the viewing context by selecting an appropriate style sheet (pagers may offer a different view of an object than a World Wide Web (WWW) page).

As seen in the following examples, there are a number of special environments defined in a Canard object:

source the name of the application that created this object.

isa the type of object(which will allow for object validation to be performed).

owner the e–mail address of the person who created the object (This allows us to notify the person when an object fails to be validated).

unixtime the time in which the object was created expressed in seconds since 00:00:00 GMT, January 1, 1970.

uniqueid a unique string identifier for the object.

A number of foundation objects within the communication model are implemented using Dtypes. Each of these types may have a subtype as well.

Sensor This object keeps track of observations made about an individual. Sensors provide a mechanism for the system to learn about an individual's activities by observing their communication devices. For example, Canard can record when a two–way pager was used to originate a message and infer from those data that a person is away from their office. In the long run, it is expected that wearable devices will provide observations about a user's situation, a potentially rich source of data for the Canard sensor objects.

Message Message objects provide a uniform representation so that evaluation modules can be developed independently. A message object typically has "to", "from", "subject", and "body" associations that are used to determine how the message is subsequently processed and delivered.

Address An address object contains the information needed to deliver a message. This information may be as simple as a string that represents which server to consult. The string is later expanded as a Dtype containing all the information needed by a transport mechanism to deliver the message. Address objects currently represent either a person, a group, or a service.

Timer This structure encodes temporal attributes. In the case of the message object, it is used to defer evaluation and/or delivery of a message.

Event The event structure [GS88] is used to communicate the occurrence of specific activities. An event may result in the automatic creation of a message object at a given time. Reminders and alerts [Sil97] are a common events used by Canard communities.

3.3.4 Any Tools

Canard data objects provide a convenient way for programmers to either store complex databases on local file systems or transmit them over the network. They are easy to distinguish from one another through application programming interfaces.

A mechanism was needed for non-programmers to create and manipulate Canard objects and databases. A suite of programs, called *AnyTools*, that can manipulate Canard objects and databases was developed to meet this need. The AnyTools suite utilizes the World Wide Web browser as the interface to Canard objects since it provides consistency across many platforms. Anytools consists of *AnyView, AnyDelete, AnyEdit,* and *AnyArchive*, which are implemented as Common Gateway Interface (CGI) programs allowing users to interact with them using any web browser.

AnyView is used to render Canard objects. It performs searches on Canard databases and displays the results of these queries. *AnyDelete* is used to delete objects from Canard databases. *AnyEdit* provides an interface for editing Canard objects. *AnyArchive* moves object into an archive database for permanent storage.

AnyAccl is a programmatic interface for checking user permissions for viewing and editing personal and shared objects. It is provided as a library for use by the other AnyTools programs and is not directly accessible by users.

3.4 System Databases and Servers

In a complex system like Canard, it is important facilitate the creation and access of local and remote databases. Database access is based on a server abstraction [She94b]. A system map file, called "server.dtype", describes where databases are located and how to access them.

An example of a server.dtype file:

```
(
# System wide servers
 ( "refid"   ( ( "protocol"  "betty")
      ( "host"       "canard.media.mit.edu")
      ("port"        13131)
      ( "language"   "ENGL") ) )
 ( "observer"  ( ( "protocol"  "betty")
      ( "host"       "canard.media.mit.edu")
      ("port"        11112)
      ( "language"   "ENGL") ) )
# System Databases
 ("accl" (("protocol" "mcdb")
 ("host" "/canard/accl/accl.mcdb")))
 ( "system_timer"  ( ( "protocol"  "mcdb")
     ( "host"       "/canard/timer/system_timer.mcdb")
     ( "language"  "ENGL") ) )
 ( "calendar_state" ( ( "protocol" "mcdb" )
      ( "host" "/canard/calendar/calendar_state.mcdb" )
      ( "language" "ENGL" ) ) )
# Personal Databases  -- Their location are determined through name2path
 ( "pcal" ( ( "protocol"  "mcdb")
    ( "host"       "pcal.mcdb")
    ( "language"  "ENGL") ) )
 ( "addresses" ( ( "protocol"  "mcdb")
 ( "host"      "addresses.mcdb")
 ( "language"  "ENGL") ) )
 ( "timer"  ( ( "protocol"  "mcdb")
      ( "host"       "timer.mcdb")
      ( "language"  "ENGL") ) )
 ( "messages" ( ( "protocol"  "mcdb")
( "host"       "messages.mcdb")
( "language"  "ENGL") ) )
 ( "archive" ( ( "protocol"  "mcdb")
```

```
( "host"        "archive.mcdb")
( "language"   "ENGL") ) )
# Web Servers
 ( "web"   ( ( "protocol" "www" ) ))
# Interfaces to System Resources
 ( "ledsign-2"      ( ( "protocol"   "betty" )
                  ( "host"         "canard.media.mit.edu" )
                  ( "port"         12332 )
                  ( "device"    "ledsign2")
                  ( "language"   "ENGL") ) )
 ("whiteboard"   (("protocol"   "mcdb")
                   ("host"   "/canard/whiteboard/whiteboard.mcdb")))
)
```

Server.dtype is a Dtype with a collection of Dtypes that describes how to access the databases. For example, the "whiteboard" database is accessed using the MCDB software library. The host environment describes how the file is stored on the local host. In the whiteboard example, a full file path is given to specify the database location.

The "ledsign" database is an example of a remote database. In these cases, the host environment contains the name of the host that is running the server. The protocol field specifies how to connect to the server.

Personal databases (e.g., messages, addresses, calendar, etc.) are described in a map file as names without a path. A full file system pathname can be evaluated with the "name2path" function.

The Server.dtype scheme allows programmers to specify a server or database with a "server tag" (e.g., "fishwrap") and to not have to worry about the specifics of how to access it. The protocol or actual host can be readily changed. This allows system administrators to move the servers and databases around without fearing that a subsystem will break.

Retrieving data is made simple through the use of *FishWrap's* Glue library. This library implements two common functions, put and fetch, using server tags. Items such as news articles and messages are represented as Dtypes and the actual translation to a database is handled by the Glue library.

3.4.1 Observer Server

The Observer server is a database of observations made by Canard for the benefit of the community. Of particular interest is tracking the location of community members as a means of enhancing community awareness. To achieve this, every message sent through the Canard system triggers an observation. The observation is a Dtype containing the user's name and the source of the observation (e.g., a message from a pager). This information is rendered as a list of people in the community who have been "seen" recently by the Canard system. The inference is made that a recently observed person can be easily reached with the system. Note that the system does not record precise activity. It is equivalent to a building entrance log that records when someone has signed in or out.

3.4.2 Timer

Most messages received by the Canard system are processed immediately. But the system has the ability to store messages temporarily. This feature enables the transmission of a message to be deferred at the request of the message author, as is the case with reminders [Sil97]. Messages are also deferred because the recipient is unavailable until a particular time, which Canard infers from the calendar database. If a message has a deferred delivery time, it is stored either in a long-term storage database that is processed once a day or, if it is to be delivered later in the day, in the short-term timer queue. The timer queue wakes up every minute and delivers messages at that time.

3.4.3 Access Control

It is important that some degree of access control be provided by Canard so that community and personal databases can be shared. An access control mechanism was implemented in which programs must first consult before accessing the database. The access control list is implemented as a Dtype:

```
(
# The Access Control List is also public
  (("db" "accl" )
    ("read" "all" )
    ("uniqueid" "canard.media.mit.edu:911063330" )
    ("unixtime" 911063330 ) )
# Whiteboard are available to every to look at
  (("db" "whiteboard" )
    ("read" "all" )
    ("uniqueid" "canard.media.mit.edu:909174059" )
    ("unixtime" 909174059 ) )
# The Event list for the tetazoo group is public
  (("db" "pcal#tetazoo" )
    ("read" "all" )
    ("uniqueid" "canard.media.mit.edu:916938458" )
    ("unixtime" 916938458 ) )
# The address book is not
  (("db" "addresses#tetazoo" )
    ("read" "spiegel" )
    ("uniqueid" "canard.media.mit.edu:917299099" )
    ("unixtime" 917299099 ) ) )
```

3.5 Personal Databases

A directory is created for each user in the Canard system as a means of collecting in one place all databases able to be manipulated by users. There is one access function, "name2path", that translates the user name to a directory path on the host computer.

3.5.1 Address Book

The address book database is used for collecting contact information. Using a Canard object to represent each entry is an easy way to keep information about how to deliver messages. As new telecommunication devices are supported, new Dtype fields can be added. The fields can be simple strings or complex Dtype structures that contain all the necessary information needed by a particular communication channel. A default delivery mechanism is described in the "method" environment of the PERSON object.

Below is an example of an address book kept by the Canard system:

```
(
#first entry is the owner
  (("PERSON" "lacsap" )
    ("isa" "address/person" )
    ("name" "Pascal Chesnais" )
    ("phone" "253-0311" )
    ("email" "pascal@mit.edu" )
    ("nocalias" "me" )
    ("zephyr" "pascal@ATHENA.MIT.EDU" )
    ("skytel" "1913372" )
    ("method"
      ("noc" ) )
    ("uniqueid" "canard.media.mit.edu:882197348" )
    ("unixtime" 882197348 ) )
# an example service that displays who was last seen by Canard
  (("SERVICE" "who" )
    ("isa" "address/service" )
    ("method"
      ("web" ) )
    ("web"
      (("url" "http://canard.media.mit.edu/cgi-bin/aware.csh" ) ) )
    ("uniqueid" "canard.media.mit.edu:886694623" )
    ("unixtime" 886694623 ) )
# reverse telephone look up service
  (("SERVICE" "rtel" )
    ("isa" "address/service" )
    ("method"
      ("web" ) )
    ("web"
      (("url" "http://canard/cgi-bin/rtel.csh" ) ) )
    ("uniqueid" "canard.media.mit.edu:900166917" )
    ("unixtime" 900166917 ) ) )
```

3.5.2 Calendar and Events

A database for personal events is kept by Canard. Each event is contained in a Dtype with salient information. If the event is occurring the same day as it was entered, the Dtype is also stored in the system timer so it can send out an appropriate notification. Once per day a process is run that examines every personal database for events occurring later in the day. These events are placed in a central database that is used for delivering notifications during the day. This way, only one database needs to be managed.

```
(
# Events like chores, and meetings can be placed in a personal calendar...
  (("description" "Lunch with Pascal, Josh, and Dana" )
    ("typeofevent" "Lunch" )
    ("starttime" 916939200 )
    ("duration" 15 )
    ("leadtime" 0 )
    ("priority" 1 )
    ("notifyme" 0 )
    ("notifyparticipants" 1 )
    ("participants"
      ("tetazoo" ) )
    ("owner" "lacsap" )
    ("isa" "event" )
    ("location" "Food Trucks" )
    ("uniqueid" "lacsap:5802078917053" )
    ("to" "tetazoo" )
    ("queued" 1 ) )
    ("unixtime" 919873132 ) ) )
```

3.5.3 Messages

All messages processed by the Canard system are stored for each user in a database. Each message is represented by a Canard data object that has all the salient features of the message – sender, destination, message subject, and delivery channel. All the original information from the source transport is preserved as well. The message object gets updated as needed (e.g., when the message is delivered or read).

```
((("source" "rfc822" )
  ("isa" "message" )
  ("rfc822"
    (("Return-Path" "<postmaster@aleve.media.mit.edu>" )
      ("Received"
        ("Received-sub" "from aleve.media.mit.edu Tue, 13 Jul 99 15:41:59 -0400" ) )
      ("Date" "Tue, 13 Jul 1999 15:43:16 -0400" )
      ("Message-Id" "<199907131943.PAA11979@tinbergen.media.mit.edu>" )
      ("From" "Graystreak <wex@media.mit.edu>" )
      ("To" "pieper@mms.com" )
      ("Cc" "hackers@aleve.media.mit.edu" )
      ("Subject" "Re: hex math in TCL?" )
      ("Mime-Version" "1.0" )
      ("Content-Type" "text/plain; charset=US-ASCII" ) ) )
```

```
  ("subject" "Re: hex math in TCL?" )
  ("to" "lacsap" )
  ("from"
    (("PERSON" "tmp_wex@media.mit.edu" )
      ("method" ("email" ) )
      ("isa" "address/person" )
      ("email" "wex@media.mit.edu" ) ) )
  ("body" "Steve Pieper suggested...
...
--Alan
'' )
  (''content-type" "text/plain; charset=US-ASCII" )
  ("summary" "Steve Pieper suggested...
Is that what you want..." )
  ("uniqueid" "msg28572931894919@canard.media.mit.edu" )
  ("unixtime" 931894919 )
  ("received" "noc" ) ) )
```

3.5.4 Archive

In compliance with the experimental protocol negotiated with MIT's human subject review board, the Canard system only keeps messages in the users' main message databases for a limited time period. This protocol required the creation of two separate messaging systems for the users. In order to minimize the overhead associated with synchronization between the systems, Canard automatically deletes the messages in its main message database after three days. The archive is a means of keeping messages longer than the limit imposed by Canard's automated system. Messages are easily moved by the user through an interface in the AnyView program.

3.6 Communication Channels

During the course of the Canard project, a number of different communication channels were integrated into the system. All were off–the–shelf products or common Internet protocols. Their integration was straightforward. In order to keep the core server free of device-specific software, small servers were developed to handle the device-specific needs of each communication channel.

3.6.1 Settings for Communication

In a group environment, there are shared communication spaces. Using them for meaningful personal communication, without sacrificing privacy, is a challenge for the system designer. On a bulletin board, private messages may be left in a public space that reveal an indication of a message's importance (e.g., "call your mother") without fully disclosing its nature.

Group members use Canard in three distinct spatial settings:

Private communication takes place without the presence of other members (e.g., using a telephone within the privacy of one's dormitory room).

Semi–Public communication takes place where other group members may be present but not necessarily aware of the members communication activity (e.g., using a pager in the dormitory's lounge area).

Public communication occurs where anyone can be present and potentially disrupt the communication activity (e.g., a cellular telephone in the lobby of a building at MIT).

3.6.2 Physical Size

It is important to consider the physical scale of the device when designing a communications system. A diverse set of devices were implemented as part of the Canard system. Weiser describes three interface scales in his article on Ubiquitous Computing [Wei91], from which parallels can be drawn to communication devices used by Canard.

Tabs : a palm–sized display that can easily fit in one's pocket (e.g., a pager). This type of device may be limited in display capability but is one that is carried almost anywhere. It is not intended to be shared with others.

Pads : a letter–sized display (e.g., a computer monitor). This device has richer display capability but is not likely to be carried with the user. It can be shared with a small group.

Boards : a bulletin–board sized display (e.g., an electronic whiteboard with accompanying projection system). This type of device is intended for a rich information display and is not likely to be readily transportable. It can be shared with a large group.

Bill Mitchell adds "Billboard size" to the list in his book [Mit99]. This accounts for communication that occurs outside of buildings.

3.6.3 Two-way Paging System Interface

A Motorola ReFlex^{TM} two-way paging system [Cou98] is integrated into the Canard system. A personal computer is used to control the transmission and reception of data packets between the base station and the mobile units. The paging system installed at MIT has a coverage of about two miles distance from campus, which is essentially walking distance around campus. This makes it an appropriate system for providing connectivity for students, who often move around.

The paging system integrates with an outside server through the use of a shared file system. Outbound messages are written into a spool directory as individual files. Once a message is accepted by the paging system, a reference identification (RefID) is issued. An acknowledgment of receipt of the message by the mobile unit is passed back to the service provider once the message has been received and properly decoded by the remote unit. The complexity of the timing of message delivery is masked from the system integrator by using a shared file system as the means of integrating the paging system with a service provider.

Inbound messages from the paging system to the service provider are stored in a shared file system. Three types of inbound messages exist:

Unit Registration When a mobile unit comes into range of the paging system, it registers its presence with the system. This information is relayed to the service provider through the shared file system.

External Out-bound Message These are messages that are sent by the mobile unit to an address outside the system. This is typically electronic mail. It can also be a fax destination – or anything that is not another mobile unit in the paging system. Messages destined to non-pager units are transferred to the shared directory for processing by the service provider's system.

Replies These are messages sent in reply to messages received by the mobile unit. All addressing information is stripped away, since it is assumed that the service provider is maintaining a database of the state of all messages processed by the system. Only a message is passed with the data.

Three types of pagers were integrated into the Canard system: Motorola PageWriter 2000^{TM} with graphic capability, Motorola Tango^{TM} units with text only, and Motorola CreataLink^{TM} display-less devices for direct data connections with remote, operator-less devices. All three kinds of pagers have the capability of originating free form messages. The PageWriter and Tango's discrete form factor makes it extremely portable and likely to accompany the user almost everywhere.

Pagers provide some opportunities for modeling the user. It is usually safe to assume that the owner is the only one using the pager, and as such, the system can make some inferences about its use. For example, when the user originates a message, there is a strong likelihood that the user will still have the pager a few minutes later. For awareness applications, it provides the means of revealing the ability to communicate with the individual. One can also check the desktop computer for activity, and if none is present, make the assumption that the user is away from the desk, but reachable via pager.

Although two-way pagers send an acknowledgment back to the system to confirm that a message has been successfully delivered, it is not yet possible to know if the person receiving the message has actually seen it, unless they send a response. Today's pagers have primitive on-board message sorting capabilities and have the ability to alter alerts based on some simple message features. The next generation of pagers will have increased awareness of their state, and it will be possible for the system to know whether or not a message was actually read.

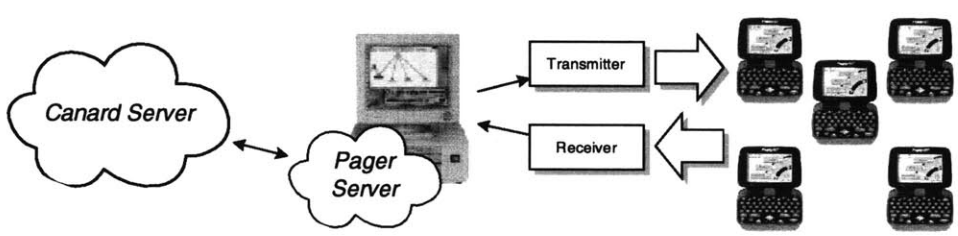


Figure 3-4: The Canard interfaces to a Motorola two-way paging system via a shared network file system. The paging system consists of a personal computer that controls the processing of messages to be sent and received over the air. A transmitter and receiver unit are connected to the personal computer via serial ports.

A mapping between Canard user names and pager identification numbers (PINs) is kept in a database by the Canard. In addition to the PIN, the pager address-type associated with the device is also kept (e.g., PageWriter, Tango, or NEWS). This allows for device-specific formatting to be done. The database is kept separate from the other user databases so that pager device maintenance functions can be more easily decoupled from the Canard system.

Whenever a message is sent out to a pager, a reference ID is issued. A copy of the outbound message is kept with the RefID. This is because the paging system does not keep information about the state of the message being processed. It is assumed that the service provider will do that level of message bookkeeping. When a reply comes from a pager, the original message is looked up in the database of outbound messages. The sender information is retrieved, and the response is relayed to the person using the same channels through which the message originated.

Inbound messages are converted into message objects for processing by the Canard system. The user's name is looked up in the PIN database maintained by Canard. If an explicit destination address and transport is specified, it is preserved in the message object. Otherwise, the address is looked up in the user's address book. If a mapping is not found, then the address is compared against the list of know Canard users. If that lookup fails, the message is sent via electronic mail to MIT's main mail server (e.g., pascal@mit.edu).

3.6.4 Electronic Mail Interface

Electronic mail (email) [Cro82] has always been a core service on the Internet. It allows simple asynchronous communication between people on the same computer, computers on the local area network, or computers on foreign networks. Modern email addressing schemes mask the complexity of the delivery of the message. Electronic mail is not restricted to the transmissions of text messages. It is possible to send complex documents using Multipurpose Internet Mail Extensions (MIME) [FB96].

Canard integrates incoming mail by creating system email aliases for each user. These aliases invoke a conversion program that takes the email and converts it into a Canard message object with the user as the recipient. All the original email header information is preserved in the Canard message object, allowing it to be used for later processing.

Currently, MIME encoded messages are converted into a text format by describing non-text information (e.g., images, sound, or other data) that is part of the document, and presenting any surrounding text. This allows users to receive the descriptions on text-only pagers.

Outgoing messages are handled by opening a stream to the UNIX mail system. Message objects are converted into a MIME encoded message that is processed by the UNIX mail agent. This allows modern mail transport technology to be integrated into Canard without having to change any of the core Canard technology.

3.6.5 Instant Messaging Interface

Instant messaging is growing in popularity [Mat99]. It provides a means for people to communicate with one another without having to worry about which machine they are connected to at the time. This becomes an issue as the number of computers available to individuals increases. It is not unusual for an individual to connect to the Internet from their home, office, or friend's home to check on email and chat. Before instant messaging systems, a user would look on various machines using the Internet Finger Protocol [Zim91] to check to see if their friends were connected to a particular machine. This made sense when there were fewer machines that were shared by large numbers of people. Instant Messenger system relieved this burden of determining user location. Typically this is accomplished through the use of a directory server that maintains a list of the availability of users and routes packets accordingly. Commercial Internet Service Providers are increasingly including some form of instant messaging as part of their community building tools.

Zephyr [DEF+88] is an instant messaging system developed at MIT as part of Project Athena, a distributed academic computing environment. MIT made available 400 workstations for student use on campus. Although these platforms were made by different manufacturers, a common computing environment based on UNIX was developed. One design requirement for Project Athena was the ability for students to be able to message one another without knowing where the student was physically located on campus. Zephyr addressed this system by using a two-layer server approach. Zephyr servers were responsible for routing communication between hosts on campus. "Host managers" maintained the state of which users were on a particular machine. Users would authenticate themselves using Kerberos [SNS88] and would be permitted to send messages to other users on the Athena system. Zephyr also provided a mechanism to evaluate the incoming messages and format them for display. This mechanism also allowed arbitrary programs to be used to process the messages. This mechanism allowed students to write their own interfaces to the Zephyr.

Interfacing Zephyr to Canard was straight forward. As with the paging transport, message objects were converted to a textual representation suitable for delivery over Zephyr. In general, the body, subject, and responses are delivered to the Zephyr addressee. Kerberos authentication was not used in the first implementation since no suitable scheme existed for an automated reverse channel back to the sender of the message. Response lists (a.k.a., "canned responses" on pagers) were converted to display "the sender of the message anticipates the following response to the message:" followed by an itemized list of responses.

3.6.6 Telephone

Telephones are probably the most ubiquitous two-way communication devices available in the United States. This makes them an attractive interface to integrate with the Canard system. The touch-tone keypads of telephones provide a mechanism for input to the computer without having to do voice-recognition. A variety of spelling schemes [Dav91] exists for using keypads to communicate to computers. The more successful schemes are application aware, such as Schmandt's PhoneShell[Sch93].

A telephone system's caller-ID also provides the opportunity to learn about the person calling[MS96][Mar95]. Actual use of the telephone can reveal the user's activities (e.g., is the person on the phone?) and present it to others[Man91]. That knowledge of activity can result in an interesting visual interface of a group's activities[Tuf90].

A Canard interface was created that allows users to interact with the system using a touch-tone keypad. This interface is intended to be similar to the one developed by Schmandt for PhoneShell. The interface consists of a Computerfone[Sun98] computer-telephone interface and a DECTalk speech synthesizer[Dig87]. The Computerfone interface does touch-tone recognition and connects audio lines to the telephone circuit. MIT's telephone system is equipped to provide caller–ID, which can be accessed through an ISDN telephone server the Speech Group developed[Won91].

The interface is intended for users to compose messages to one another and to review messages that were received by the Canard system. It is possible for the computer to record audio as well, so that voice messages can be left for other users.

3.6.7 World Wide Web Interface

The Canard system leverages the current popularity of World Wide Web (WWW)[BL89] to provide users with a consistent interface across the many computer platforms. This interface enables access from the privacy of a user's residence or from public access computers. In the current implementation of Canard, a central server maintains the user databases. WWW browsers provide a rich interface for Canard users to manage databases that affect their communication. CGI programs allow users to read and compose messages as well as consult and edit their personal calendars and address books. These programs require the user to identify themselves with a user identification and password, ensuring proper authentication before any personal databases are manipulated on the system.

With privacy an important concern, Canard's WWW server uses Secure Socket Layer (SSL)[FKK96] to assure that all communication between the user's browser and server are encrypted. By providing a secure channel, sensitive information may be relayed without fear that the communication will be tapped along the way.

The WWW is a tremendous resource for valuable information services that are freely available. For example, MIT students enjoy access to the Encyclopedia Britannica through the web. To allow Canard users the benefit of access to these resources, a Hypertext Transport Protocol (HTTP) connection is provided. This connectivity is achieved by sending the body of a Canard message object as the query to the remote web service. Results from the query are sent back to the user through the same channel as the original request. This allows users to request web pages and do simple searches from their pager.

Enterprising users can run their own web servers and provide useful CGI services. The benefit of doing this with Canard is that services can be provided both through Canard and any web browser without having to worry about how to convert the information for display on different devices.

3.6.8 Electronic Whiteboard Interface

Electronic whiteboards are an interesting example of large scale input devices that can be shared by a group of people. They are a familiar and convenient interface for the novice and a facile replacement for the traditional whiteboard. The markings on the surface are digitized by the whiteboard controller and converted into a representation suitable for printing.

These devices were integrated into the Canard system and installed into the public space of the undergraduate residence of one of the experimental study groups. Coupled with a large scale display, an LED moving sign, this interface makes an ideal two-way connection to a group in a public setting. Material captured on a whiteboard can be transformed and sent to the group's WWW page as an image file or sent to graphical pagers [RS94]. Remotely located community members can communicate back to the public space by using the LED display.

A current limitation of electronic whiteboards is that the identity of the author is unknown. It is not difficult to imagine that a digital camera attached to a whiteboard will be rigged to take a snapshot of the person pressing the "publish" button. Face recognition software [TP91] [MP95] could then attribute the whiteboard data to an author. Alternatively, one could perform a handwriting analysis of the markings to help determine authorship.

For the Canard project, an electronic whiteboard from Microtouch [Mic97] was used. It is relatively inexpensive and easy to interface. The device comes in a number of different

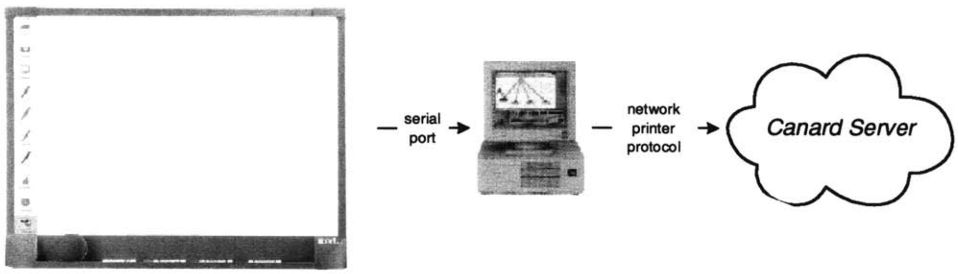


Figure 3-5: A whiteboard is attached to a personal computer through a RS232 serial port. Strokes from the whiteboard are processed by the personal computer into PostScript. When the user presses the "print" button on the whiteboard, the PostScript is sent to a network print queue on Canard. Canard then converts the image into multiple representations and creates a message object to transmit the image to its intended destination.

sizes: 1'x2', 3'x4' and 4'x6'. It is limited in that there is no published API to interface the whiteboard to customized applications. Also, given the digitizing technique used, only one pen can be detected at a time, prohibiting its use for collaborative drawing applications. The device connects via a serial port to a computer running Microsoft Windows.

The markings on the whiteboard are converted into PostScript [Ado85]. This allows the capture of images from the whiteboard by setting up a network printer spool on the Canard system. Images from the whiteboard are sent to the print spool, which ran a series of filters. First, Aladdin Enterprise's Ghostscript[Ala89] package is run to convert the postscript into a PBM portable bitmap representation. This bitmap is then converted into representations suitable for display on the WWW site and graphical pagers. Once the images are all prepared, a Canard message object is created so that the image can be delivered to the appropriate user or community. The destination of the image is determined by setting up a specific print spool for each community/user. The default settings of the whiteboard sends an image to the whole community. The print spool can be manually over-ridden in order to direct the image to another person or community.

3.6.9 LED Sign Interface

Moving sign displays are common in public spaces such as train stations, banks, and store fronts. It is convenient to convey short messages to a large number of people. Moving signs have the limitation of being able to display a small number of characters at a time. However, the dynamic nature of the display attracts attention to it. Messages are often repeated for the benefit of the viewers that may have missed a portion of the displayed message.

Virtual moving sign displays, or tickers, are common on computer interfaces. It is a popular means for transmitting stock and news information, such as PointCast^{TM}[Poi99], on the computer desktop. Netscape's browser has a special HTML mark-up tags for tickers. Configurable single line displays can used for group communication [FPSK98] by providing a single line display window on the user's desktop.

Public displays pose an interesting problem in terms of presenting personal information without sacrificing privacy. The use of bitmaps to reflect personal information activity is being explored. For example, "Joe" may be in the lounge watching TV with his peers when he sees his name on the LED sign followed by an e-mail icon in red. This would tell him that an important message is waiting.

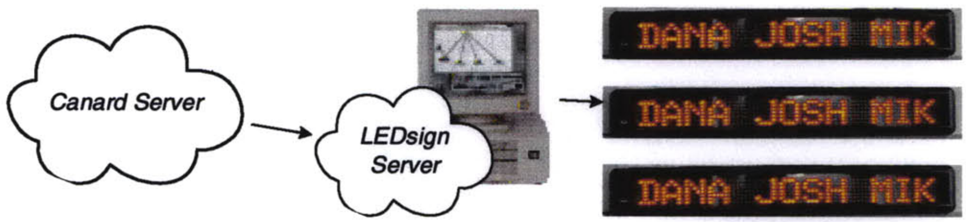


Figure 3-6: Canard interfaces with LED moving signs through a server running on a remote personal computer on the network. The Canard message server prepares the salient information for display on the LED displays. The LED sign server on the personal computer is responsible for managing the actual display queues by adding and removing messages at the appropriate times. One or more LED moving signs can be connected to a server through the use of RS232 serial ports. The LED sign server also maintains the databases for the virtual LED signs that are displayed on web pages.

A single-line LED moving sign display from Pro-Lite [Pro98] was used for the project. Its red and green LEDs are capable scrolling messages in 16 colors. It has a 32K memory divided into 26 buffers. It is capable of being remotely programmed through a RS232 serial port. The displays can be chained together and each one can be programmed separately by using programmable display identifiers. The display has internal timers that allow messages to be uploaded and displayed at a particular time for a set duration.

An interface was written for Canard that permitted the management of multiple displays connected to the system. Each of the display buffers and queue times are maintained on the Canard server. Text messages are passed as a message object to the transport layer of Canard. The message object is then converted into a form that the moving sign display server can interpret. The server also maintains the databases for the virtual signs.

The virtual moving sign display is written in Java [Bro95]. It is designed to emulate a Pro-Lite display. Canard provides the necessary HTML to integrate the virtual moving sign into their personal web pages.

Figure 3-7: Virtual ticker written in Java.

3.6.10 Family Radio Service Interface

The concept of public communication spaces exists even in the radio frequency spectrum. In the 1970's, Citizen Band Radio (CB) enjoyed great popularity with 40 radio channels set aside for public use. In response to its popularity, the Federal Communications Commission (FCC) eased the formal licensing process. No longer is it necessary to apply for a special call sign in order to operate a radio in the CB spectrum. CB radios exist in three distinct form factors: hand-held units (or popularly known as "Walkie Talkies"), mobile stations (usually installed in cars), and fixed base stations (typically found in homes). The hand-held units were popular with hikers. Mobile units were the mainstay of commuters who wished to get geographically appropriate road condition reports. Communities built themselves on the 40 shared channels. Most notable was the allocation of channel 9 for emergency use and channel 19 for highway reports. Channel 19 was used by truckers who routinely informed the rest of the on-air community of the locations of speed traps and accidents. No attempt was made through the use of technology to separate user's point-to-point communication on the channels. Anyone could listen to all communication on the channels.

The FCC recently set aside an additional spectrum for Family Radio Service (FRS). Like CB radios, they do not require a special FCC license to operate. They also have a set number of channels (14) that must be shared. However, the main intent of the band is to provide communication between a small group of people (or family) on an agreed channel without too much interference from others using the same channel. In essence, the intent is to create a semi-public communication space. Here, technology is used to separate communication by having a sub-channel code. This code is transmitted before voice communication is transmitted, allowing the receiving radio to un-mute itself only when a strong signal is received with the agreed upon code. Unlike CB radios, there is no squelch knob to attenuate the signal. In general, users of family radio are not privy to other activities on their channel.

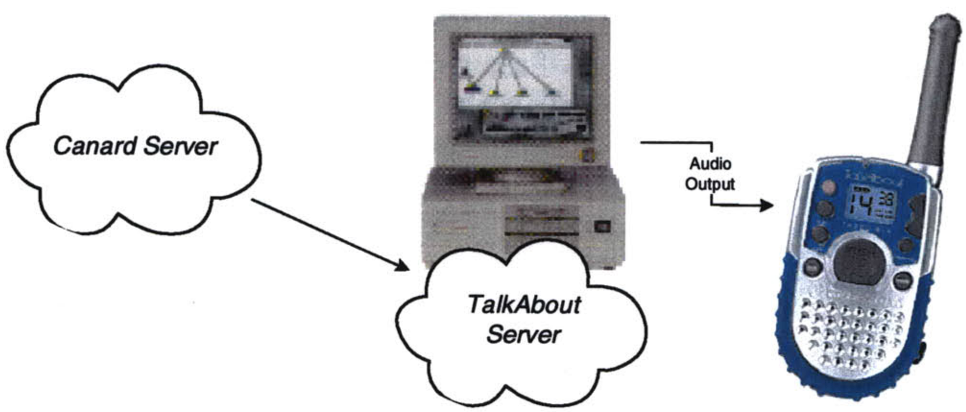


Figure 3-8: Canard interfaces with TalkAbout radios through a server running on a personal computer equipped with text-to-speech capability. The Canard server database does the necessary conversions of the messages to make them suitable for transmitting over the air. The TalkAbout server takes care of all the issues surrounding the use of the radio (e.g., pausing the transmission at regular intervals to prevent radio overload.)

Motorola manufactures Family Radio Service radios, called TalkAbout^{TM} [Mot98]. In order to extend the communication network, a transport layer for the TalkAbout radios was devised. The transport layer transmits the message over the computer network to a server that broadcasts it on a specific channel. The TalkAbout radios are capable of transmitting for about 60 seconds before an internal timer shuts off the transmitter. Then they need to be off the air for 10 seconds. The Canard TalkAbout server fragments the message by pausing for 10 seconds after each 60 seconds of transmission. Electrically, a Digital DECTalk speech-to-text synthesizer [Dig87] is connected via audio cables to the TalkAbout radio. The radio is able to activate its transmitter when it detects a signal in its microphone input (VOX operation). The speech synthesizer is connected to the system via a serial port.

There is no technical reason why any computerized audio cannot be sent over the radios in this fashion. However, as with the text-only pager messages, any unsupported media stream is described (e.g., "A thirty second voice message from Walter is waiting for you."). The main benefit for using an audio interface is that it frees the hands and eyes for other tasks, like operating a motorized vehicle.

Given the radios are on a public spectrum, the types of messages suitable for transmission are limited. There are community service messages that might be appropriate to "announce" over the air.

3.7 Summary

The Canard system was well suited as a platform to explore the ASE Framework. It allowed us to rapidly adapt it to the needs of our communities. The Canard messaging model is intended to be simple for people to understand. Its modular approach is intended to encourage participation of people with differing levels of programming proficiencies. The breadth of device integration by the Canard system developers is a testimony to the effective of this approach. However, simpler tools for novices to use is still lacking. Although there is an underlying principle to open up the system to anyone, in practice the system has been directly manipulated by a small group of developers. The field studies described in the next chapter illustrate that regardless of the technical barriers individuals will overcome the complexities and adapt the technology to their needs.

Although Canard offers opportunities with evaluating messages there has not been development in this area. This is due to the restrictions imposed by MIT's human subject committee as to how messages could be manipulated. Such opportunities are explored in great detail in Stefan Marti's Active Messenger project[Mar98a] which leveraged off of Canard technology.

Another barrier to fully appreciating the potential of the Canard system was the decision to implement it as a centralized server, rather than a distributed set of servers that resided on personal computers. This decision was made because it was easier to implement and deploy a centralized system. The decision did allow testings of the system with a community that did not have their own personal computers.

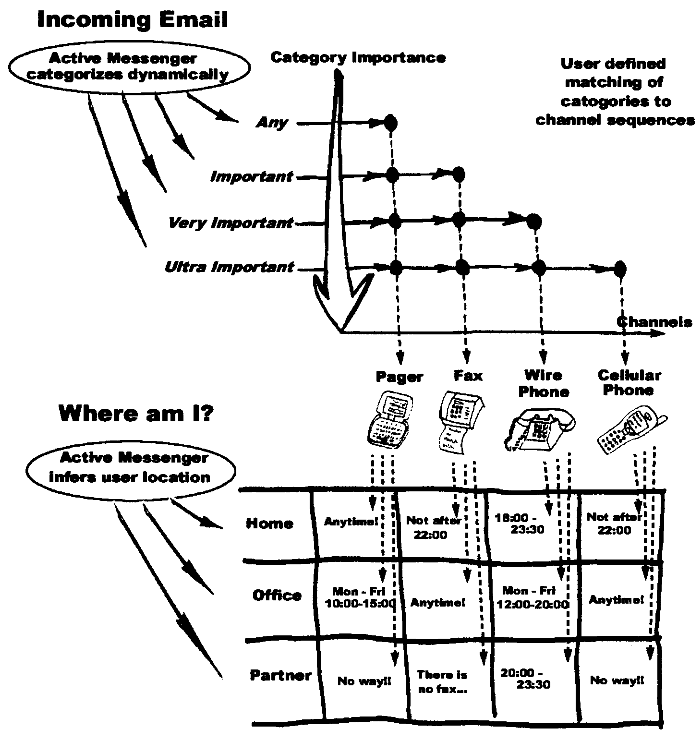


Figure 3-9: Active Messenger evaluates a message importance against the location of the user.

Chapter 4

Field Studies

The Canard platform was used by five communities: an MIT undergraduate residence in support of their group activities; a group of freshmen as part of a joint class project; the Media Lab's Speech Research Group in integration with their existing tools; three Media Lab research groups sharing a common space as part of a wireless environment; and 100 children attending an international conference at MIT in order to coordinate their activities with each other and with their on-line collaborators.

The five communities shared the characteristic that all community members regularly used electronic mail, but they were quite diverse in their telecommunications needs and their internal technical expertise. The communities were provided with different levels of support from the researchers. In some cases, support was simply a matter of answering rudimentary questions, while in other cases support including adding customized features. The ASE Framework described in Chapter 2 was conceived as a result of these field studies.

The time period that each community used Canard varied from one week to two years. All groups were observed on-site for extended periods. With the exception of one group, their communication contents and patterns were not analyzed. The researchers routinely invited the experimental subjects from one group for focus studies and one-on-one interviews. As a result of such meetings, new features were added to Canard.

Each community in the field study had an appealing set of characteristics (i.e., age group, technical proficiency, and community setting), but all seemed as though they would benefit from a communication platform designed to support group cooperation. Through successes and failures, each provided insights that lead to the ASE Framework. This chapter is devoted to reporting the experiences gathered from the field studies. Using the ASE Framework, each group study is broken into:

- an overview of the group,
- a discussion of the group's abilities,
- a discussion of support provided,
- a discussion of the efforts by the individuals and group to overcome the complexities of adopting the technology,
- examples of individual, group, and social constructionism,
- observations and reflections.

4.1 Experimental Protocol

Before engaging in actual field studies, an experimental protocol had to be devised and presented to a human subjects committee at MIT. The federally mandated independent review board at MIT is called the Committee on the Use of Humans as Experimental Subjects (COUHES). Project proposals involving humans as subjects are reviewed by COUHES with special consideration paid to the risks and benefits to the experimental subjects. The committee, working with the researcher, can propose modifications to the research protocol until all of the committee's concerns are addressed. No research involving human subjects can proceed at MIT without the committee's approval.

The Canard experimental protocol application [Che96] went through a long review process [Che97b] that delayed the start of the project by seven months and sharply curtailed what was permitted in terms of communication integration. Instead of an integration that included access to all available communication channels and databases, Canard's experimental use was transformed into a peripheral addition to the existing communication channels. The protocol changes required by COUHES shifted the focus of the project from the implementation of a broad range of technologies to the detailed observation the experimental subjects using a subset of the Canard platform.

Within the context of the experimental protocol, COUHES considered a sender of electronic mail to be an experimental subject. They were concerned that the experimental nature of the system might compromise the robustness and security of electronic mail delivery and they felt it was important to inform people of this potential risk. Given this definition of an experimental subject, the investigators were asked to secure informed consent release forms or send notification of risks and benefits of the experiment to each sender of electronic mail, including incoming electronic mail. Initially, only a web interface (see Figure 4-1), which allowed the experimental subjects to send messages to pagers, was permitted, since it provided a mechanism to alert the user that this was an experimental system before actually sending a message.

Securing informed consent of electronic mail users was not a request that could be easily satisfied. The test communities used electronic mail extensively, for both one-on-one communication and group communication through the use of large electronic mail lists. It is technically possible to send an automated response to each incoming piece of electronic mail in a manner similar to the UNIX^{TM} `vacation` [Uni91] program. COUHES argued that this mechanism be used to inform potential senders of electronic mail into the system of the risks and benefits of the experiment.

An automated response approach is not without several risks. A significant number of automated electronic mail list servers automatically remove users from the list if they received an automated note. This is because many list servers were designed with a simplifying assumption that any automated response received is the notification of an undeliverable message. A privacy concern with automated response is that the sending of notification reveals membership to mailing lists without regard for the consequences. For example, a bigot could send a note out to gays@mit.edu and as a result of the automated response mechanism learn who were the list members. On the "netiquette" front, sending unsolicited electronic mail is frowned upon.

The protocol was modified to make reception of electronic mail by the Canard server a redundant system. Electronic mail would be received by the user's primary mail server and then copies would be routed to two destinations – their current electronic mail server and the Canard server, which would then, when appropriate, route the mail message to the

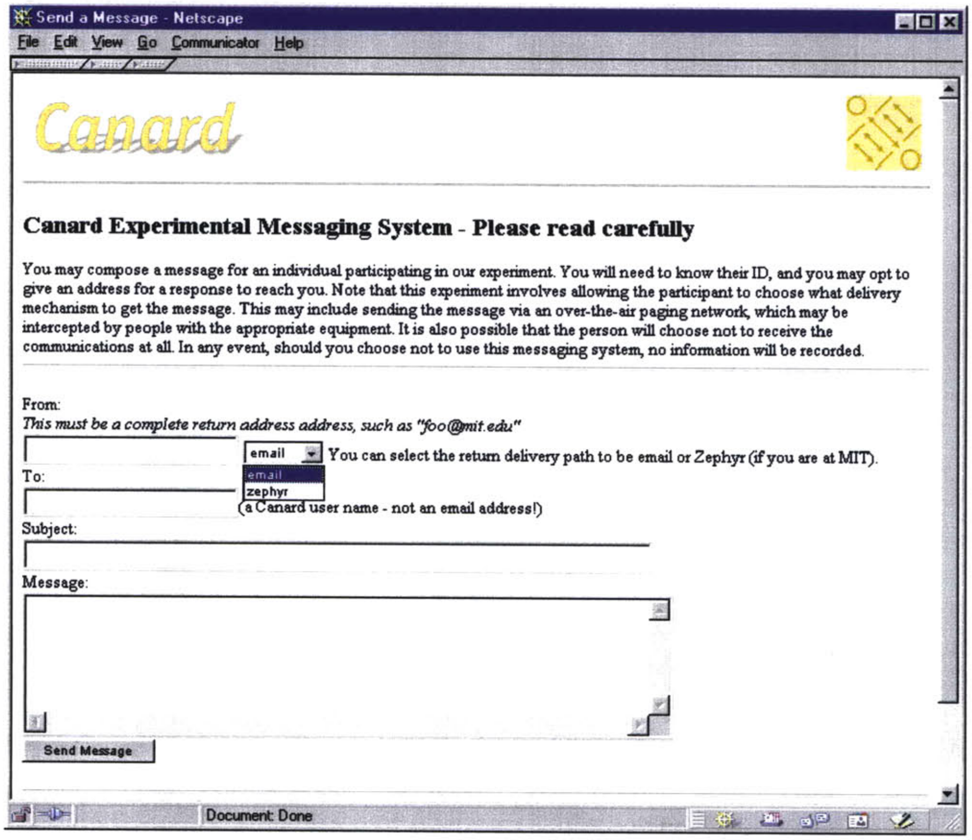


Figure 4-1: Initial Web-based interface for submitting messages to be delivered to pagers. A reply channel, electronic mail or Zephyr, could be specified in the interface. By using the Hypertext Markup Language (HTML) source of the interface, Canard users were able to create their own custom Web interfaces and gateways to other communication channels.

pager (see Figure 4-2). This change in protocol meant that, regardless of the features of the communication, multiple copies of the communication would be kept on separate servers. The goal of reducing redundancy was sacrificed to robustness. After an additional three months, the protocol for allowing electronic mail [Che97a] to be added to the experimental protocol without notification was approved.

The experience with COUHES illustrates a number of important points. First, the current forms and practices of electronic communication usage are not well understood. Second, once an individual or group has adopted a model for understanding a particular technology's use, it is very difficult to change it. An apparent generation gap was evident when the committee fixated on a model for electronic mail that equated it with the services provided by the United States Postal Service. The reality is that electronic mail is passed from one independently administered machine to another with no government oversight. More astonishing is that all this communication is passed through public networks unencrypted. This surprised one senior researcher at the MIT Media Lab. He had assumed that, with all the advances in telecommunication, encrypted streams would be the norm rather than the exception. A more accurate model would be to compare electronic mail with a system where postcards are passed from person to person on public streets.

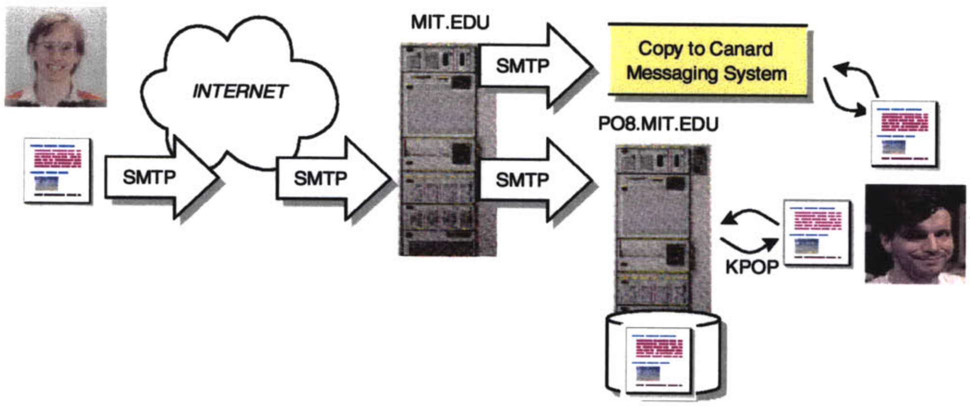


Figure 4-2: Electronic mail in the system approved by COUHES is first routed to the main electronic mail server, and then a copy is sent to the Canard system.

An important lesson learned from the experience with COUHES is that a most difficult challenge faced by researchers is in explaining these underlying issues to their experimental subjects in order to obtain truly informed consent. If a group of highly educated professionals had problems understanding the inner-workings of the simplest of technologies, then it suggests that explaining the risks and benefits of new systems to laypersons will be difficult.

There are many factors affecting the acceptance of technology by society. Established social structures [Rog95] are good examples. In the case of COUHES, their established position within the MIT community makes the task of presenting a new protocol by an unknown researcher more difficult than for a respected one who had regularly conducted business with them.[1] Their task is to protect the rights of experimental subjects and, when faced with a newcomer, they err on the side of caution. This process of protecting the rights of human subjects allows an independent group to examine the risks and benefits with minimal conflict of interest.

One encouraging observation is that despite the many real and artificial barriers that exist between the user communities and access to Canard, the system is very heavily used. This suggests that Canard is meeting an important need.

4.2 Third East Undergraduate Living Group

4.2.1 Overview

Finding a suitable community within MIT to evaluate the effectiveness of Canard proved to be a complex process in itself. A preliminary survey of potential test communities within the MIT undergraduate residential system was conducted in the Fall of 1995. An on-campus living group was sought who would immediately benefit from the messaging system and would have the ability to immerse themselves in the technologies and extend the system as needed.

Undergraduates tend to be self-sufficient, reducing the need for an intensive support structure. Residential living groups at MIT share a common space and have communal

[1] A personal discussion with the COUHES chairman two years later confirmed this was indeed the case.

Figure 4-3: East Campus Alumni Memorial Housing – a traditional dorm located at MIT. The architecture supports frequent impromptu face-to-face encounters between students as they move within the dorm.

needs, such as making entertainment arrangements for the group, which extend beyond the physical confines of the residence. They also tend to be small in number, comprising of approximately 40 students.

East Campus Alumni Memorial Housing (see Figure 4-3) is a dormitory located across the street from the Media Laboratory. This physical proximity to the Media Laboratory made it a convenient choice. The dormitory is composed of two parallel buildings, each five stories high. Each floor is called a hall. For historical reasons, each hall is broken into three distinct entries that are connected via a single corridor (see Figure 4-4). Along this corridor are the students' rooms, which are mostly single occupancy, but are also some doubles and triples. Each hall has a common kitchen, five bathrooms, and two lounges. Lounges connected to the kitchen area are spacious enough to hold all the hall members for group meetings. These kitchen lounges have dining tables that usually double for studying. The second lounge is half the size of a kitchen lounge and is used mainly for social purposes. Although each hall has the same physical resources, the decor of each tends to reflect the whims of the residents.

In order to select a test group from within the East Campus dormitory, a meeting was held with the house government to discuss the proposed project and see if any of the halls were interested in participating. The house government is composed of a president, vice-president, and treasurer that are elected in a yearly dorm-wide election. In addition, there is a representative from each hall. The house government seemed receptive to the idea of their dorm being the site for the experiments. Their enthusiasm reflected the students' interest in using the two-way pagers.

Meetings were arranged with halls that were interested in hosting the field test. Three

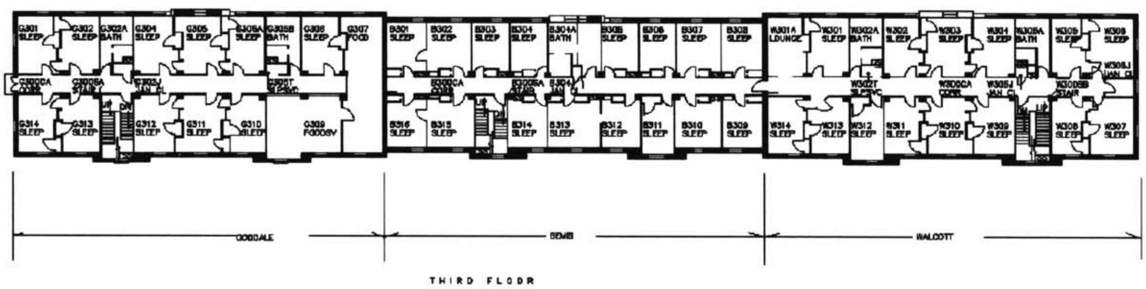


Figure 4-4: Floor plan of the Third East undergraduate living group. Common spaces include a Kitchen lounge at G307 and a smaller lounge, located at W301A. All of the rooms open onto the center hallway.

halls invited the researchers to give presentations. Finally, one hall invited the researchers back so they could present their reasons why they should be the test site. The demographics of the hall were interesting in that a significant number of students were not electrical engineering or computer science students. There was a good balance in terms of gender. Ultimately, the decision was made simple in light that they were the only ones still interested in investing time in the project. The other halls opted not to participate.

> *Third East, located on the scenic third floor of the east parallel of East Campus, is home to 44 undergraduates and one graduate resident tutor. We are 25 males and 20 females, with a racial mix of about 50% white and 40% Asian (and 10% other). There are 14 frosh, 10 sophomores, 12 juniors, and 8 seniors. As anyone probably could have guessed, course 6 is the most popular major, with 9 upperclassmen in EECS. Other popular majors are 2 and 8 (see table A.1 for details). With 6 frosh already planning on choosing course 6 (see table A.2), it looks like that trend will continue. Two cats also live on the hall.*

The residents living on the hall refer to themselves as "Third Easters", reflecting their location in the dormitory. The demographics of the group are similar to others on campus. MIT students refer to their majors by numbers (e.g., course 18 is a math major). Computer Science and Electrical Engineering (course 6) is the most popular. The hall has a significant number of students who can program computers. A large number of students have their own computers in their rooms. There are a number of "public" computers that residents can use in the lounges. At MIT, one can not survive without knowing how to use electronic mail or the Web (registration process requires the student be able to pre-register for classes using a Web browser). Electronic mail is the primary form of on-campus communication for these residents.

Each resident has a phone line in their room. Not all use answering machines. MIT's phone system has a distinctive ring pattern to distinguish on-campus calls from off-campus ones. The students mainly use phones for communicating to people off-campus. Some students use cordless phone while they are out of their rooms, while others will go to extremes to receive calls when they are elsewhere.

> *When I am expecting a pizza delivery I run a long phone cord from my room into the lounge.*

One student uses his computer as an answering machine but has not written custom software for tying his phone system into other applications. This is not due to his inability to program, but due to the lack of time to immerse himself in the task.

In lobbying to be part of the experiment, the residents produced a proposal to be considered for the experiment. Part of the proposal was a hypothetical scenario of how Canard would fit into their daily routine:

> *A Day in My Life with Canard*
>
> *I awoke to the loud beeps of my alarm clock. I took a quick shower to wake me up and was ready to face the day. I picked up my Canard and smiled at the Squanch*[2] *on its cover. I opened the cover and looked at my tasks for the day. "Oh, no! That problem set is due tomorrow!!" I must remember to meet with Ben Bitdiddle and Alyssa P. Hacker*[3] *for help this afternoon. I then recalled that my Canard knew what I knew and would remind me when the appointed time came. I went to class.*
>
> *Near the end of the class, I received a page regarding new mail that met the requirements of my e-mail filters. It was a posting to reuse for a VT220 that nobody wanted. "Perfect for my Linux box! I'll be the first to respond!" I selected the item to quote the message and send a quick note back to the person stating I was interested. A few minutes later, I received notification of mail which said I had responded first and the VT220 was mine.*
>
> *Later in the day, after another class, my Canard vibrated and informed me that I had library books to return. "Oh, I almost forgot!" I thought to myself. After that task I enjoyed, well, I ate lunch at Walker before punting, er, attending another class. During this one, Louis Reasoner paged me using a Zephyr-to-pager gateway. "Sorry to bug you," he wrote, "but I'm having a lot of trouble with my 6.001 problem set, as usual. Do you think you could help?" I selected my 4th canned response, "Sure.. I'll see you on the hall tonight." He replied "Thanks! I don't understand why none of my code ever works..."*
>
> *After my last class, I felt relieved to be done for the day and was headed home, when my Canard reminded me that it was time to meet with Ben and Alyssa. I headed to their hall like we had arranged. When I got there, Ben was there, but Alyssa was not. I then received a quick page from her: "Running late.. be there soon." Ben and I smiled, knowing that with her reputation for punctuality, that was one message we knew she would use a lot. An hour or two later, Albert paged me: "Hall feed/meeting Real Soon Now." Oh that's right! I nearly forgot. Well, I'm glad Canard has helped to improve my memory. I definitely wouldn't want to miss the food, and after the meeting a couple of us planned to change some of the preprogrammed Canard messages so as to ameliorate communication that night as we went to ... explore the tallest building on campus.*
>
> *After a long day, I decided to watch a movie. But where is the Hall TV? I need only consult TV Comm's schedule using my Canard. Ah, Santiago has it. Cool.*

---

[2]Squanch is a cartoon character that acts as the hall's mascot. It looks like a coat hanger.

[3]Ben Bitdiddle, Alyssa P. Hacker, and Louis Reasoner are fictitious characters used in MIT's introductory computer science course, known by its course catalog number, 6.001.

At the end of the night, I reviewed the Canard stats for the day on the PC that came with my Canard and edited some of my e-mail filters and Zephyr filters. I also scheduled some more activities for the next few days, so that Canard would serve as my memory when the time came.
–December 23, 1995

In order to minimize the intrusion on the test community, interactions with the group were limited to:

Brainstorming meetings where the goals of the experiment were revealed and feedback was received as to how well these goals matched those of the community.

Individual interviews with the residents which were essential in getting an accurate profile of the hall.

Electronic surveys in the form of a questionnaire sent to the hall about specific aspects of the project.

On-site visits in order to account for the physical setting when designing devices to be placed in shared spaces.

Close observation of the group's communication. The Canard development team was added to the hall's mailing list, which provided a window into ongoing debates and everyday activities.

4.2.2 Ability

Participating in Canard required acquiring few new skills on the part of the residents. To receive messages on the system, they needed to know how to manipulate the Athena database that controlled how electronic mail is routed. Access to the Canard databases was made possible through a web interface.

Canard offered the residents a number of opportunities to extend the system. Students with their own personal computers running WWW servers were able to offer web-based services to other members of their community. In the way that the Canard system was presented to the residents, it was not possible for them to take the technology and use it elsewhere. They did, however, have the capability to operate their own Canard servers.

4.2.3 Support

For starters, we expect that Canard would make life easier (why get involved with something that makes life harder?), by allowing people to communicate better and find each other more easily. As many of us are interested in the technology, we expect that there would be opportunities to hack on the system and improve it. We certainly hope that there would be some technical support from the researchers in the form of help getting started and maintenance of any communications servers that are necessary for the functioning of the system. Also, feedback from the researchers would be welcome, such as how you think the project is going and what we can do to help work towards your research goals. Finally, we expect that we could get pagers for some off-hall friends (e.g., Dwight) who interact a lot with hall residents.

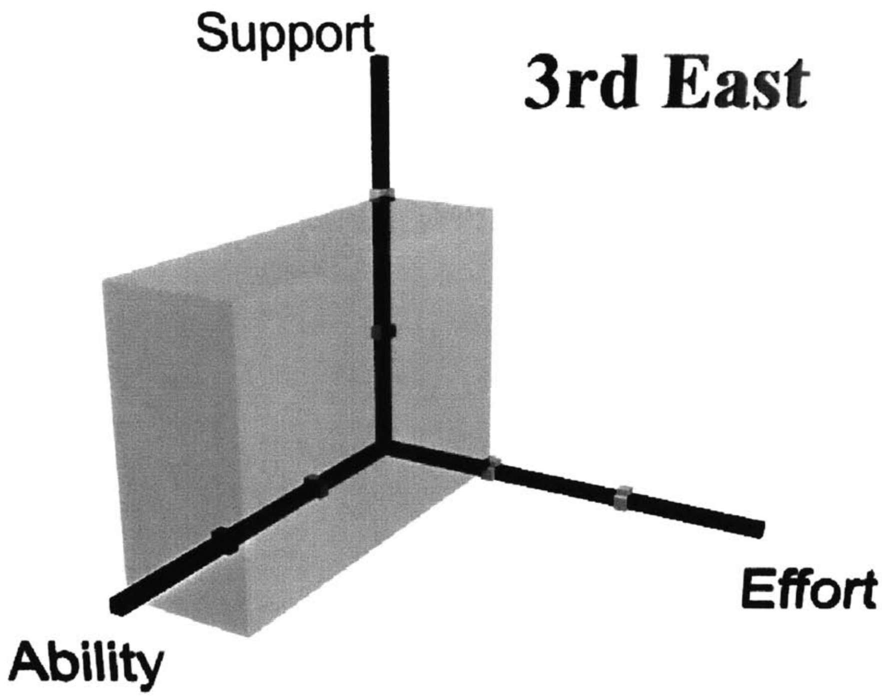


Figure 4-5: Third East Undergraduate Living Group. Residents had a full spectrum of abilities to use and extend the Canard system. Spot intervention was provided to add new features as needed. Overcoming the complexities adapting to us the Canard proved to be effortless.

> *Most importantly, we expect that it will be a fun project.*
> *-from Third East's proposal (see Appendix A)*

Equipment and programming expertise were provided as needed. Any hall member who wished to participate in the research was given a two-way pager. An LED sign and electronic whiteboard system were designed to be placed in one or more of their lounges. Particular attention was paid to providing a web site tailored to the hall's needs. The web site contained an archive of messages sent to the residents and a snapshot of the last image drawn on the electronic whiteboard. Also planned, but not implemented, was an interface for a group calendar.

4.2.4 Effort

> *If we get the project, you can expect a group of people that is interested and excited about using this technology and discovering what kind of things can be done with the system. The vast majority of the residents on the hall interact with each other (i.e., we have very few "ghosts" who are hardly around), so you can expect a high level of participation in the project. Especially with the large number of enthusiastic freshman on the hall, you can also expect a sustained level of participation in the future. You can expect us to try and use all the functionality of the system and to push the limits of what the system was designed for. To this end, we are also more than willing (nay, eager) to hack on the*

system and create any new software that needs to be written, or any software that sounds cool for that matter. If there are problems with the system or things we don't like about it, we will definitely provide feedback about how the project is going and what could/should be done to improve it. Most of us aren't afraid to be honest about "what sucks" and what doesn't.
-from Third East's proposal (see Appendix A)

From the very beginning, the residents wanted to take an active role in the development. They were willing recipients of any labor the researchers put into the project, but if need be, they were ready to add to it as well.

4.2.5 Constructions

As noted earlier, the scope of the experiment was sharply curtailed by COUHES and, even more frustrating, it was significantly delayed. As a result, the intended date for delivering the experimental apparatus to the students was pushed from the beginning of the semester to near the end of the semester. Also, the students were told they could only submit messages to the pagers through a web-based interface. In this project's first example of constructionism at work, the students wrote their own software to circumvent the COUHES restriction. They created an electronic-mail-to-web gateway to submit messages to their pagers.

This was easily accomplished since the protocol COUHES approved allowed for an electronic mail return address to be specified on the web form. The students wrote a program that would route electronic mail to a pager using the web interface.

Given the way the Canard web interface was implemented, it was impossible to know if a message is being submitted by a human or a machine. It was only because the researchers were on the resident's mailing list that it was discovered that the illicit gateway existed. Fortunately, COUHES did away with the restrictions a few months later.

Since the researchers could not provide the students with the communication gateway they needed, they took it upon themselves to author their own. Judging from MIT's mail forwarding directory, the students who implemented the gateway had sufficient trust of their peers that allowed the gateway to become the de facto standard for this group. The exploited the opportunity to support community needs.

This raises an ethical question. Should the researchers have stopped the experiment because of the rogue gateway? The community's taking control of the software would repeat itself during the course of the experiment and was, in fact, exactly the kind of behavior the researchers wished to promote, but in a more controlled environment.

Personal Interfaces and Gateways

In addition to the electronic-mail-to-web gateway the students wrote other gateways that they shared with their community. A group of residents wrote a program called "Cwrite" that mimicked Athena's Zephyr command line program "zwrite". With this program students could type a message to be sent by the Canard system from any Athena workstation. Replies from Canard could be sent back to the student on the Athena workstation using the Zephyr instant messaging system. A more industrious student wrote a server that would automatically route Zephyr messages from Athena to his pager on the Canard system.

Although an API had not been provided to the students, they were still able to interface easily to the system. Some interfaces to Canard require no programming skills are all, e.g., a number of residents integrated the web interface onto their home pages (see Figure 4-6). Others provided electronic mail address links on their web pages that would route messages to them on Canard.

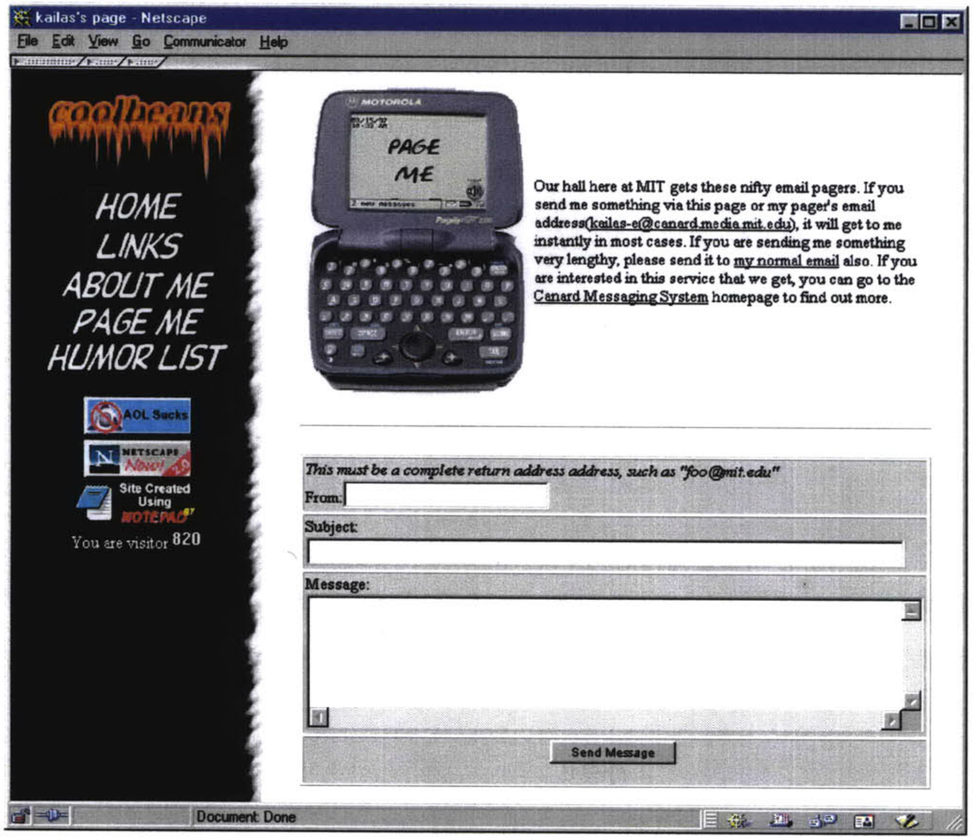


Figure 4-6: A web interface to Canard appears on a resident's home page. The resident model used the Canard web interface as a template for his page. Other residents provided electronic mail aliases as a means of routing messages directly to their Canard two-way pagers.

Taking Charge

One particular system reliability issue that the residents took under their own control was the verification that the Canard electronic mail forwarding was in synch with the MIT forwarding. MIT assigns all the students a "Post Office" from a pool of mail server machines. Which mail server machine the student is assigned can be changed dynamically by MIT. This allows the load to be distributed evenly and the easy replacement server machines. The problem is that the mail forwarding facility at MIT only gets updated once a day. It is possible for a server to change and electronic mail not to be delivered to the correct location. So the resident hackers wrote a monitoring system that checks the status of the MIT "Post Office" servers and notifies the Canard users on their pagers if there is a change that requires their intervention. A similar note gets sent to the researchers to make sure that messages are archived safely.

4.2.6 Other Observations

Non-participants

For a variety of reasons, not everyone on the hall wished to participate. Four residents chose not to participate (which was okay, since the communication model was designed to be transparent, meaning that if a person did not have a pager, the messages would be directed to their MIT electronic mail account). Two were particularly vocal about how this experiment was a waste of MIT tuition money (in fact, the experimental apparatus was provided independently by an equipment grant from Motorola). The dissenters did not actively try to stop others from participating. Nor did the participants attempt to coerce those who did not choose to participate into changing their minds.

The Extended Community

Although a pager was given to each of the residents, it was not certain if this was sufficient to cover their community. As noted in the residents' proposal, they considered certain alumni as members of their hall. During one of the group discussions, residents were asked if the assumption that giving the hall members each a pager would help contribute to better community communication was correct. The answer was mixed.

One group of residents felt that they tended to mingle with small groups off the hall. A Motorola representative, who was present at the meeting, offered an additional pager to each of the students present. They were allowed to give the pager to whomever they wished. There were a number of possible ways the students could have distributed the pagers:

- give to a close friend or significant other,
- group with other residents receiving additional pagers and decide to create a new community of pager users,
- use the additional pager dynamically by creating an alias and loaning out the pager to different people to accomplish tasks,
- give it to no one.

Roughly half the students receiving an additional pager gave it to a special friend or significant other. Half did not request the additional pager. One student asked for a second additional pager because he had two "special friends" and one had learned about the other getting a pager. (An additional pager was made available to help the student keep his friends at peace.) One could hypothesize that those in the group who did not collect an additional pager did so in order to not play favorites among their friends, i.e., to keep a social order. An interesting side effect from the distribution of additional pagers was that two were given to students living in the same hall.

The result was a stronger bond between the two halls as was observed by a member of the second hall, "Our hallmates are participating in more of Third East's activities than ours."

Impact of Technology on Community Identity

The residents decided that the project would be very appealing to the "geekish" elements of MIT and that, during MIT's residence selection period, the existence of the experiment should not affect a freshman's choice to live on the hall. Efforts were made not to disclose the pagers to incoming freshmen while they were visiting the hall. These efforts were effective. The freshmen who were housed on the hall were surprised to learn they could be part of the experiment if they wished. All chose to, although a number stopped carrying pagers after the novelty wore off.

Public Spaces

The researchers had the opportunity to meet with the residents as a group to discuss potential extensions to the system. One such meeting centered on the subject of installing an LED ticker sign coupled with an electronic whiteboard in one of their lounges. Residents were polarized by the suggestion. Through face-to-face and electronic mail discussion, it became apparent that the residents categorized their activities into two groups: Tooling and Punting. Tooling included activities and communication directly relating to their academic mission (e.g., graduating!). Punting encompassed all other activities (e.g., mostly social). A good number of residents were working on trying to graduate, and hence viewed the Canard project as a "punting" activity. Placing a whiteboard and LED sign in a space which they used for serious work disturbed them. The deployment of the whiteboard and LED sign was delayed until a time when there was less academic pressure present.

One resident thought it was a mistake to ask the hall for an opinion. Rather, the whiteboard and LED sign should have just been installed in a unilateral fashion. With the students preoccupied with their "tooling" activities, there would have been no debate, and eventually the extensions would have been accepted by the group.

Figure 4-7: Electronic whiteboard and LED sign installation in lounge. The whiteboard and LED combination provided a two-way link between residents on and off the hall. The LED sign also provided a means of displaying the communication activities of students who were off the hall.

Eighteen months later, the whiteboard and LED sign were installed in the kitchen lounge. This equipment was received without debate and the residents took immediate interest in it. One selling point was that the markings on the whiteboard were recorded as a PDF file, so that students working on academic assignments could preserve and print a copy of their musings. The students felt the whiteboard was less important than the LED sign. To date, the majority of writing on the board is artistic and serves no function other than to amuse the residents.

The LED sign proved to be a source of activity for the residents. One student took it upon himself to send regular "fortune cookies" to the LED sign. The result was that the LED sign was continuously displaying messages. This non-stop display caused another student to remark that it was "annoying" that she could not tell what was an important message for the group. This prompted a suggestion that the "fortune cookies" be sent five minutes before the hour as a means of communicating that it is about time to run to class, a "tooling" activity!

There was no further discussion about the LED sign for about a month. The resident who had been exploring what messages to send to the LED sign left the hall. As a result, the LED sign remained blank apart from debugging messages that would appear after a system reboot. A message was sent from the graduate tutor that the LED sign had been disconnected from the Canard system and instead it had been connected to a server that a student had opted to write from scratch (He did not exploit any of the Canard features). The same student also wrote a web-based graphical editor that enabled the posting of drawings to the sign. In essence, the student hijacked the LED sign from the experiment. The residents were provided with a second unit and their changes were incorporated into Canard.

During group discussions, another issue concerning the LED sign surfaced. The LED sign is essentially a one line by 16 character display. The residents complained that it was difficult to read messages originating from electronic mail as our system contained a verbose preamble before the message was displayed. More problematic was the lack of context (e.g., who sent the message and what was the subject of the correspondence) when watching the sign in the middle of displaying a long message. Even though attempts were made to use color to differentiate the salient features of the message, it proved insufficient to reduce the overwhelming experience of a non-stop text display of long messages. As a results, viewers tended to tune out the display. A similar criticism was raised during the author's general examination. A prototype LED sign was present during the examination to show the scope of the devices used. During the examination the signed continually scrolled messages of a banal nature. On a normally sedate internal mailing list at the Media Laboratory, a back-and-forth discussion had erupted on how the best way of destroying an electronic "pet". The messages were particularly long, and the incessant display distracted the examiners. Ultimately, one of the examiners turned the display around so it could not be seen.

Two-way Pager Usage

As a group, over 25% of the residents carried the pagers with them at all times. Another 25% carry the pagers when they are expecting an important communication or felt that there will be a need to coordinate activities while moving around town. Most of the students perceived the pagers as a means of instantly receiving and sending messages. The Canard system was sufficiently unreliable that they viewed it as a secondary means of seeing their electronic mail communication.

Coordination of Activities

One year, two of the hall residents were the dorm government's president and vice-president. They immersed themselves in the Canard messaging system to coordinate the activities relating to their official roles. They could also communicate with one another with the expectation that messages would be delivered in a timely manner and that administrative decisions could be reached quickly without having to meet face-to-face at a set time or physical location.

Other students found the pagers useful for coordinating off-campus activities such as buying movie tickets. This freed other members of the group to engage in other activities and, as a group, operate in an efficient manner. When a particular movie selection was sold out, they could quickly negotiate an alternative selection.

Timely Opportunities

The "hard core" pager users found that they were able to respond to time critical messages reliably. For example, `reuse@mit.edu` is a mailing list for used equipment being made available for free around MIT. This list is very popular and equipment advertised on the list is typically gone within a half hour. It is common that the first responses to the messages carry a `@canard.media.mit.edu` – a message from a Canard pager. Certainly the pile of used equipment that has accumulated in the lounges is a testimonial to the pager's success in receiving time critical messages from other channels.

Technology and Fashion

There are a fair number of residents that do not ever carry the pagers. A common problem is that they are perceived as too big and bulky. In particular, current fashion does not afford young women a good place to secure the pagers. Even those with belt holders have found them to be destructive to the belts! Most importantly, the size of the pagers draws attention to the wearer, which is a quality that some residents do not like.

Indirect Benefits

The residents who did not use the pagers did have indirect benefits through the use of other people's pagers. A common activity was asking someone to send a message on their behalf. These residents realized that the pagers offered unique benefits and were appreciative that others within their community were willing to share the resources with them.

Most Wanted Feature

When asked about what would be an important community function to add as a service to the messaging system, almost all of the students replied that they wanted the ability to locate people – using whatever means available. They specifically suggested using MIT's Zephyr [DEF+88] and the `finger` program [Zim91] to check on specific computers.

Connecting to electronic phone directory listings, such as MIT's on-line directory, was considered a useful but not a vital feature.

Group Awareness

When the researchers proposed having a listing of people who were observed using their pagers displayed on the lounge LED signs, it was met with lukewarm enthusiasm. The residents could see some utility in knowing who was around, but it was not something they thought they would need on a regular basis. The residents expressed an interest in a "pull" technology rather than a "push" for locating other residents and friends. A concern was how a person would control their "visibility" to the group.

Communication Expectation

The subject of knowing who was around led to discussions about communication expectations. When a message is sent through electronic mail, it is unclear how long it will take someone to receive it.

With the Canard system, the observations about the community could be used to set expectations by communicating back to the sender if the intended recipient was recently observed. This feature is feasible within the system but has not been implemented.

Community Services

The residents were enthusiastic about the promise of being able to create gateways from their pagers to web services. The problem in the initial Canard implementation was that it assumed that the residents had easy access to web servers and could write the programs themselves. A resident complained that although it seemed like a cool idea, he did not know how to write CGI scripts. Another resident complained that he did not have a computer to run a web server on. These critiques lead the researchers to try to devise a way for the Canard servers to host web service gateways.

4.2.7 Reflections

The residents of Third East were a great group to work with. They were enthusiastically engaged in the process of developing the system from the beginning. In short, they co-owned the approach and solution. As a result when the system failed, they were willing to work to solve the problems that caused the failure. In addition, they added value to the system by developing their own interfaces and services. Granted, MIT students are not a typical community since they are in a technology-centered environment, but socially they have similar needs to undergraduates at other schools.

4.3 Junior Summit

4.3.1 Overview

In November 1998, 94 children from 54 countries came to the Media Laboratory to attend Junior Summit. The children ranged in ages from 10 to 16 years old. They had been selected as delegates by their peers to represent 3000 children from 139 countries who had been participating in an online forum during the three month prior to the summit at MIT. The online forum had been split into 20 discussion topics – ranging from the environment to child labor issues. The goal of the summit was to focus the discussion topics into a smaller number of action plans that could be reduced to practice.

Figure 4-8: Junior Summit delegates using pagers at breakfast. It was common for Junior Summit delegates to send messages to one another in the same room. l-r Jacqueline from Canada, Henrik from Norway and Jose from Bolivia.

The perceived challenge for the Canard researchers was to help bridge the transition from an established virtual online community to a physically-based community. The delegates would be in the Boston area for one week, traveling daily between a hotel in Boston and the MIT campus. It was deemed critical that a reliable communication infrastructure be in place for them.

The mobile needs of the delegates (see Figure 4-15) made Canard a good fit. The delegates were roaming around the laboratory and attending meetings during the day. At night they were moving around the hotel between their rooms and the common areas.

4.3.2 Ability

The children who attended Junior Summit participated in online forums and communicated with one another via electronic mail for three months prior to the summit. By the time the delegates came to Cambridge, they were familiar with each other's names and interests.

4.3.3 Support

The Junior Summit needed and was provided an extensive formal support structure. They needed to be trained rapidly in how to use the communication devices. Special applications were developed for the delegates to use on their two-way pagers.

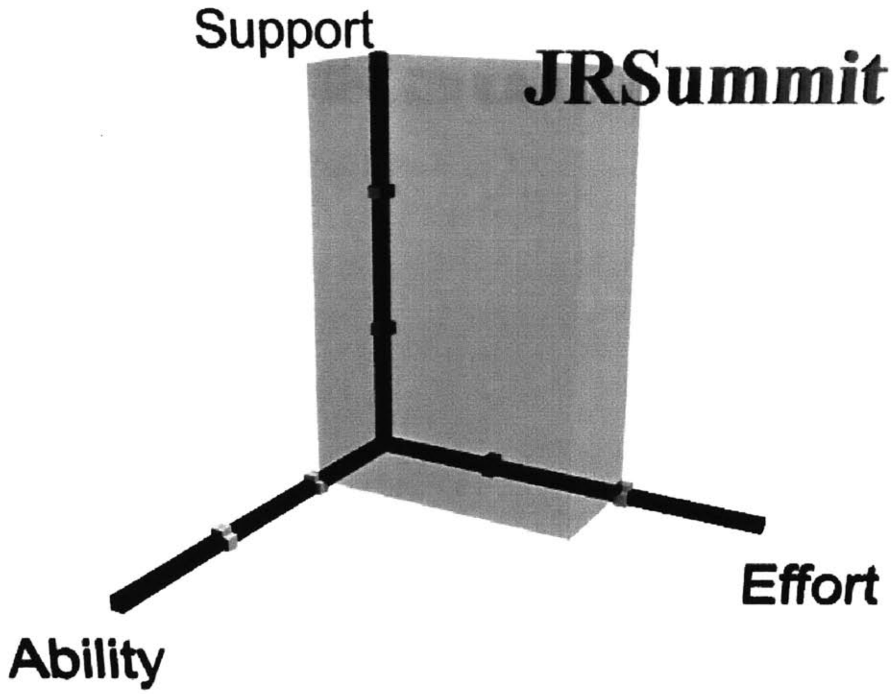


Figure 4-9: Junior Summit delegates had basic computer communication skills. The Media Laboratory provided full time support during Junior Summit to ensure that the delegates made a smooth transition from their online forums to the week long conference in Cambridge.

Equipment Support

Motorola provided 100 pagers to give to the delegates. The organizers of the event did not feel it was necessary to have units for the adults, since they would be communicating by two-way radios. This decision by the organizers caused the adults to be segregated from the communication channels employed by the delegates.

Instruction

> *On the first day I got my pager, I was quite confused how to use the pager. When in the dinner hall, I was absorbed in the pager but not my meal. It's really unbalanced to see so many pagers ringing but mine wasn't. Well, when I came back with some more food, I couldn't find my pager!!! I was frightened. I guessed that someone once sat on my chair and mistook and finally I was right. Anyway, I got back my pager later and I got to know how to use it when it responded more quickly, 'coz I'd like to make experiment and if the result was not swift to know, the experiment would fail.*
>
> *Then I had installed dozens of email address of my friends. I enjoyed sending them email before going to bed using "send to all".*
>
> *-Chinese Junior Summit Delegate*

Teaching 100 people how to use a new technology can be a challenge, more so when dealing with the logistics of language; there were 6 official languages at the summit. Also, due to scheduling constraints, it was necessary to deploy the pagers in a half hour period.

It was decided that the distribution of pagers and demonstration of their use would be integrated into the kick-off event for Junior Summit. The delegates had never met face-to-face at that point. Rather than distributing the pagers in an orderly fashion, the pagers were handed out randomly to the delegates as they entered the room. No instructions or explanations were given. Within five minutes the delegates were learning the operations of the pagers and teaching their peers. The researchers then announced that a mistake had been made in handing out the pagers and that they would have to figure out a way to get the pagers to the rightful owner. A brief, formal introduction to sending and receiving pages was given to them at that time. After 15 minutes of cheerful chaos, the delegates had everything mostly sorted out. One crafty delegate decided to employ the public address system as a means of organizing his peers. One misunderstanding occurred when one of the Chinese delegates refused to hand over the pager he had been entrusted. A translator helped resolve the situation.

The pager deployment was so successful that 600 messages had queued up from the dinner, and overloaded the Canard servers, which were unaccustomed to such intense use. It turned out that the researchers had left a debugging trace on the queues, which resulted in messages being processed slowly. Once this was detected, messages flowed fluidly. The delegates then sent out an average of fifteen hundred messages a day (one a minute!) during the rest of the summit.

Figure 4-10: PageWriter 2000 custom applications for Junior Summit. The leftmost image shows the interface used for delegates to indicate which room they were located within the Media Laboratory or the hotel. The center image shows a confirmation screen verifies that the delegate indicated the correct room. The rightmost image shows the interface the delegates used to locate other delegates.

Specialized Pager Applications

Although the Canard system models communication as a single, general-purpose application, it is sometimes useful to develop specialized front-ends to solve community problems.

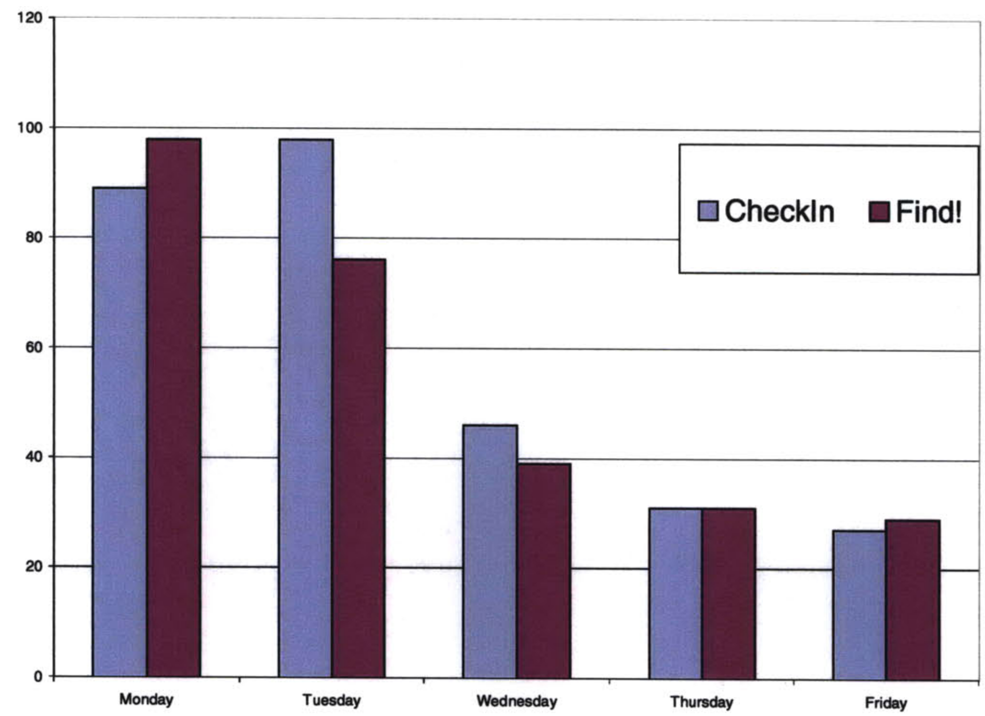


Figure 4-11: Daily usage of CheckIn and Find! applications during Junior Summit week. At the beginning of the week the use of these applications represented 13% of all messages sent by delegates from their pagers. By the end of the week the use of the applications dropped down to 6% of outbound messages.

In the case of Junior Summit, a number of applications were developed for the needs of the delegates. Since the delegates had been communicating via an online forum prior to the summit, it was decided to modify the Canard addressing scheme to mimic that of the forum. Address databases were set up such that a delegate could send messages addressed by full name, username, discussion topic number (to communicate back to the online communities), and their "action plan" groups. Also, the pager's address book application was modified by Motorola to take advantages of these new address databases, thereby simplifying adding entries.

It was suggested that the pagers would be useful for tracking the children attendees. Two front-end applications for the PageWriter 2000 were developed for this purpose. The first application, CheckIn (see Figure 4-10), allowed a child to indicate to Canard their current location on a schematic floor-plan of the Media Laboratory and the hotel. Since the delegates were only using a small number of locations (ten at the Media Laboratory and four at the hotel), it was possible to design an adequate graphical interface on the relatively low-resolution display of the pager. The CheckIn program sent a message via electronic mail with the location of the child to a Canard server that maintained and updated a delegate location database. A "nag" message was sent out throughout the day to remind the delegates to use the CheckIn application at the beginning of each session.

Many delegates voluntarily used the programs at the beginning of the summit (see Figure 4-12) but the number dwindled because of the frustration of not finding other people. The

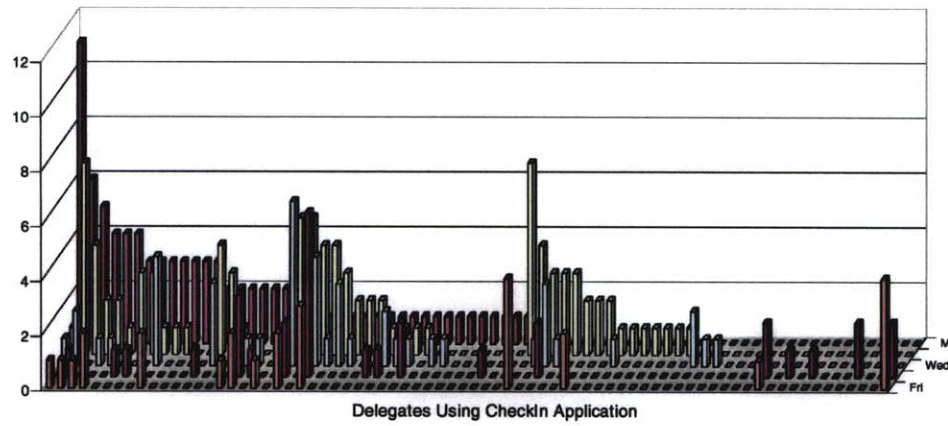


Figure 4-12: 77% Junior Summit delegates used the CheckIn program at during the week. Most occurred at the beginning of the week but usage dropped off during the week.

adults running the event did not encourage the delegates to check in when they changed locations causing the database to be out of date. The delegates turned to sending messages (see Figure 4-13) asking where someone was located. Often this could occur from within the same room that the delegates were occupying, illustrating the lack of addressability of the CheckIn program. One could not indicated one's precise location within room.

Effects of Supervised Operations

Normally, the Canard communication infrastructure runs unsupervised for long periods of time. For the Junior Summit, extra care was taken to supervise the system on an almost 24-hour schedule. The researchers sent broadcast messages to all the delegates at 11pm and 7am to notify them of "official operations" status. The delegates' messaging traffic would usually increase significantly with the "wake up" message, as if it were a signal that it was acceptable to send messages to one another. Given that the delegates came from 24 different time zones, getting into a synchronous time frame was a challenge. The Canard system status broadcast served as a social cue.

Personalized News

The researchers have strong roots in personalized news, so it made sense to handle some of the news needs of the summit. The Associated Press made their news feeds available in French, Spanish, English, German, and Swedish. Personalized news profiles, similar

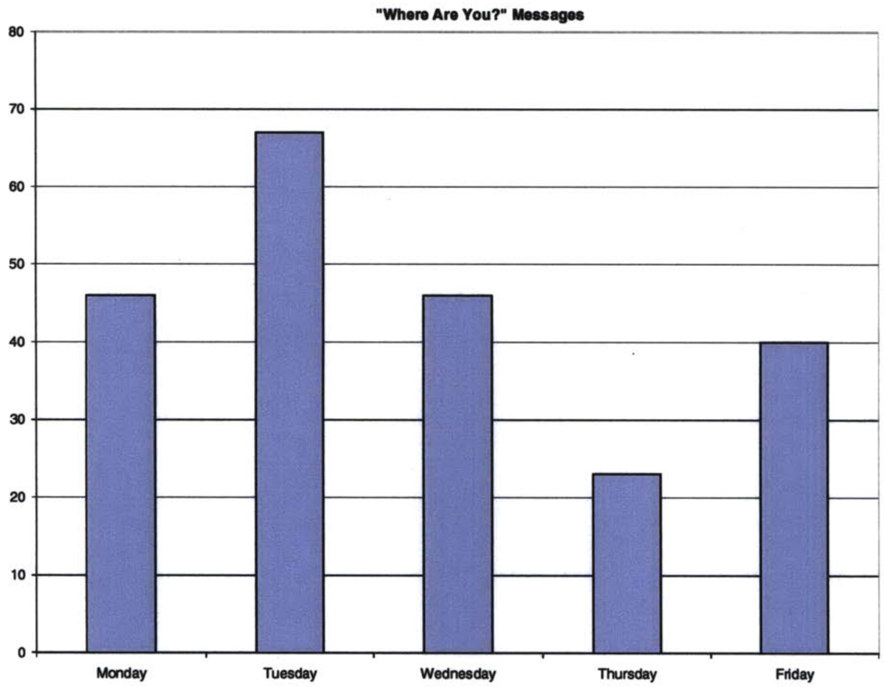


Figure 4-13: Roughly half of the Junior Summit delegates sent messages explicitly seeking the location of someone during the week. It raised the question whether or not the CheckIn! application was well suited to the delegate's needs. Explicitly asking a fellow delegates location does not require a database to be accurate.

to those used in *FishWrap* [CMS95], were created for each of the delegates, listing their hometowns and native language. A news summary was sent to their pagers in their native language every day at noon. In addition, if there was news from their hometowns, it was sent to their pagers, regardless of the language it was written in. The original intent was to pipe these messages through automated translation software, but resources to do this were unavailable at the time the summit. Had this lack of automated translation software been known sooner, the Canard message processing server could have translated messages while they were enroute to the pagers. Regardless, the delegates were grateful for the news summaries.

WhiteBoard and LED Signs

Another facet of the Canard infrastructure that was a potential benefit for the delegates and the online forums was the use of the electronic whiteboard and LED sign devices. This technology has the ability to record meeting notes and instantaneously communicate them to the online forum in graphical form, as well as keep an archive for later use by the delegates. In addition, members of the online forum would be able to send messages to the delegates in the rooms where the deliberations were taking place.

The organizers chose not to provide this link to the online forums, arguing that it violated their "lowest common denominator" policy. Text sent via electronic mail was the only acceptable form of communication to the organizers. This decision lead to the disabling of a communication channel that perhaps would have further engaged the online forum participants. Activity on the online forum was greatly diminished during and after the summit. Delegates sent 140 messages the first day of the conference (see Figure 4-16) and dropped down to 43 the last day.

WhiteBoard Application

A fortuitous use of the electronic whiteboards arose when one was set aside for the Chinese delegates to use. Motorola had acquired five one-way pagers that were capable of displaying Chinese text. Unfortunately these pagers were on a separate network. Also, there was no local know-how to convert messages into the Chinese character set necessary for display on these pagers. However, the whiteboards provided a convenient mechanism for authoring in any language because, in the Canard system, the graphical image is sent to the pagers without the need of a specialized system. Using this system meant that the Chinese delegates did not need to carry an additional device and did not need to learn a special authoring tool (that had not been written) to send messages. The drawback was that messages were sent to the whole group, not to individuals. Fortunately, the same process that converts images from the electronic whiteboard is able to accept PostScript from any device on the network. The Chinese delegates used their word processors to send messages to print queues that would route the image to individual pagers (see figure 4-14).

> *I was most impressed when Mr. Chesnais helped Chinese students to setup the image system. Using the whiteboard everywhere, we could send Chinese as images. Thus, the problems of input and reading are solved at once. The first message we sent is one asking my friends to have dinner in the hall. I wrote the message and added a smile face beneath it. Wonderfully, I got it minutes later. After decompressing, the image is on the screen.*
>
> *–Chinese Junior Summit Delegate*

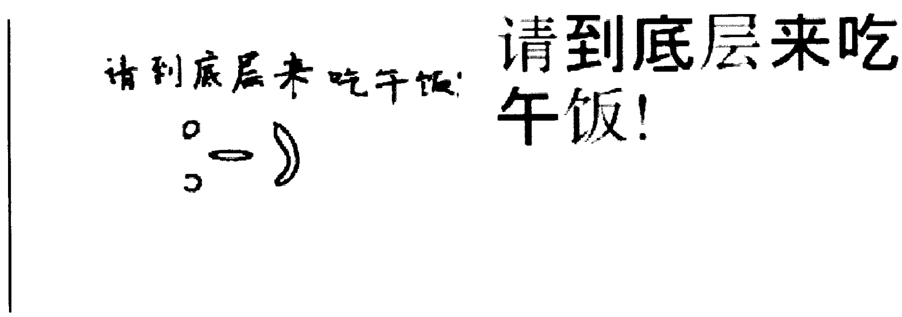


Figure 4-14: A handwritten message in Chinese is captured from an electronic whiteboard and converted into a bitmap suitable for display on the Web or transmitted over-the-air to pagers. Left: a handwritten message from an electronic whiteboard. Right: a message from a word processor

4.3.4 Effort

Financial Barriers

Participating in the Junior Summit online forum proved to be an expensive endeavor for some of the delegates. In France, telecommunications technology is a closely held monopoly.

> *And as you know, France Telecom is 'fighting' cable, so connection is very expensive–our first two months at Junior Summit gave my mother a telephone bill of almost 7 000FF!!! (roughly 1,200US$)*
> *-A French Delegate*

Figure 4-15: Junior Summit delegates used pagers to communicate with other delegates and online forum members, even while in transit on buses. photo: Jacqueline from Canada.

Pocket E-mail or Pocket Chat?

> *Hello Pascal, my favorite e-mail was the one that my Dad sent me on my pager. My Dad just got e-mail set up for the first time at his office while I was at MIT.*
>
> *Every morning at MIT a boy from Norway would send me a wake up e-mail. The pager beeped when there was a new e-mail, so when he sent a good morning e-mail it beeped and so it was a kind of wake up e-mail. Lots of people were asking me the first evening to put my e-mail address on their pagers, so the first morning when I got this wake up e-mail I was not sure who it was from, but we became friends.*

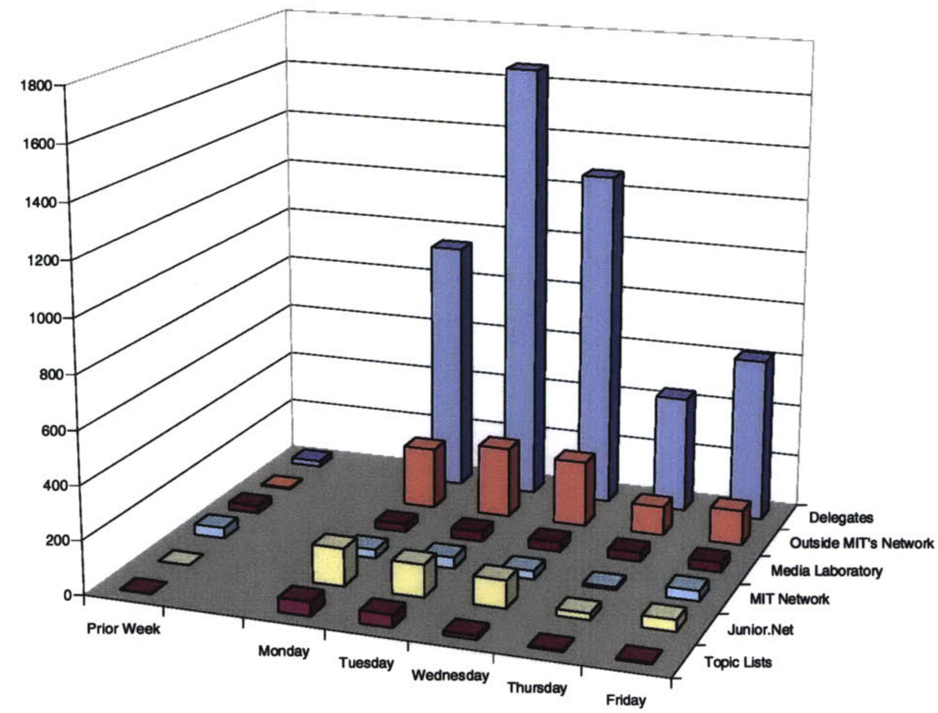


Figure 4-16: Outbound messages from the pagers were primarily directed to other delegates using the Canard system. Delegates uses the pagers to send messages to friends and family outside of the MIT network. Delegates sent messages to Junior Summit online forums (Topic Lists) and addresses (Junior.net), but the number of the messages dropped off during the week. Messages were addressed to MIT and Media Laboratory addresses as well.

> *Also, I sent reports home to my school. The school secretary read my report describing my pager and decided to try tracking my mom down at MIT with me to let her know her volunteer help was needed the first day we were getting back.*
>
> *Also, I sent e-mail to my Junior Summit topic group and homeroom. My first pager e-mails worked OK so I assumed they all would and learned the hard way not to assume, as some later ones did not work. Luckily the messages I sent were still on the pager so I was able to copy them at home.*
>
> *–Junior Summit Delegate*

Pagers were perceived by the delegates as "pocket email" devices. During the summit the delegates sent thousands of messages (see Figure 4-16) from the pagers. Some of the delegates were surprised that they could send messages from the pagers to friends on the Internet without having to do anything special. On the first day a quarter of the messages were sent to addresses outside the Media Laboratory. Judging by the lack of comment most of the delegates found this facility natural. By far, most (65%) of the message traffic from the pagers were destined to other delegates. During the summit, adult counselors were with the delegates. They quickly realized that the delegates were immersed in this communication system, and asked if they could be included. Additional pagers were made available to them, and the results were immediate – the counselors found themselves bombarded with questions from the delegates around the clock. They found this very convenient, since it allowed them to keep in touch and feel like part of the delegate's community. The other adults in the summit were the moderators. By the second day, they were begging for units to stay in touch with the delegates. Unfortunately there were not enough units to equip them all.

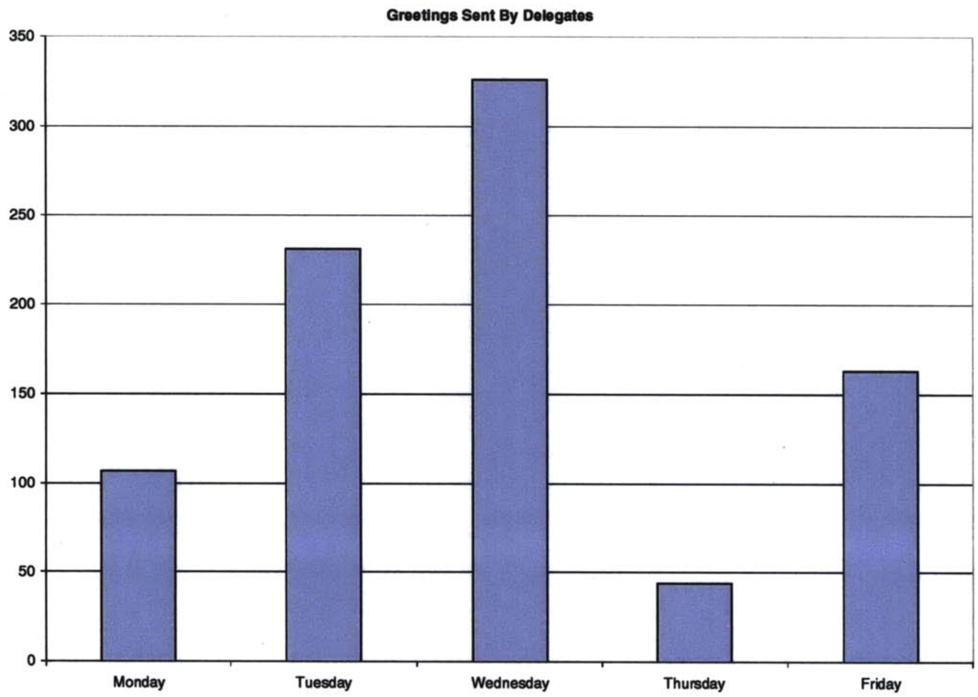


Figure 4-17: Junior Summit sent a lot of messages that were just salutations (i.e., "good morning", "hi", "hola!") with no other substance to the messages. As the delegates got to know one another during the week the frequency of greetings increased.

Salutations of the form "hi", "Good morning" were frequent (over 25% of messages to delegates on Wednesday, 17% overall) during Junior summit week (see figure 4-18). These messages contained no other content. They seemed to serve the same social function as one would find in a chat forum[Mat99]. As can be expected with children who are spending a week together, a lot of messages sent with the Canard system were of a social nature. A common subject of messages dealt with parties (see figure4-19) at the hotel.

4.3.5 Constructions

LED Signs and WhiteBoards

The LED signs and electronic whiteboards were present in five of the meeting rooms. The whiteboards were specially designed with counter weights (Figure 4-20) to allow the kids to adjust the board to their height. Unfortunately, the organizers of the event neglected to familiarize the moderators and counselors with their use. Two were either not set up properly or disconnected by the adult session moderators. The result was that accurate

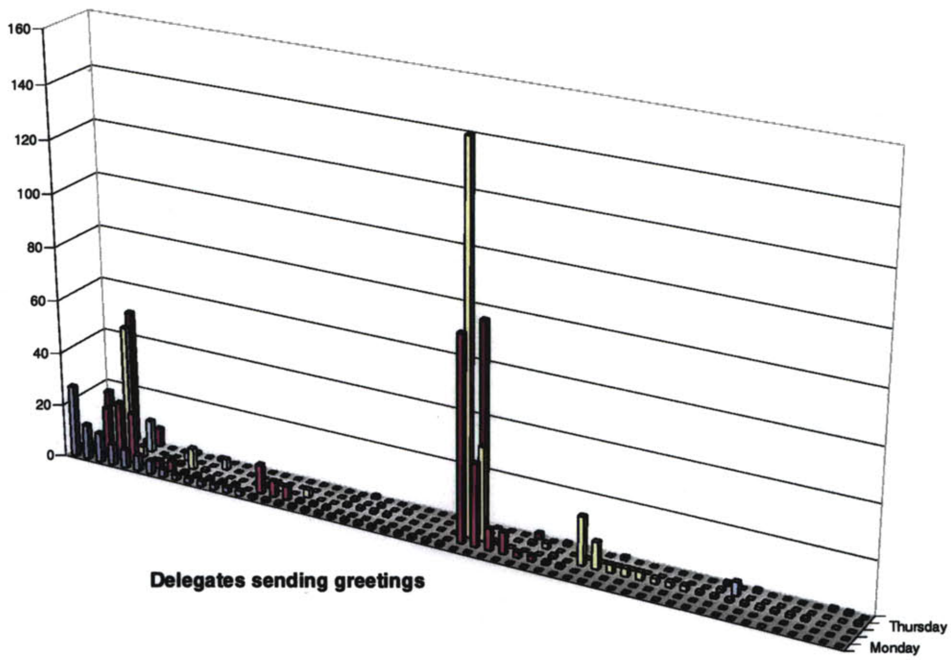


Figure 4-18: A small number of delegates were responsible for sending a lot of greetings.

records of the discussions were not kept. Perhaps more important, the goal of using the whiteboards as a mechanism for communicating with the online forum participants was not met. These virtual participants were not engaged with the activities at the Media Laboratory and were largely ignored by the delegates during the summit.

Of the remaining electronic whiteboard and LED sign combinations, the most utilized was the one in the auditorium. This device was given an early introduction to the delegates. The most important feature for them was the ability to send a message to the LED display from the pagers. This resulted in nearly 500 messages being sent (see figure 4-21) to it the first day. The messages were initially sent anonymously. These messages were mostly from delegates having fun with technology. The next day, the messages were displayed with the author's name. This lead to some delegates blushing and the content of the messages shifted to more serious issues.

During the summit the delegates were frustrated by the overly structured schedule that did not allow them much opportunities to explore issues on their own. Early in the week the delegates organized a revolution using the electronic whiteboard in the auditorium to take notes. These notes were then printed, photocopied, and then distributed to the other delegates. The revolution ended when the adults made concessions.

> *It was nice as an idea but I think that it was not used for any serious purpose.They were used just for fun.They helped the user to have it display the sender of the messages.The most usual message I noticed was just "hello".The LED signs were closed after a while as they caused annoyance because all were*

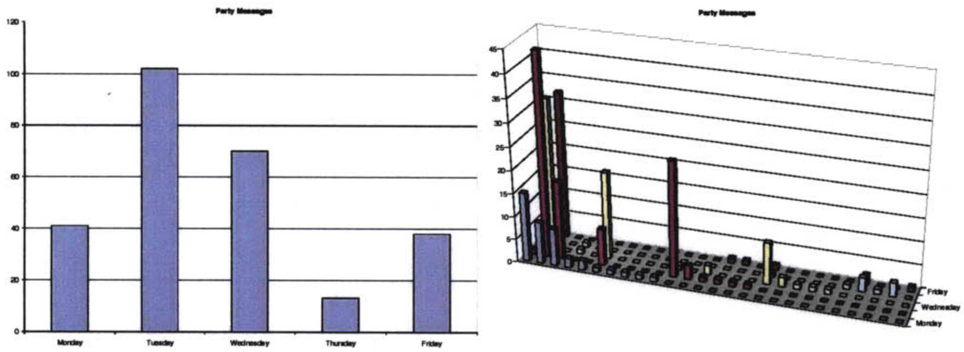


Figure 4-19: Junior Summit organized parties with the pagers. The messages were sent by a small number of socially active delegates.

> *using them and Justine was not heard when she was speaking. –Junior Summit Delegate*
>
> *It was funny when we used to broadcast our parties, our names, and I made up the message that made some of the delegates, (or maybe one of the techs, when they saw it,), think that the LED's memory was going to fail and will crash at any moment, sure it was cool. –Another Delegate*

Although the delegates really enjoyed this facility and sometimes used it to communicate their frustrations with their peers or the speaker, the adult organizers did not appreciate this candid and very public channel of communication. Ultimately, the adults ordered the delegates not to use their pagers during speeches. When that order was ignored, the adults disabled the device by yanking out the cables. This experience illustrates how the adults were unwilling to adapt to a new form of communication etiquette, or to relinquish control over communication.

4.3.6 Reflections

The delegates took to the new technology like wildfire. It was a natural transition from their online forum to the collocated discussion. Given their local mobility needs, the two-way pagers were a good fit as a primary telecommunication-communication device, certainly preferable to forcing the delegates to use computer clusters. The delegates enjoyed experimenting with new technologies and adapted rapidly to the potential offered by them. The flexible nature of the infrastructure allowed the support team to adapt tools to unplanned challenges, such as the creation of a channel of communication for the Chinese delegates.

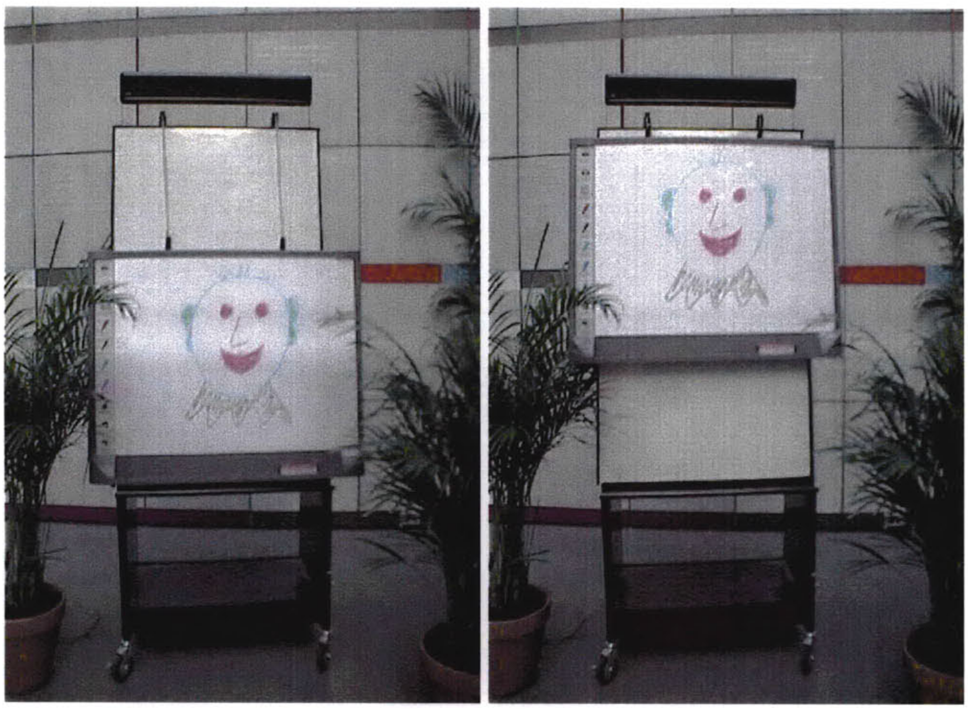
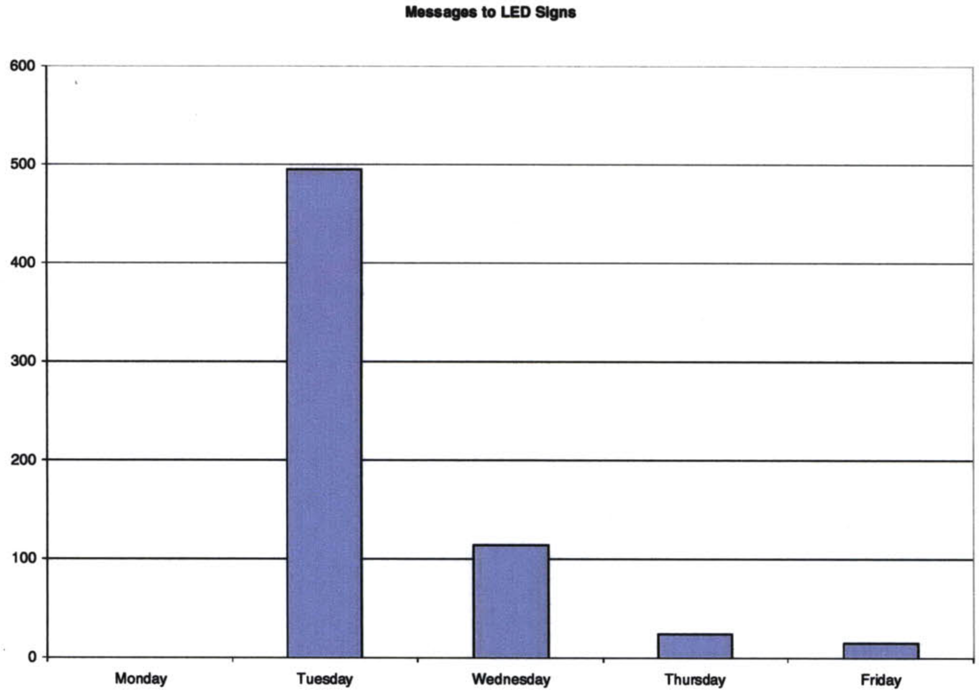


Figure 4-20: By using a system of counterweights, the whiteboards could be adjusted to the delegates's height. Lashaun Collier designed this installation specifically for the Junior Summit.



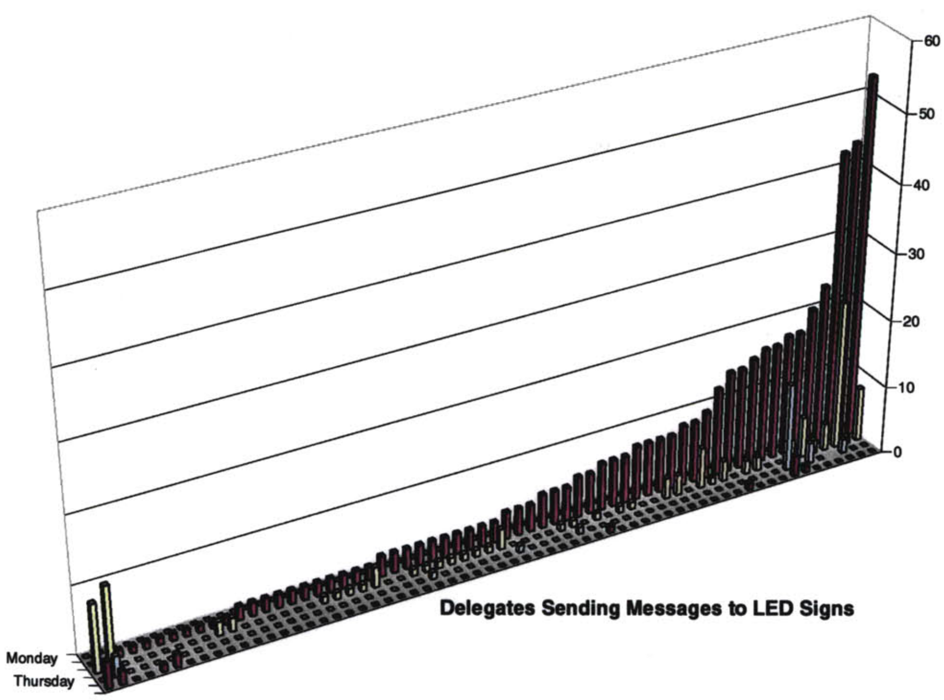


Figure 4-21: Messages to LED signs over the whiteboard were very popular from the time they were made available. Then the usage dropped off when the delegates were told not to send messages to them, and they were subsequently unplugged from the system. Virtually every delegate sent messages to the LED signs.

4.4 today@mit.edu: Freshman Campus Calendar Project

4.4.1 Overview

The project started with high hopes of being the central calendar used by all MIT groups and activity planners.

-today@mit Team Member

Five weeks into the fall semester of 1997, an MIT freshman died as a result of consuming too much alcohol at a fraternity event. What followed on MIT's campus was a major introspection. The campus was divided about what should be done to prevent similar tragedies. A common complaint among students was the lack of knowledge of what social activities were available as alternatives to the traditional college drinking party.

As a class project, an MIT freshman seminar undertook the task of creating a campus-wide calendar. They were deeply affected by the death and wished to help the MIT community in some positive fashion. Given their advisors were developing applications surrounding wireless communication systems, they proposed equipping a number of students (themselves) with two-way pagers so they could collect information about events on campus from bulletin boards and enter them on-the-spot into a centralized campus calendar database.

At the time, there was only one active online calendar hosted by the campus computer group, SIPB (Student Information Processing Board). This calendar system had some interesting features and an easy to use interface. But though the calendar was easy to access via the world wide web, it was seldom used (see figure 4-22). The freshman seminar group believed they could do a better job.

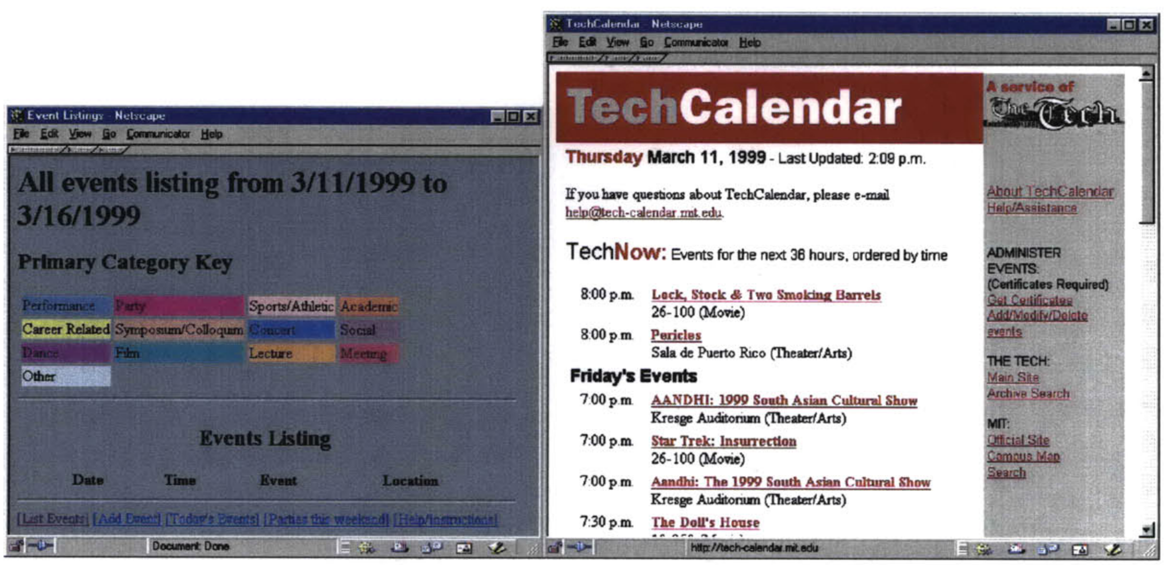


Figure 4-22: On the left is the SIPB calendar of events. No entries were present for the next five days and only thirteen events were listed for the year. On the right is The Tech's calendar. It is full of events that appear both online and in their printed editions.

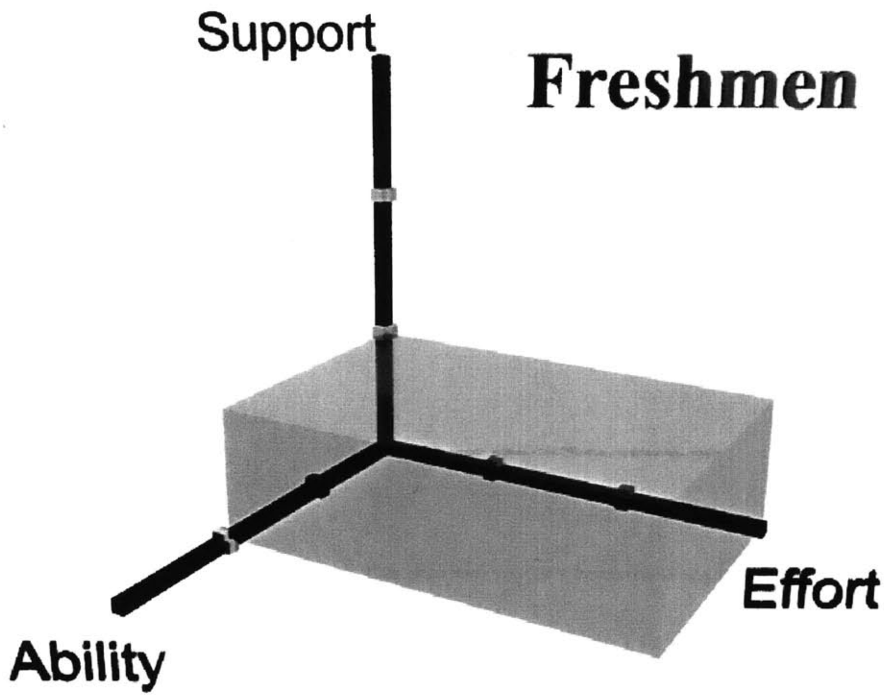


Figure 4-23: Freshman Campus Calendar project members had ample abilities to implement their project. No external support was provided to help this group. As a group they were willing to invest significant amount of effort to overcome the complexities of the technology.

4.4.2 Ability

> *Our assumptions about the usefulness of and complexity therein needed may have set us on the wrong path from the beginning. We focused on using too many pieces of proprietary software that only a few of us could use.*
>
> *-today@mit Team Member*

The group had a good mix of talent. Some had strong backgrounds in programming while others had good design skills. One student in particular had professional experience developing dynamic web servers. None in the group had experience managing a software project, but all had a keen desire to see the project succeed.

4.4.3 Support

At the time the freshman embarked on their project, the Media Laboratory received the PageWriter 2000 two-way pagers. These pagers allow graphical displays and they support a programming environment can handle multiple text fields for structured data entry. There were a sufficient number of pagers available to equip the freshmen for their project. The freshmen also had access to a Sun workstation web server and an IBM RISC Station 6000 database server. Beyond access to these hardware resources, the freshman group would have to be self-reliant.

4.4.4 Effort

The result was a half-hearted effort because the group was so divided on lines of experience and technical ability.
–today@mit Team Member

The freshmen worked on the project for approximately one year. They had grand ideas and developed some compelling prototypes of the interface. However, when it came time for implementation, the project ground to a halt. They were overwhelmed by engineering details.

The community hadn't organized itself in a way that encouraged or enabled all participants to play a role in the development. Rather than taking an incremental approach, the experts within the group focused on esoteric issues surrounding choice of a software environment. The programming novices of the group felt they could not contribute and subsequently dropped out of the project. Those who remained with the project were dependent on the expertise of one student who alone possessed the programming skills necessary to exploit the chosen proprietary software package. But this student wasn't able to see the development process through on his own.

A great deal of effort was placed in developing data gathering applications for the PageWriter 2000, which turned out to be useful for other projects. A web server was launched to gather information, but unfortunately back-end database issues never got resolved.

4.4.5 Constructions

The group did devise a general architecture for their project whereby entries could be submitted through a number of communication channels. The two channels they focused on were a World Wide Web and a two-way pager interface.

A WWW server was set up by the group to host the calendar project. From a browser, one could submit events to a central calendar service or browse on-campus activities. The WWW server was also designed to allow remote administration of the databases by event organizers. A custom two-way pager interface was written by one member of the group. The interface allowed the group members to communicate event information to the calendar server via an electronic mail gateway. The same interface could also be used to populate other on-campus calendar databases.

A calendar database server was specified by the group. They decided to use a commercial database server that was used to drive e-commerce web sites on the Internet. The problem was that only one person in the group knew how to use this complex package but never had the time to finish that aspect of the project. Without a server, the system, as a whole, did not function.

4.4.6 Reflections

The project failed because too much effort was placed on the engineering aspects of the project. As a result, those with little skills in software programming felt left out. Also, no tools were made available for the novices to prototype their ideas.

Had the freshman tried the same project a year later, the Canard tools would have been more friendly to the non-experts in the group. One of the researchers in the project assembled the tools together to simply author a community calendar. The tools are there,

Figure 4-24: The freshmen designed a pager interface with which to submit campus events to a calendar database server.

but the question of whether or not the services will be used still exists. Looking at the SIPB calendar, one can understand why it is seldom used. It is not part of anyone's daily information environment. Entering information into the SIPB calendar is only accessible through one interface. No other manifestations are available. Had the SIPB group printed their calendar every Friday morning for displaying campus activities in central locations across campus, people would probably have submitted more events. The Tech, one of MIT's student newspapers, took this approach by gathering calendar events on their web site and printing them in their Tuesday and Friday editions (see Figure 4-22).

Had the freshman ignored the issue of creating their own database and instead focused on the distribution of the event information to multiple campus calendar databases, they would have probably been more successful. They would have benefited from the richness of existing set of on-campus databases that were already being collected by other groups. Rather than being viewed as a fringe group at MIT, they would have probably earned praise from other groups at MIT for adding to the success of these groups' calendar services.

With Canard tools, it is easy to connect to web-based services. For accessing The Tech's calendar from a pager, it is only necessary to know the URL of the search engine. Creating an interface for submitting events is a bit harder, but with HTML form support on the pager, a person can submit structured information (see Figure 4-25).

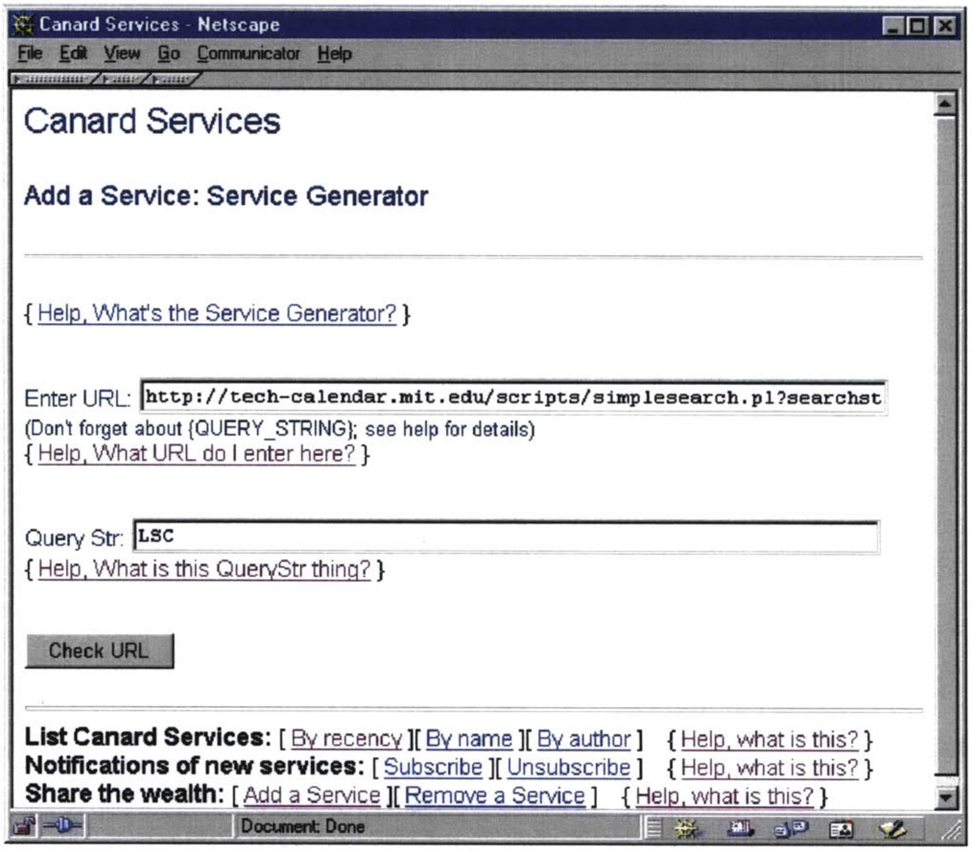


Figure 4-25: Canard service generator form for accessing The Tech's calendar search engine. One only needs to enter the URL of the desired service and a test query to make sure it works. The user can later tailor a filter to trim excess lines.

4.5 Speech Group

4.5.1 Overview

The Speech Group at the Media Laboratory has been researching multi-modal messaging systems for almost two decades[Sch93][MS96]. As a result, their goals and needs are much different than the other test groups. Canard provided them with an unlimited campus-wide two-way paging service and the opportunity to gather information from the system that is normally unavailable from commercial service providers.

4.5.2 Ability

All members of the Speech Group have extensive programming abilities and have developed communication technology applications. Adoption and integration of new technologies comes easily to this group.

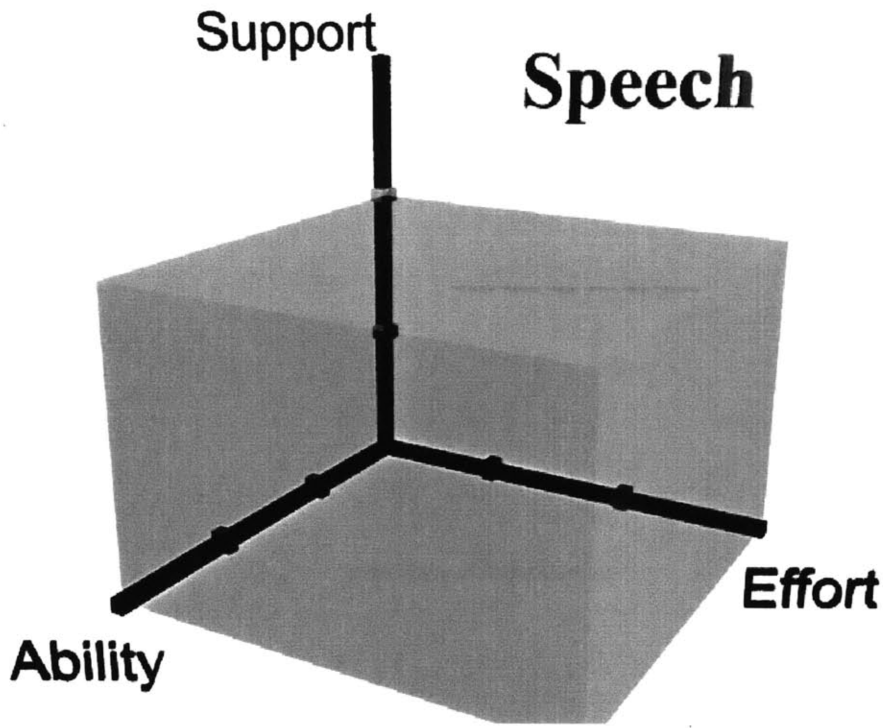


Figure 4-26: Speech Group group had the expert ability to design their own messaging system. Canard provided spot support to this group in order to integrate its unique two-way paging device and message tracking to their system. The Speech Group only needed to invest a little effort to overcome the complexity of adapting to the new technology.

4.5.3 Support

This group had minimal need for formal user support from the researchers. They were mostly self-supporting and able to determine how most aspects of the system worked. The group's director and its graduate students were all provided with two-way pagers. The undergraduates of the group were not provided with pagers because they were mostly transients during the time frame of this project.

Unlike other service providers, information about the disposition of messages is readily available in the Canard system. For example, two-way pagers register their presence with the network, when they come into range, when have been turned on, and whether or not a message has been received. Canard has not made this information readily available. Modifications to the system were made by the Canard team to allow scripts to be executed when a pager comes into range. This modification allowed the Speech Group members to use Canard to trigger inferences about location and to adjust the user profile accordingly[Mar98a].

4.5.4 Effort

The Canard system provides some features that the Speech Group could not obtain through other paging systems. Two such features are the status of how messages are being processed by the system and the notification of pagers coming into range of the system. These are critical features that allowed the group to demonstrate new functionality in their own messaging system [Mar98a].

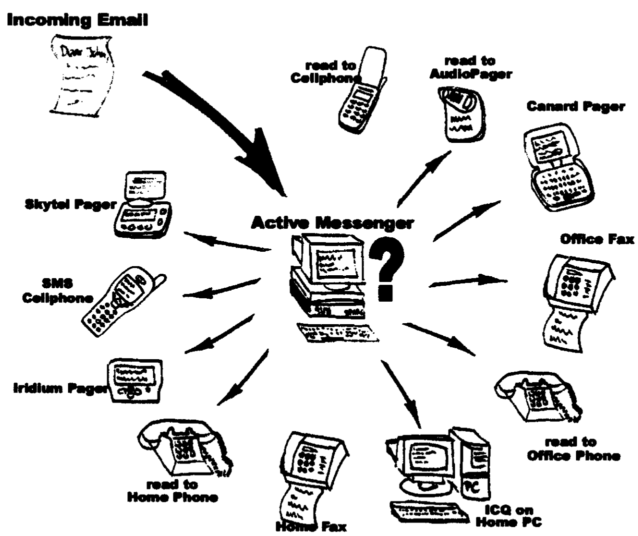


Figure 4-27: Active Messenger is able to route messages to many different communication channels including Canard.

4.5.5 Constructions

Interfacing to Canard

The Speech Group uses electronic mail as their primary means of exchanging data across platforms. Through the use of electronic mail gateways, they also have access to data on their desktops. By using the electronic mail interface, they were able to integrate Canard into their own systems with little effort. Getting access to some aspects of Canard, e.g., the delivery status of messages, was more complex. This required web access libraries which were not part of their normal computation environment. Given the benefit of knowing the status of a message's delivery, they were willing to add the required low-level functionality to their system.

Inrange Notification

A pager's ability to notify the Canard system when it is inrange is of particular interest to the Speech Group. No other system offers them that capability. However, there did not exist a simple means for the Canard system to communicate this information to other systems. In response to a Speech Group request, the Canard developers implemented an in-range notification system called "Quack". When a pager comes into range Canard sends electronic mail to a specified electronic mail address.

The Speech Group routes this electronic mail to a subsystem of own that retrieves the last half-hour of unread messages from a user's mailbox and sends them to their pager. This is particularly useful for commuters to campus.

Mangled Message Notification

The paging system used by Canard was designed to be a demonstration unit by Motorola. It was never intended to be used as a running system. In particular, there are problems with the messages that originate from the pagers. These messages are sometimes be truncated by the system. However, the pagers are capable of appending a line of text as a signature to every outbound messages. The Speech Group exploited this feature to build a system to verify the integrity of outbound messages.

One of the Speech Group graduate students wrote a subsystem that would examine each of the messages he sent for this signature line. If it was not present, it would send him a message indicating that his message was probably truncated. This capability was made possible by the Speech Group's system. Every outbound message is routed to their system where they augment the message, if necessary. This approach was derived from their earlier telephone interface research where it was difficult to compose messages solely with the touch-pad. By using a centralized system, they can augment their messages with contextual knowledge about both the author of the message and the device that they are using.

4.5.6 Reflections

The Speech Group represented the ideal community. The group members had sufficient ability to use the Canard platform without it being a hardship. The Canard researchers provided the skilled support when needed, allowing the system to be well-adapted to the Speech Group's communication environment. The group was willing to invest the effort needed to adapt the Canard environment to their own. The effort required, however, was minimal. This was mostly due to the fact that many of Canard's tools were inspired by the Speech Group and as such were well suited to their needs.

An indication of how much value the Canard paging network provided the graduate students of the group was the fact that two students moved to an apartment that would be in range of the campus paging and computer networks. One of the students, Stefan Marti, wrote a paper [Mar98b] examining the social aspects of two-way paging.

4.6 The Cube

4.6.1 Overview

In September 1998, three established research groups that had been working in different areas of the Media Laboratory were all moved to a new temporary (five years) facility called The Cube. It is a vast, two-story room. The research groups, Electronic Publishing, Explanation Architecture, and Epistemology and Learning, had only a limited history of collaboration in the past. One goal of housing the three groups together was that more inter-group projects would develop. In designing the work space, it was thought that mobile information technologies might play a vital role in life in The Cube.

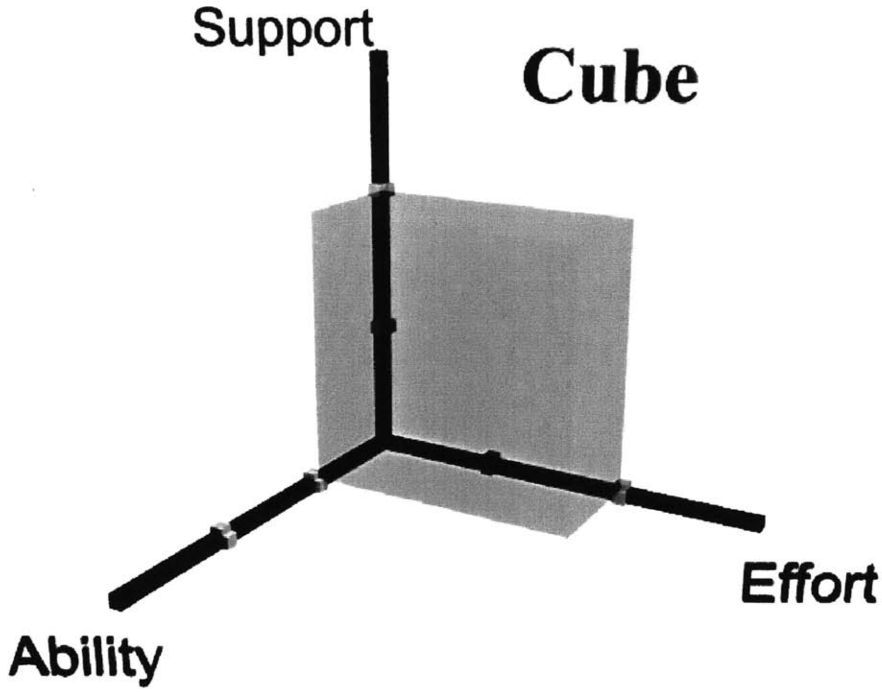


Figure 4-28: The Cube needed few skills to use the Canard system. Spot intervention was provided to reconfigure the system to allow undergraduates to benefit from the opportunities of the wireless system.

4.6.2 Ability

There are twenty graduate students, three administrative assistants, two faculty members, and one senior scientist occupying offices in the space. A score of undergraduates also work in a central common area while they are not in class. Most of The Cube occupants have extensive programming experience. All of The Cube occupants use computers for electronic mail and web activities in their daily activities.

4.6.3 Support

Given the unique duplex architecture of the space, special care was made to support the communication needs of The Cube's occupants. Although each office had a telephone, three phones were installed in the common area to encourage graduate students to work there alongside the undergraduates. Each of the graduate students phone lines appeared on the common area phones. In addition, laptops with wireless local-area-network adapters were provided for the students in order to maintain an online computational environment no matter where the students were in the building. For out-of-the-building communication connectivity, everyone in The Cube was provided with a two-way pager on the Canard system. Given that Canard was being developed in The Cube, all occupants had essentially unlimited access to the researchers.

4.6.4 Effort

An underlying assumption in planning The Cube communication infrastructure was that there was a strong need for local mobility[BB96]. In actuality, the graduate students do not routinely leave their offices and few of them frequent the common area.

The Cube provides either an unobstructed view between offices or the offices are adjacent to each other. As a result of this layout, the graduate students always have either visual or audible cues as to one another's presence with whom one needs to collaborate.

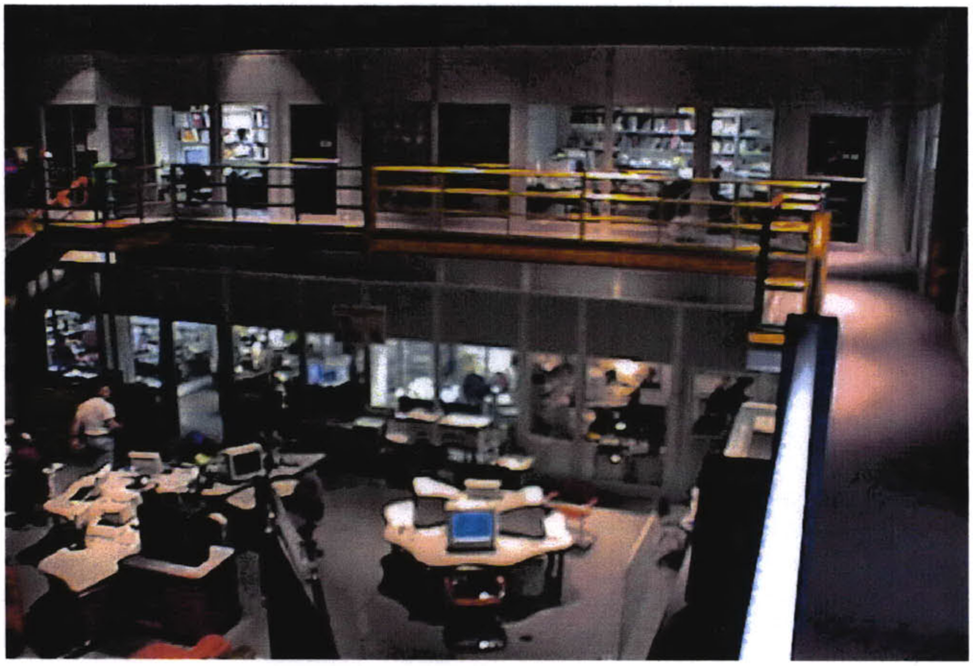


Figure 4-29: Offices in the Cube have a virtually unobstructed view of one another.

I was initially very excited by the pager, and proudly attached it to my growing utility belt. I remember the novelty of sending emails from the other side of campus while sitting outside having lunch. My officemate and I would shoot pager mails back and forth from each other through the day even though we were right next to each other, and sometimes we would send messages back and forth into the night for fun.

Then the realities of graduate school life at the media lab began to set in. Working in an extremely networked environment day after day began to break down the novelty of communication with a pager; in fact, it became a nuisance to use the small keyboard over the fullsize keyboard directly in front of me. The volume of regular email received in an average day also grew extremely high, and my priorities for which emails needed to be forwarded to a pager and which did not changed on a regular basis. Taking the time to set up message forwarding in my email application whenever my priorities changed grew tedious.

However, there were times when the pagers tied my extended lab community together. My urops are not here on a regular basis in the lab, and getting infor-

mation from them with brief notice is difficult. I used the canard system a few times to reach them directly, often to set up a telephone conversation, with great success.

–Cube Graduate Student

Although the graduate students did not frequently move far away from their offices, the undergraduate students working for them were mobile. As a result, two-way pagers were given to the undergraduates. The benefits were immediately apparent.

I know I've only had my pager for a few weeks, but I can honestly say that I'm addicted to it. I've worn a pager in some form or another for a few years, and it's nice to have a way for people to contact me. The bidirectionality of the 2-way pager transforms paging into a communications channel, instead of a digital leash. I always hated being paged in the middle of a class, or during a movie because there is no way to respond without physically leaving (obviously pulling out a cell phone to answer a page in the middle of one of the new "Star Wars" films isn't one of the ways to win friends and influence people). Now I've got a way to reply to a page merely by sacrificing a bit of attention instead of physical presence. The ability to send messages makes using a pager less like being one of Pavlov's dogs (where you become trained to salivate at the sound of the bell, or beep, or buzz...).

Another thing I like is the migration of my e-mail from a hard-wired medium to a device that fits on my belt. The pagers have done for e-mail what the cordless phone did for telephony – suddenly you're not chained to your desk waiting for an important message. I also like the idea of increasing the number of ways to receive a message without increasing the number of "contact points" (phone numbers, email addresses, etc.). I've already got a phone number for my dorm room, a phone number at home (in NH) a pager number, a cell phone number, at least 6 email addresses – the last thing I need is another number or @ symbol associated with my name.

–a very satified undergraduate

Due to the enthusiasm of the above undergraduate student, a graduate student solicited the other graduate students for their pagers. Since they were not using them they were available so she could distribute them to her other undergraduate research collaborators.

I've never wanted a pager. I didn't want to be "followed around" by work that I wanted to leave at the office.

I work too hard as is, and I figured, when I take a break - I don't want to be forced to worry about work.

For a recent project, my view has changed. Turns out, the four members of my research team never seem to be in the same building at the same time. They are at home, in class, in their offices, en route, at the movies or grocery store.

With a tight deadline and a changeable production schedule, I sometimes need to communicate with them immediately.

With a project which necessitates that team members work simultaneously yet individually, and collaboratively: sharing their thinking, team members need to communicate with each other immediately, as well.

We all need to schedule and reschedule, ask, answer, and clarify questions, and coordinate our thinking (and computer code). I don't have time for individual team members to go barking up the wrong tree due to lack of communication.

I'm really proud of my work. I think it'll be terrific. But my time is short in which to make it a reality. If I want to do a good job - I'm willing to be interrupted by a pager...

While I still don't ever WANT a pager. And won't use one once this project is over, this nay sayer will admit that she can now see the definite value in such a communication tool!

–Cube Graduate Student

4.6.5 Constructions

Movie Service

One of the graduate students hypothesized that two-way pagers might be useful in the activity of choosing a movie to see. he envisioned Canard as a means of retrieving reviews and screening times. True to the spirit of the lab, this student felt that having a mechanism for peer reviews of movies was important. He wrote a web-based service where contributions can be submitted online using a desktop computer and the results can be readily retrieved using pagers when and where the information was needed.

It proved to be easy to build this movie review system with Canard, but the system has seen limited use. This is due at least in part to the fact that few people within the community know of the existence of the service. This is a general problem for Canard services. There is no informal mechanism for notifying the community of Canard users about available services and applications. In an attempt to solve this problem, a student has created a directory service for Canard services. The directory service, in addition to providing pointers to available services, also allow community members to run web-based services without having to support a personal or local web server. At the time of this writing, it was too early to assess the success of this facility.

Home Automation Interface

Home automation systems, such as X10, allow wireless control of lamps and appliances through the home by exploiting the existing electrical wiring infrastructure. These systems often have the addition capability to use RF receivers to relay home automation sequences. One researcher in the Cube uses such a system, but instead of simply controlling electric lamps and appliances, the system is programmed to execute UNIX commands. The researcher uses a wireless remote control to change his Canard message profiles. One button sets the "I'm at home" profile, another sets the "I'm at work". This system can be used to update to sixteen different profiles as well as control the lights in his office. This combination of home automation system and Canard makes the task of changing Canard profiles transparent. Turning on the lights in the office also signals a change in Canard user profile. Actions from other external systems can be combined with Canard as well, one research used his telephone activity to change his profile. By picking up or putting down the handset on his telephone a signal was sent to the computer to change the contact information presented to people using the *finger* program.

Computer System Status Monitor

Given the near instant notification opportunities that Canard offered, it was inevitable that a system for monitoring the computer systems would be developed. One student who was managing a large project developed a system that would update him as to the status of the various subsystems. Regular updates were sent early in the morning and urgent reports were immediately communicated to the student's two-way pager. This helped the student resolve problems with his computer systems quickly.

4.6.6 Horror Stories

Although most of Canard users realized that the experimental paging system was prone to unannounced outages, some believed that it should behave otherwise.

> *I started using it when I first got it. But then I stopped. Here's why:*
>
> *The service was unreliable, in a particularly bad fashion. Specifically: the person who sends the page has no way of knowing if it's delivered (email pages don't bounce if they aren't delivered), and also, the person receiving the page (me) has no way of knowing that a page was sent and not delivered.*
>
> *Now, I have since discovered that I can log in and read a list of pages sent to me, including ones that weren't delivered.*
>
> *But, this caused a Situation between me and* xxx, *where she sent me a page saying "Hey lets get ice cream after you finish band tonight", I didn't get it, I coincidentally called her 5 minutes later and said "hey I'm going to grab pizza with* yyy *tonight", and she thought I read her page and dissed her.*
>
> *You can see the problem. If her email page had bounced, she would have known I didn't get it, and wouldn't have minded me wanting to have pizza with* yyy. *But as it stood, she assumed I GOT the freaking page because I was calling her minutes later.*
>
> *Needless to say she was pretty unhappy with me.*
>
> *We stopped using the pager shortly thereafter.*
>
> *–a dissatisfied Cube Occupant*

The sender of the message assumed that the recipient would receive the message immediately. This unrealized expectation was an artifact of how the pager was represented to his friend. He had described it as a means of instantly receiving messages regardless of his location. When the expectation of immediacy was not fulfilled, the user felt betrayed by the technology and chose to discontinue use.

Other users had problems with the reliability, but took it in stride.

> *The one problem that arose is that others who had my page address expected an immediate response. When this wasn't forthcoming, this quickly led to frustration. It's interesting that they would never expect such a quick reply from email messages, but they took it for granted that I would be available to write an appropriate response to their message.*
>
> *–A Graduate Student*

4.6.7 Reflections

The Cube proved a challenging environment in which to do a field study. As a member of The Cube community, this challenge was compounded by a frustration due to the apparent apathy of my peers. However, over time I began to understand some of the reasons why the graduate students were not interested in putting effort into the project. Since mobility was not a factor in their day-to-day routines, Canard offered them little direct benefit. The students who lived within walking distance of the lab tended to value the convenience of having a pager. However, even these students did not see much value in the use of their laptop computers. The original plan of using technology as an enticement to use the common space failed. However, those graduate students who were collaborating with undergraduates realized the benefits of Canard as a means to coordinate their activities.

4.7 Summary

The five communities that used the Canard platform provided valuable feedback that was the basis for developing the ASE Framework. These communities all shared the characteristic that their members believed that telecommunication systems play a pivitol role in their collective needs. Adapting telecommunication systems to meet their needs could be satisfied by either internal or external technical support and expertise. The amount of external support provided was a function of the receptivity each of the community.

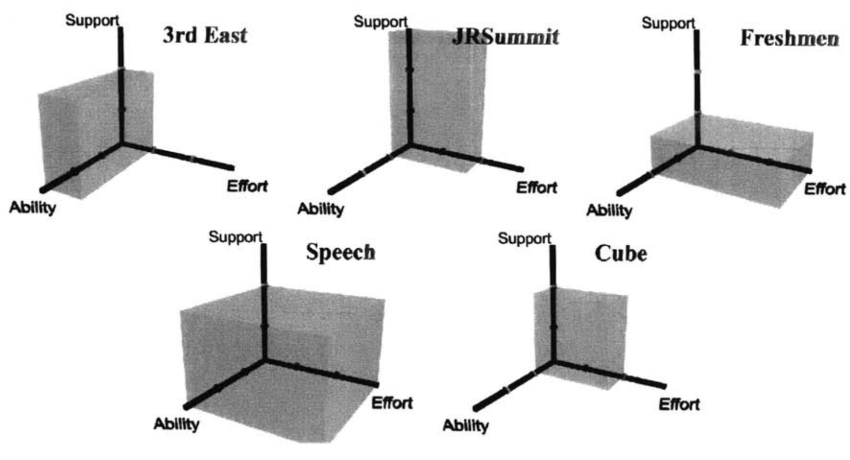


Figure 4-30: ASE Framework applied to five communities using the Canard experimental platform.

The MIT undergraduate residence, Third East, provided the initial setting for developing the Canard experimental platform. This community had a wide spectrum of abilities and needs. As a group they were willing to work with the developers to adapt the system to their needs. In some cases community members took the initiative to build support systems that would ensure greater use of the Canard system. The combination of high levels of ability, support, and effort ensured successful adaption of the technology.

The Junior Summit delegates greatly benefited from Canard's messaging system. It was their only means to easily communicate with one another. The notion that the technology could be used for chatting to one another was not a design goal for the system, but rather was in answer to the group's needs. Specialized applications were developed to meet the communication needs of the delegates. As a result, the delegates enjoyed a rich, easy-to-use communication environment that was adapted to their abilities and needs. The levels of support and effort were sufficient to lead to a successful outcome.

The group of students working on the Freshman project were motivated by a strong desire to help their fellow students during a difficult period on campus. As a group they had sufficient motivations to accomplish their project, but they were missing a framework that would help them identify the difficiencies in executing their plan. Effort in the absence of support and ability is not a likely scenario for success.

The Media Laboratory Speech Group had an existing infrastructure for messaging into which they easily integrated Canard. Canard presented them with an enhanced opportunity to monitor the status of messages and the wireless devices. As a group they had the technical ability to be self-supporting. Spot intervention by the Canard developers facilitated rapid integration and new services, but this community was probably capable of making these advances autonomously. High levels of ability and effort can take a community far, even in the absence of support.

The Cube example illustrates how it is easy to make false assumptions about a community. The original hypothesis suggested that the graduate students would benefit from a completely wireless environment. When the hypothesis proved wrong, the equipment fell into disuse. However, the enterprising members of the community organized a redistribution of the wireless devices to benefit the undergraduate community they worked with. Effort, coupled with moderate levels of support and abilities, led to a successful adoption of the technology.

Chapter 5

Conclusions

This thesis is motivated by an underlying belief that social constructionism as an epistemological approach is crucial to the introduction of technology to communities. As members of a community, individuals can be active participants in building the systems that affect them. Member's level of participation can impact the system development. Many are content to just use a system, others wish to extend systems, and still others will want to take the benefits of a system and apply them to new communities. The telecommunication industries should embrace such an approach as they hope to play a central role in a community's well being in post-industrial societies.

The telephone system example described in the introduction to this document is a revealing one. It highlights the positive aspects of social constructionism as well as the potential for failure of traditional design. As a group, the Garden community engaged itself in enhancing its telecommunication system and the community members were invested in these improvements. The MIT telecommunication office disrupted this activity by imposing an incompatible system.

Could have the MIT telecommunication office avoided the disruption? The ASE Framework suggests that by adopting an appropriate framework, in particular, a modest amount of support for "hacking," the social-constructionist activities in the Garden could have been sustained. The MIT telecommunication office might have discovered that this community and others at MIT had the ability to extend its own phone system but needed spot support to encourage this activity. The Garden was certainly willing to overcome many of the complexities associated with interfacing their system with the MIT phone system. But, without external support, the call announcement system could not integrated with the new phone system.

The ASE Framework presented in this thesis is meant to bring together a number of established design approaches so that a system designer can better understand a target community.

- A hacker[Lev94] ethic that promotes opening software systems not only to developers, but ultimately to the end users is critical. "Transparent" systems that reveal their inner-workings increase the chance that contributions will be made by community members.

- An iterative design approach that treats design as dynamic and adapts it as competencies and needs change was found to be an effective means of engaging a community.

- Ethnographic tools were found to be important design resources for social constructionism. Observations of community settings reveal social nuances that cannot be obtained through a traditional requirements gathering approach to design (e.g., waterfall design).

- Assessing the impact of external support on a community is a critical design stage. Formal support is usually necessary in the short term, but it is difficult (and expensive) to sustain. Self-supporting systems are desirable because they allow communities to take ownership of their problems and solutions. When a community initiates the use of a technology, they take ownership of the design rationale as well. External support is not an all or nothing proposition; spot intervention has proven effective and can inspire greater participation from a community.

- The three attributes weighed in the ASE Framework, ability, support, and effort, enable the designer to examine technology integration holistically. Thus, the designer may avoid the common mistake of equating functionality with acceptability. The framework suggests that when examining these attributes it is important to look beyond how community members are using a system. Opportunities for extending a system capabilities or applying it to new communities also merit consideration.

- The field studies were a means to evaluate the reception and utility of a social-constructionist approach to community-centered messaging. The ASE Framework evolved from a close examination of both the successes and the failures. It serves as both a predictive and evaluative tool.

- The ASE Framework is not an absolute predictor of success. Some factors such as individual and community effort are difficult to assess in advance. The ASE Framework offers more consistent results when used as an evaluative tool, where it provides guidance as to how to adapt a design for the further benefit of a community.

The communities that used the Canard messaging system benefited from external support provided to them by the Canard developers. The level of interaction varied from group to group, but in every case it caused the community to participate actively in the design process. In response to the enthusiasm expressed by these communities, the developers paid more direct attention to community needs. For example, the Junior Summit delegates would not have been able to construct their communications system on their own. Yet their enthusiasm and the uniqueness of their endeavor inspired Motorola to participate more deeply in the summit than just as a financial enabler. Specialized applications were developed to address the perceived needs of the community. When those applications did not fit the delegate's needs, the developers quickly adapted the system. Both the undergraduate residents of Third East and the Media Laboratory's Speech Group developed unique applications based on model applications developed by the Canard team. The only community that did not fare well was the freshmen seminar project in which there was little external intervention. Had they been provided the same support as the other groups their project might have had a more positive outcome. These field studies illustrate the relationship between developer and community is not strictly one way. Communities inspire developers and, in some cases, communities are developers.

The Canard experimental platform provides a means for exploring the ASE Framework. Its modular design is simple to understand, implement, and operate. It enables individuals to adopt and adapt different components of the system. Its design evolved from prior systems that were intended to simplify the task of developing complex systems by groups of people. The program competence of the developers ranged so much that attention was placed on providing rich examples for new programmers to use, allowing them to focus on the problem space and not the programming details. This process of simplifying the technology tools had the ultimate goal of allowing the users to be developers.

5.1 Future Work

Canard could have been implemented either as a centralized server or as a set of distributed clients. However, the experimental protocol approved by COUHES allowed for only a centralized approach. The cost of this limitation was two-fold: (1) the limited trust between the users and the service provider meant that there were few personal databases shared with the server; and (2) much of the implementation remained hidden from the users. The opacity of the implementation made it more difficult for individuals to add functionality to Canard. The communities, in a number of cases, were able to overcome these difficulties on their own or with assistance from the Canard developers.

Although managing a distributed system can be a logistical nightmare, in the long run, it would be desirable to employ distributed architecture for Canard. In a distributed system, personal databases would reside on an individual's computer and thus be more readily accessible and maintainable by the individual. Also, running communication servers on local machines gives the user more ready access to the servers and the opportunity to learn more about how the servers works and how to adapt them to their needs.

Over the course of the field studies, the Canard tools evolved from an interface that only experts could use to one that "power users" might find useful. It is unclear if these tools could be further adapted for novice users. One aspect of Canard that needs further exploration is a toolkit for building services. Canard holds promise as a platform for learning about the manipulation of databases and servers, but the benefits have not be tested.

The ASE Framework presented in this thesis is not intended to be all encompassing. There is more work to be done in refining and extending it. In particular the Ability and Effort attributes are far more complex than presented in this thesis. In this thesis we focus on intra-community and service provider support, while there exists other mechanisms for support. *Inter*-community support also holds promise. For example, the seniors in the Melrose community are now engaged in assisting a senior group in Revere.

Technological changes effect the rate adoption of technology. As consumer technology matures it becomes better understood and easier to use. There are social forces at work as well. Currently "Geek is chic." Technology holds a prominent play in television sitcoms. Popular actors are seen using laptop and hi-tech telecommunication devices. As a result, technology is becoming part of popular culture.

The choice of communities evaluated in this thesis was limited to ones physically proximal to the Media Laboratory. It may be instructive to try similar experiments off of the MIT campus. Intervention had a positive impact on the children attending the Junior Summit. Using this approach with seniors in an assisted-living community might also prove beneficial.

Canard as used by the experimental communities was presented as a closed, centralized system. Restructuring Canard as an open source system would encourage development of community-centered messaging in other environments. Learning from these developments could hold benefits for other communities.

The predictive capability of the ASE Framework has been illustrated in a number of cases. For example, the framework predicts that systems which rely heavily on external support are difficult to sustain past the period of formal intervention and are likely to fail. Systems that encourage sustained effort by community members and are able to leverage the variety of abilities found within the community are likely to be sustainable, regardless of the extent of external support. While the ASE Framework has only been applied to the domain of community-centered messaging, it is likely to be useful wherever technology is being integrated into established communities.

Appendix A

Third East's Proposal to Participate

A.1 Hall Profile

Third East, located on the scenic third floor of the east parallel of East Campus, is home to 44 undergraduates and one graduate resident tutor. We are 25 males and 20 females, with a racial mix of about 50% white and 40% Asian (and 10% other). There are 14 frosh, 10 sophomores, 12 juniors, and 8 seniors. As anyone probably could have guessed, course 6 is the most popular major, with 9 upperclassmen in EECS. Other popular majors are 2 and 8 (see table 1 for details). With 6 frosh already planning on choosing course 6 (see table 2), it looks like that trend will continue. Two cats also live on the hall.

A.2 Hall Communal Activities

The hall does many communal activities, the most popular of which can be classified as "hanging out in the lounge." On an average day, there are usually people in one or the other lounge from 11am until about 6am (or later when there is a problem set due), with activity peaking around 12-1am. Other activities that we do together include: hall feeds, problem sets (especially for popular classes such as 6.*, 18.02, 8.01, 8.02, etc.), eating

Table A.1: upperclassfolk by major

1	1
2	4
4	1
5	2
6	9
7	3
8	6
9	1
10	1
14	1
unknown	1

Table A.2: frosh by intended major

2	1
6	6
8	2
22	1
unknown/undecided	4

at Walker, eating out (Durgin Park, Buzzy's, etc.), cooking, watching movies, IM sports (tennis, soccer, hockey), multiplayer computer games, going to the Coffeehouse on Saturday nights, and random other "fun" hall activities.

A.3 What We Expect

It's difficult to formulate expectations with a project that is in such an early stage of planning, so we don't have any concrete expectations for Canard. We don't expect that the pagers will boost our GPA, get us a better social life, or cure cancer–we mostly think it would be really cool to have a system like Canard around to make our lives easier, hack on, and play with. From what we've heard, though, we can come up with some pretty general expectations:

For starters, we expect that Canard would make life easier (why get involved with something that makes life harder?), by allowing people to communicate better and find each other more easily. As many of us are interested in the technology, we expect that there would be opportunities to hack on the system and improve it. We certainly hope that there would be some technical support from the researchers in the form of help getting started and maintenance of any communications servers that are necessary for the functioning of the system. Also, feedback from the researchers would be welcome, such as how you think the project is going and what we can do to help work towards your research goals. Finally, we expect that we could get pagers for some off-hall friends (e.g., Dwight) who interact a lot with hall residents.

Most importantly, we expect that it will be a fun project.

A.4 What You Can Expect

If we get the project, you can expect a group of people that is interested and excited about using this technology and discovering what kind of things can be done with the system. The vast majority of the residents on the hall interact with each other (i.e., we have very few "ghosts" who are hardly around), so you can expect a high level of participation in the project. Especially with the large number of enthusiastic freshman on the hall, you can also expect a sustained level of participation in the future. You can expect us to try and use all the functionality of the system and to push the limits of what the system was designed for. To this end, we are also more than willing (nay, eager) to hack on the system and create any new software that needs to be written, or any software that sounds cool for that matter. If there are problems with the system or things we don't like about it, we will definitely provide feedback about how the project is going and what could/should be done to improve it. Most of us aren't afraid to be honest about "what sucks" and what doesn't.

A.5 Time Commitment

We are excited about this project and willing to put any reasonable amount of time into providing feedback to the researchers. Group meetings every couple of weeks (maybe as often as once weekly) would probably not be a problem, and individual meetings would also be acceptable. As long as participation in the project doesn't become as time-consuming as an average class, it would probably be okay. A time commitment from us on the order of a laid-back freshman seminar would probably be just about right.

A.6 Life With Canard

"After almost pulling an all-nighter, I woke up groggy this morning for class. Afterwards, I checked with Canard to see if anyone had brought the Globe up from the front desk–guess not. I grabbed a falafel from the food truck and the Globe from desk and headed up to the lounge to relax, eat, read the comics, and do the crossword puzzle. Halfway through lunch, Don buzzed me asking if I had turned in that problem set yet. Oops! I nearly ran down to building 2 to get that thing in before noon. On my way back, the pager beeped again–my idiot roommate had locked himself out *again*. Since I was on my way back, I buzzed him and let him know that I'd be back soon. Good thing he didn't lock his pager in the room."

"11am class: It's getting awfully boring, what with the professor reading straight from the lecture notes. So I get out the pager and see if Kathy or Larry want to eat at Walker after class. Kathy's schedule says she's in class until 2, and Larry's pager is accidentally turned off (I'll have to page Mike to wake him up and tell him to turn it on). Since I can't reach either of them, I send a page out to see if anyone's eating soon and get a reply from the group at Walker. Looks like it'll be Pizza Hut for lunch."

"The puntlist is so much faster with the pagers. Ever since Debbia wrote the pager interface, we don't have to hunt people down and expect people to walk by the bulletin board to pick rooms anymore. Sure, we still stand around the paper copy and speculate about room sizes and choices, but now we can hassle whomever's up, even if they're not on the hall. The pagers also turned out to be ultimately useful for the IAP Mystery Hunt. With people running all over the institute, it became really useful to be able to get someone's attention while they were god-knows-where in the Institute."

"Top Ten Reasons Why Third East Should Do The Canard Project"

10. Free food with follow-up meetings

9. Steal a pager and gain access to that person's *exciting* mail

8. Save time - read email during lecture instead of sleeping

7. Those canned responses make flaming quicker and more efficient

6. With the handy dandy scheduler, you'll never forget to punt class again

5. Pagers are cool, heh, heh, heh

4. Free Vibrator with unlimited batteries

3. 2 words: Pager Doom

2. Fifth East shouldn't get it and the number one reason ...

1. 'Is that a vibrating pager in your pocket or are you just happy to see me?'

Appendix B

Junior Summit Charts

During Junior Summit week a database of all messages processed by the Canard system was kept. This provided the raw data used in the field studies section. For privacy reasons, all messages that originated from outside the Canard system were discarded. In addition, all messages that were destined to non-Junior Summit members were discarded in compliance with MIT's Committee on the Use of Human Subjects experimental protocol requirements. All that remained were the system logs and messages that Junior Summit delegates sent. From these messages the identities of the sender was removed. Scripts were used to analyze messages for patterns. These logs and script results made up the data that were analyzed and charted in this thesis.

B.1 Outbound Message Destinations

Junior Summit delegates used two-way pagers to originate thousands of messages a day. The breakdown of the destination of these messages can be seen in Figure B-1. As can be seen the majority of the messages were sent to other delegates using the Canard system. A noticable drop of messaging activity occurs on Thursday. The delegates spent the day in a room preparing presentations.

The two-way pager is able to send messages to single or multiple addresses. All the Junior Summit used the pagers to message single destinations (see Figure B-2). However, a large number of Junior Summit delegates did not use the pager's ability to send the same message to multiple recipient.

The Junior Summit delegates sent a number of messages to addresses at jrsummit.net (see Figure B-3). Upon closer examination it was evident that the delegates were a bit confused about how to address messages to other delegates at the conference at the Media Laboratory. Additionally, it was discovered that the jrsummit.net mail server prevented the delivery of messages without subject lines.

Some delegates collaborated with individuals on the MIT campus (see Figure B-4). As their final presentation day approached the volume of messages destined to people at MIT increased.

The Junior Summit delegates came from around the world to attend the conference at MIT. It is not surprising that a large number of messages were destined outside of the MIT network (see Figure B-5).

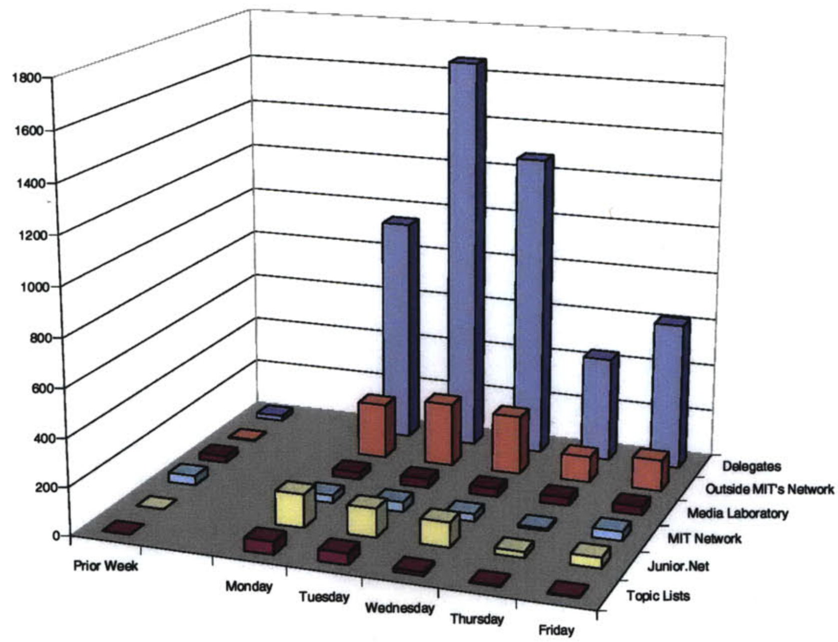


Figure B-1: Outbound messages from the pagers were primarily directed to other delegates using the Canard system.

B.2 Messages to Teams and Leaders

The two-way pagers were pre-programmed with addresses book entries that were useful for addressing groups withing Junior Summit. There were five "Team" in Junior Summit, each had a counselor (a team "leader"). The address book entries allow messages to be sent to the leader (see Figure B-6) or Team (see Figure B-7). Not many of the delegates used this facility.

B.3 Salutations and Parties

Given the social nature of the delegates, it was not surprising that a large number of messages salutations of the form "hi", "good morning", "hola!"... This message activity can be seen in Figures B-8.

Parties were the subject of numerous messages during Junior Summit week. The two-way pagers allowed delegates to organized and announce these events (see Figures B-9). These messages were sent to delegates via the on-line forums, team broadcast addresses, and the LED signs in the conference rooms at the Media Laboratory.

B.4 CheckIn and Find Applications

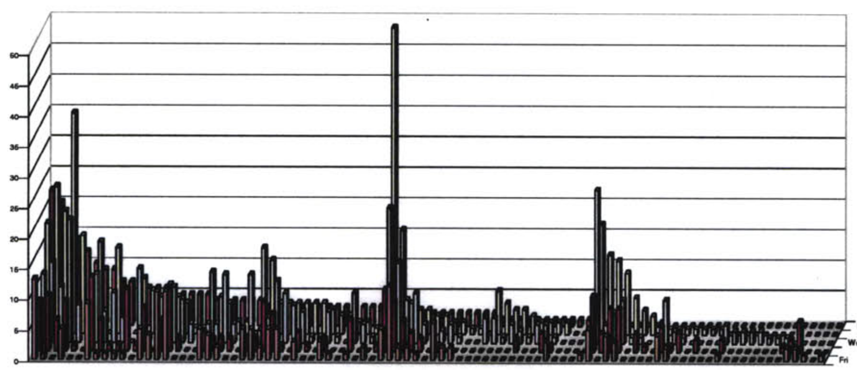
Two pager applications were developed for Junior Summit: CheckIn and Find. These applications allowed delegates to locate one another through a centralized database. CheckIn was used to update a delegates location in the Media Lab building. Find was used to search for a delegate's location in the database. Usage of the applications was voluntary. Both CheckIn and Find were used by a majority of delegates at the beginning of the week (see Figures B-10 and B-11), but the usage dwindle as the database was not kept up to date by individual delegates.

Some delegates found it easier to send an explicit message out to the individual they were looking for (i.e., "where are you?"). Figure B-12 shows roughly half the delegate sent such messages.

B.5 Messages to LED displays

LED moving signs were very popular during Junior Summit week. Practically all the delegates sent a message to the displays (see Figure B-13). The signs came on-line on Tuesday, and Wednesday the delegates were told not to send messages to the displays.

Messages to Other Delegates



Messages to Multiple Delegates

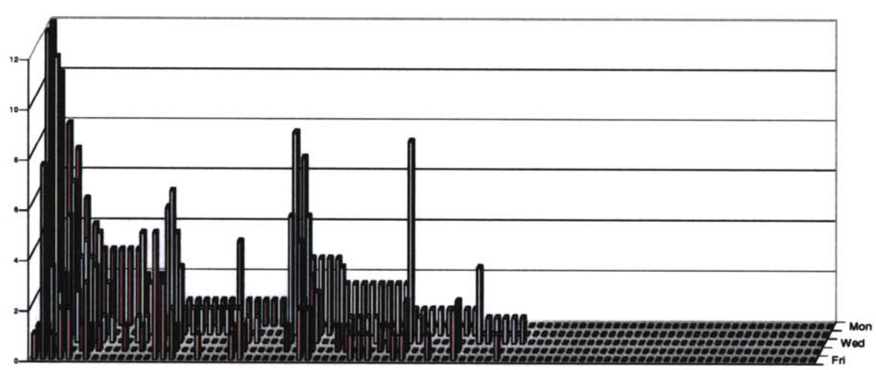
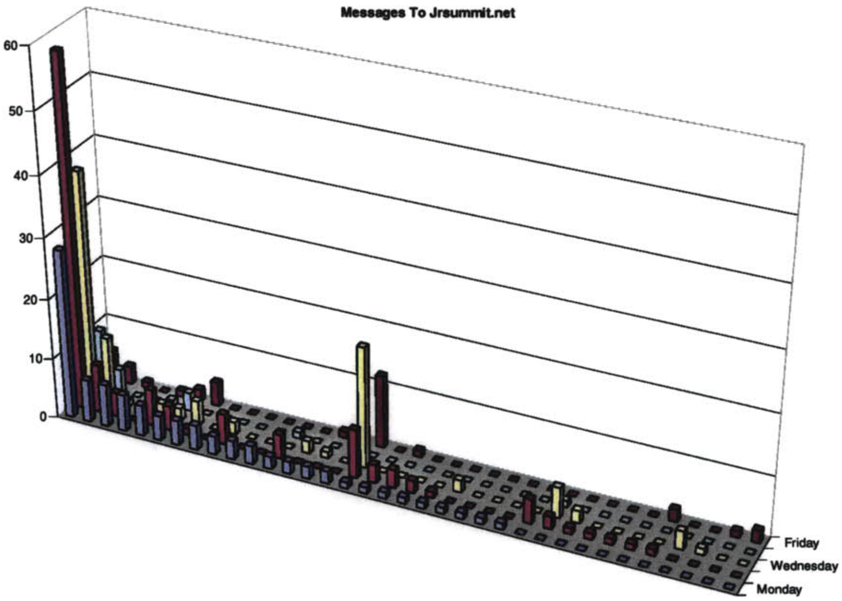


Figure B-2: Top: Users sent mostly messages point to point. Bottom: A significant number of delegates did not ever send messages to multiple destinations.



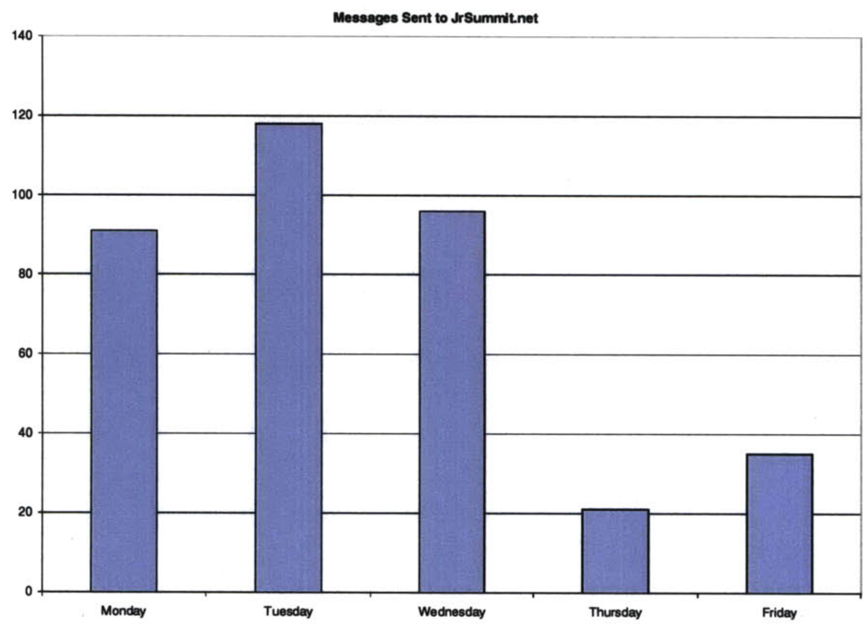
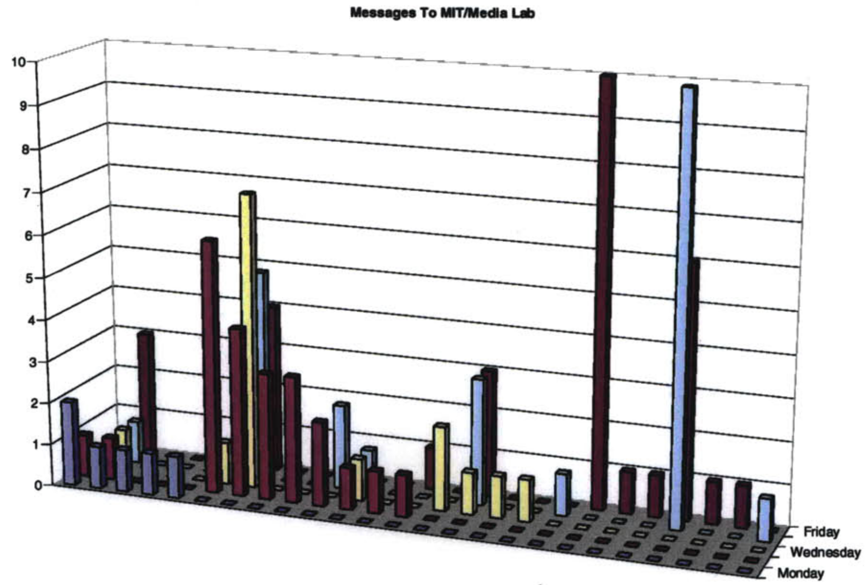


Figure B-3: Top: Messages destined to addresses at jrsummit.net Bottom: Daily total of messages destined to addresses at jrsummit.net



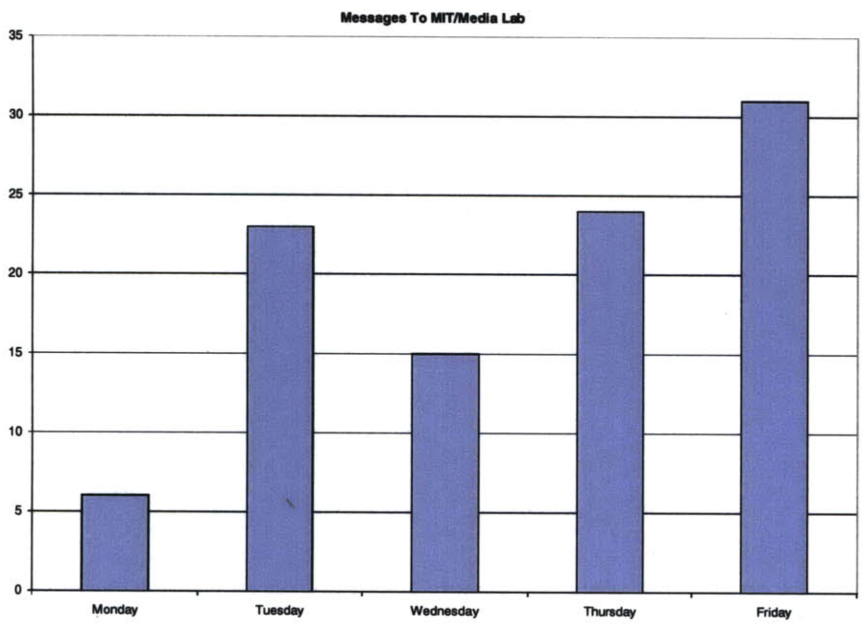
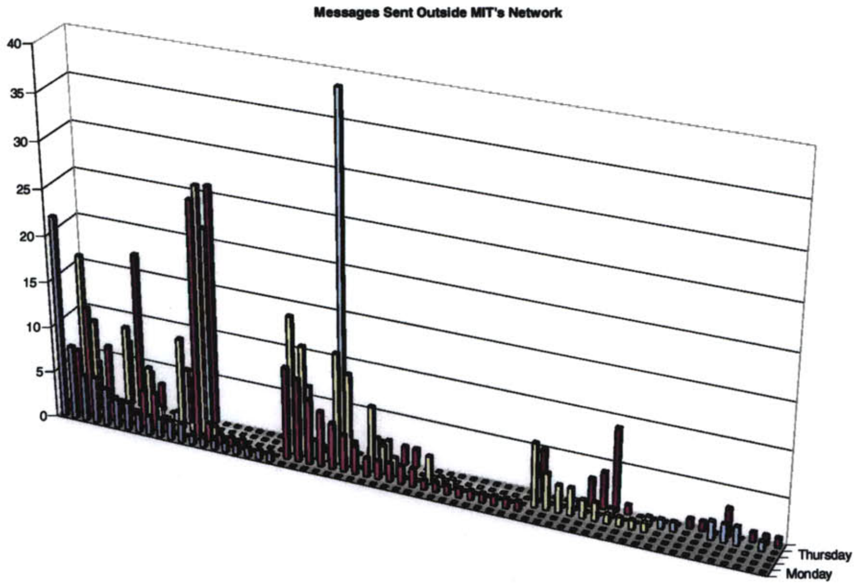


Figure B-4: Top: Messages destined to addresses on the MIT network. Bottom: Daily total of messages destined to addresses on the MIT network.



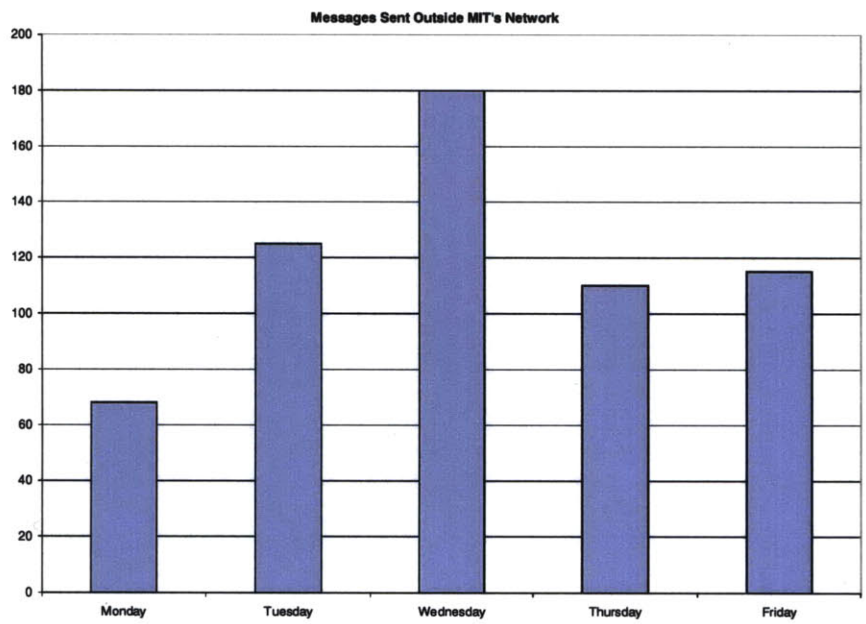


Figure B-5: Messages destined to addresses outside the MIT network.

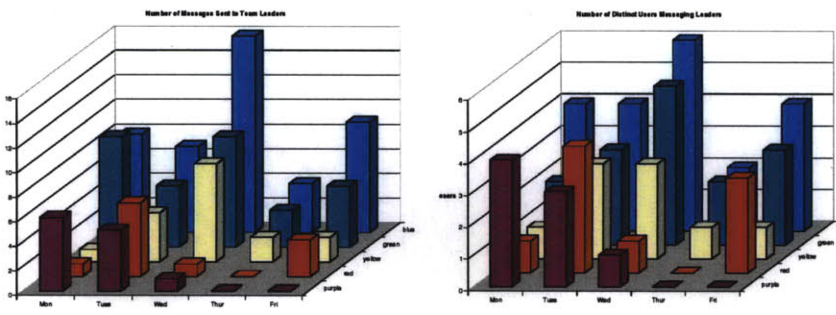


Figure B-6: Left is the total messages sent to team leaders. Right is the total number of users sending the messages.

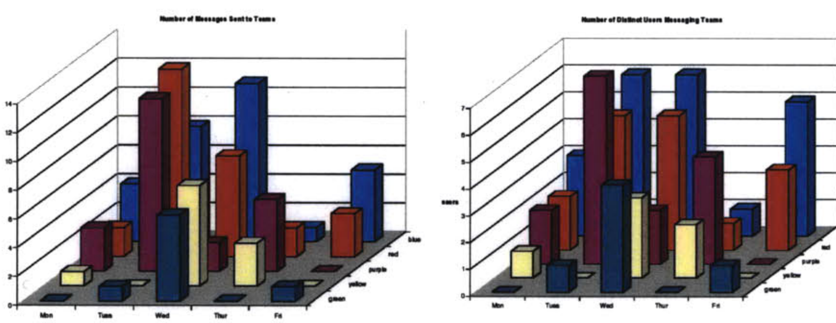
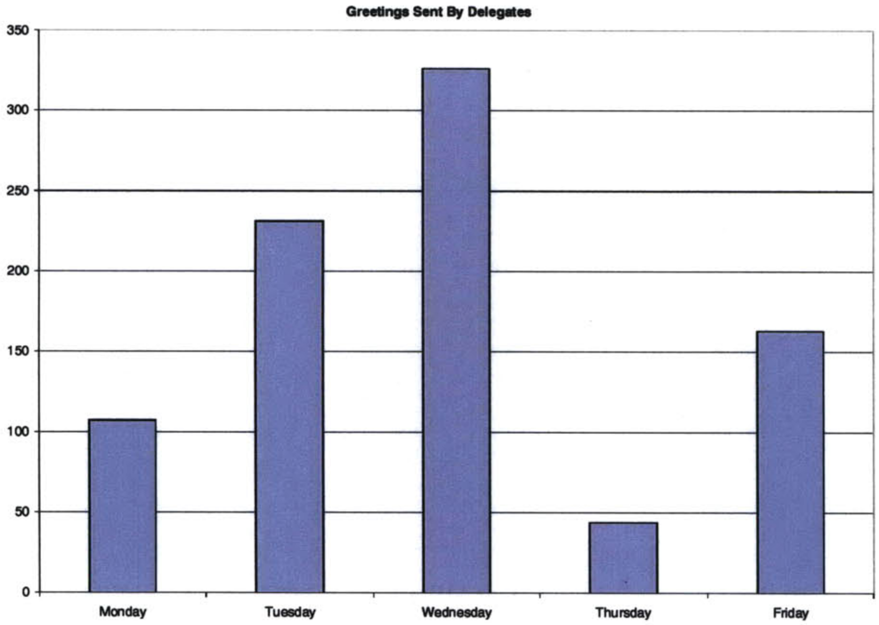


Figure B-7: Left is the total number messages sent to team broadcast address. Right is the number of users sending the messages.



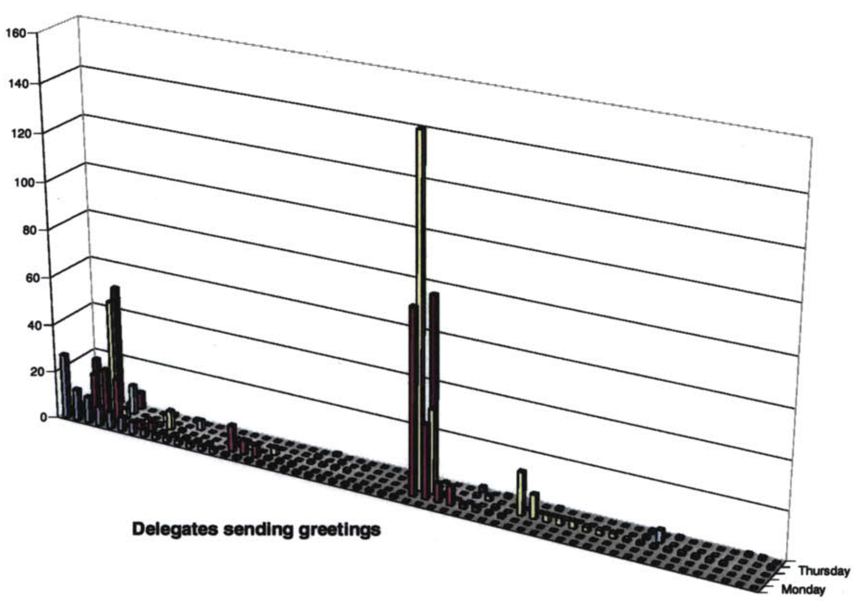
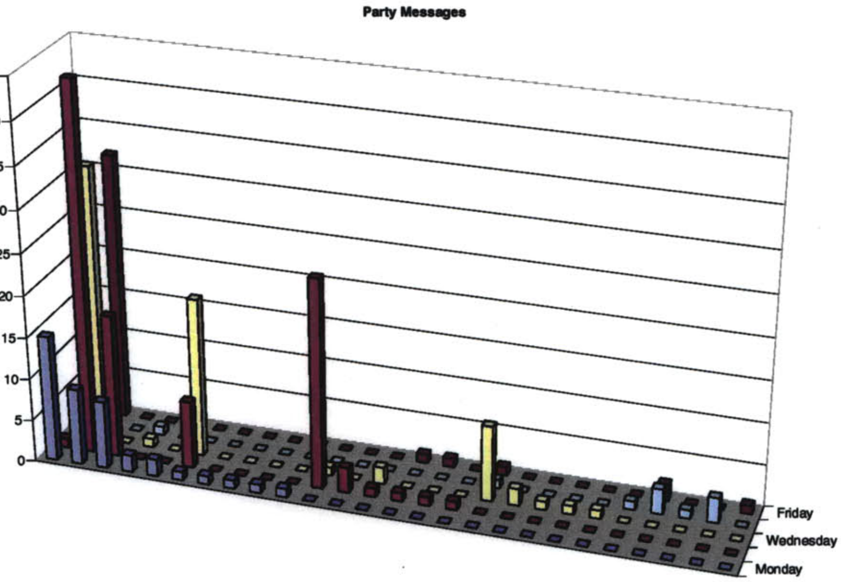


Figure B-8: Junior Summit sent a lot of messages that were just salutations (i.e., "good morning", "hi", "hola!") with no other substance to the messages. Total saluations each day of Junior Summit.



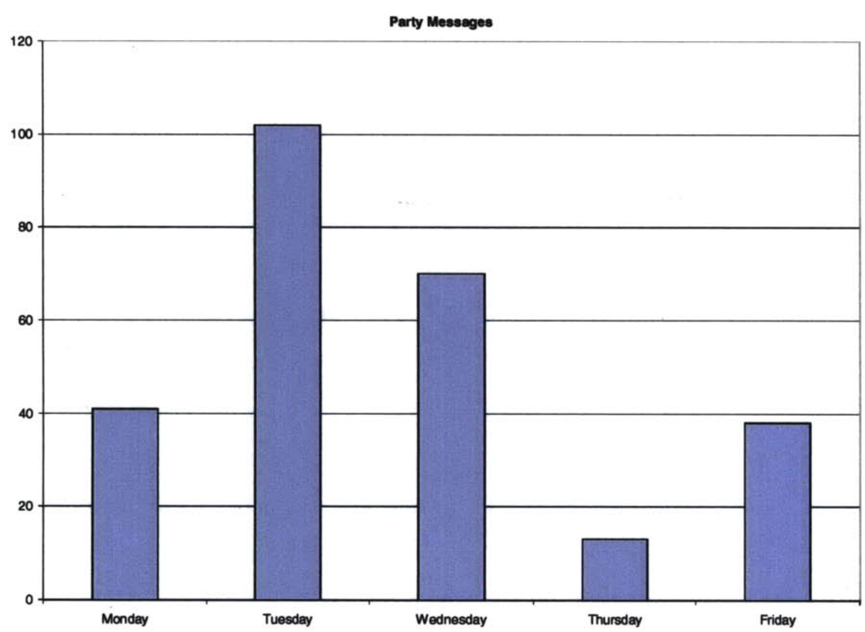


Figure B-9: A few Junior Summit organized parties with the pagers.

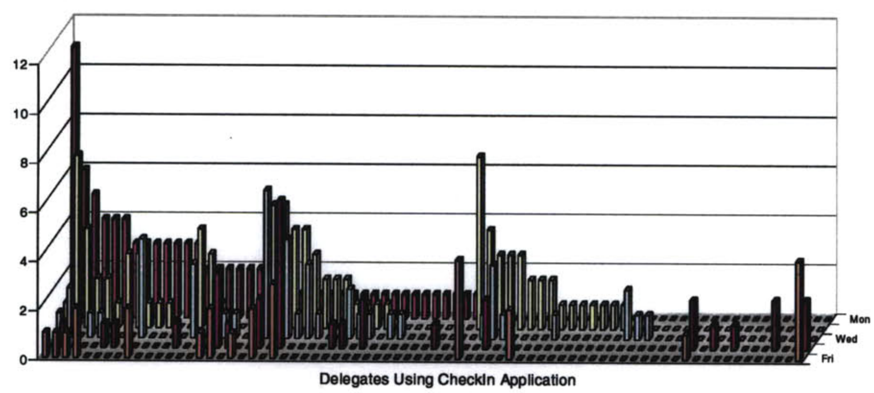
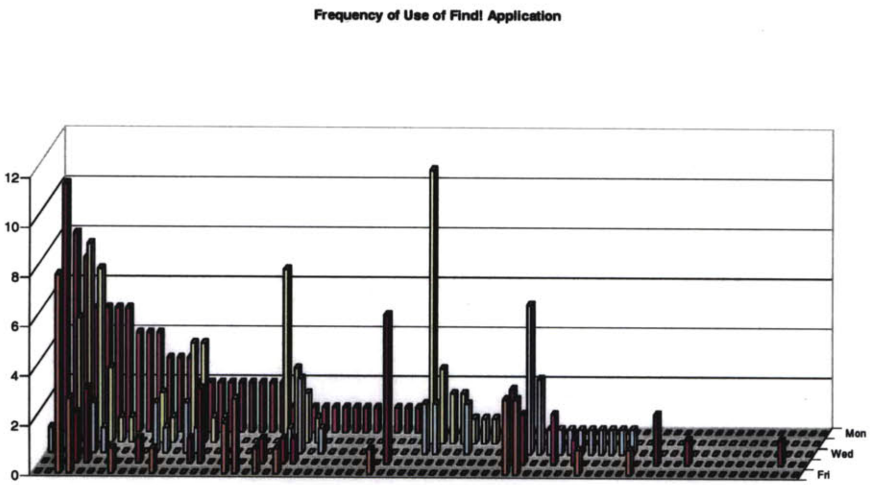


Figure B-10: Usage of the CheckIn program at



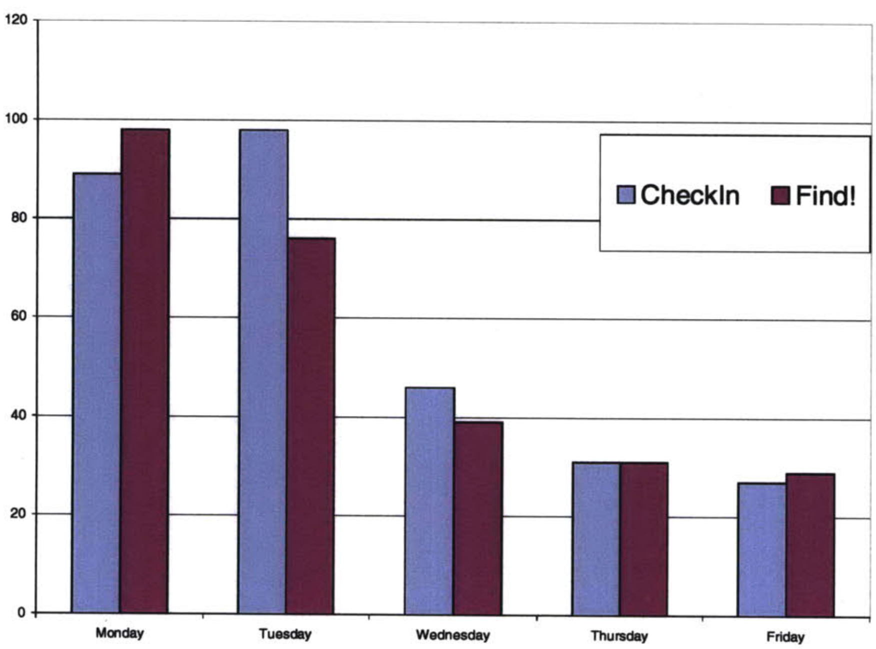


Figure B-11: Usage of the Find! pager application.

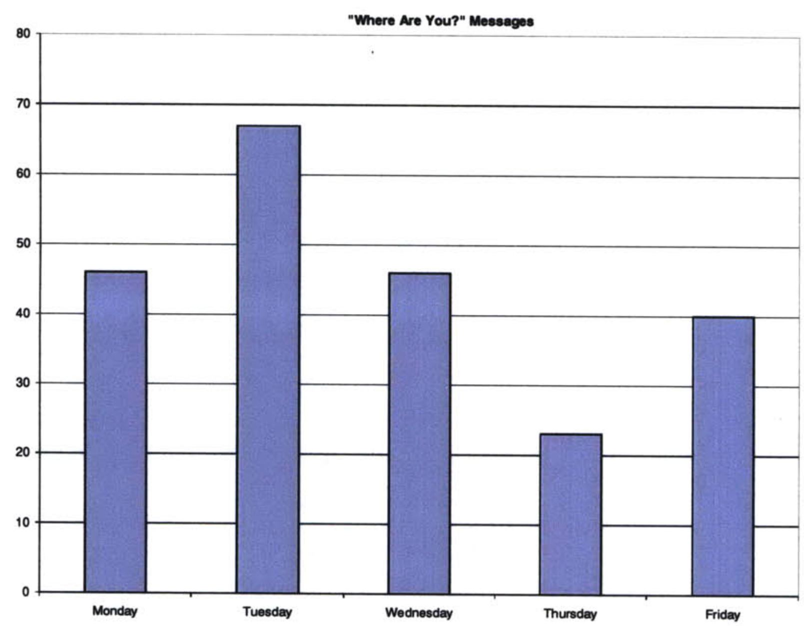
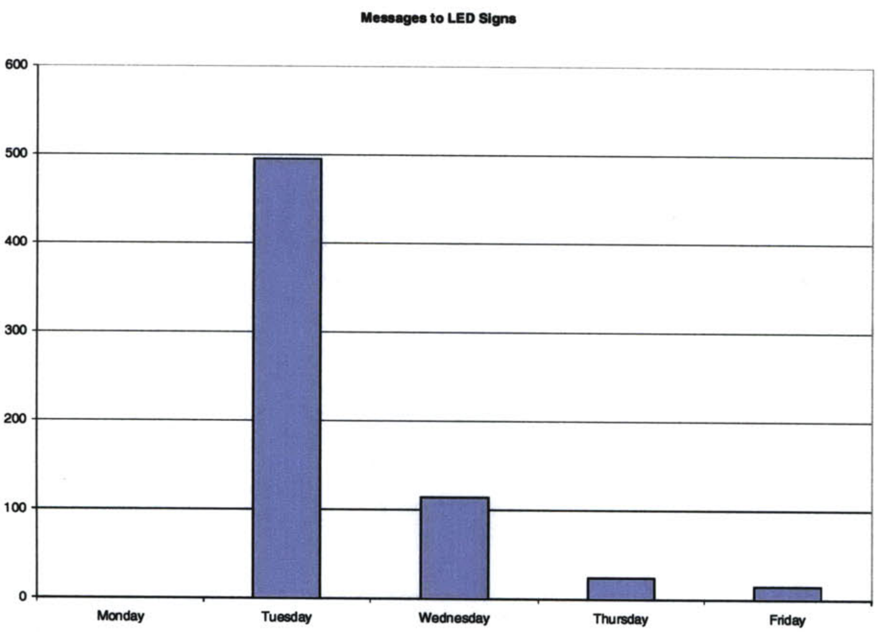


Figure B-12: Roughly half of the Junior Summit delegates sent messages explicitly seeking the location of someone during the week.



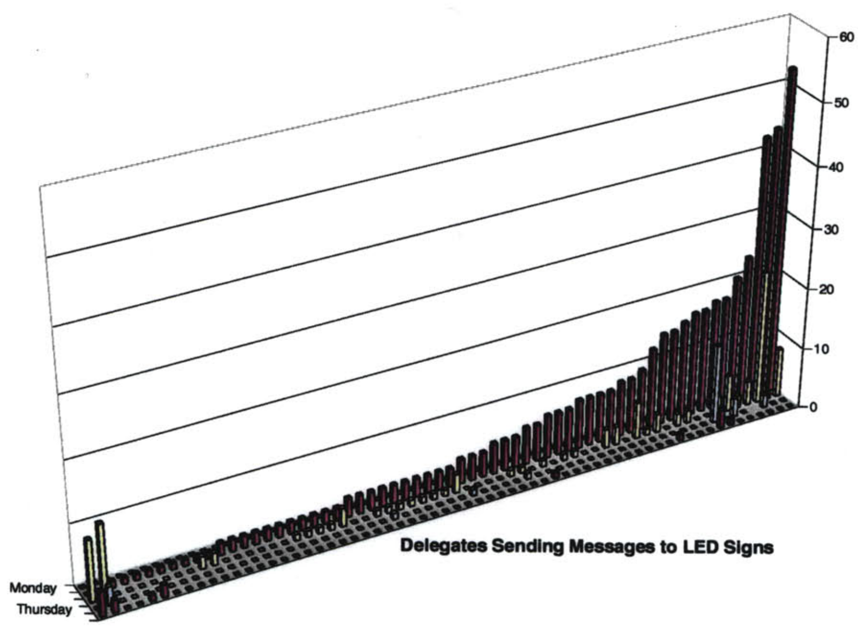


Figure B-13: Messages to LED signs over the whiteboard were very popular from the time they were made available. Then the usage dropped off when the delegates were told not to send message to them, and subsequently were unplugged from the system.

Bibliography

[ABB+93] C. Dennis Allen, Don Ballman, Vivienne Begg, Harold H. Miller-Jacobs, Michael Muller, Jakob Nielson, and Jared Spool. User involvement in the design process: Why, when and how? In *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, pages 251–254. ACM, April 1993.

[Abr92] Nathan S. Abramson. The Dtype library or, how to write a server in less time than it takes to read this manual. Technical report, MIT Media Laboratory Electronic Publishing Group, Cambridge, MA, 1992.

[Abr93] Nathan Shalom Abramson. Context-sensitive multimedia. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, February 1993.

[Ado85] Adobe Systems Incorporated. *PostScript Language Reference Manual*, 1985.

[Ala89] Aladdin Enterprise. *GhostScript User's Manual*, 1989.

[All99] Scott Allen. Stuck in park. *The Boston Globe*, pages E1, E4, February 1999.

[alt99] IBM Intelligent Agent Services. Available at http://www.research.ibm.com/iagents, August 1999.

[Amu96] Michael C. Amundsen. *MAPI, SAPI, and TAPI: Developer's guide.* Sams Publishing, 1996.

[anp84] Wire service transmission guidelines. Technical Report 84-2, American Newspaper Publishers Association, Washington, DC, 1984.

[AP96] Mark S. Ackerman and Leysia Palen. The Zephyr Help Instance: Promoting ongoing activity in a CSCW system. In *Proceedings of Computer Supported Cooperative Work 96*, Cambridge, MA, 1996. ACM.

[apr99] Projeto aprendiz. Available on the web http://www.uol.com.br/aprendiz, August 1999.

[Aro92] Barry Arons. Tools for building asynchronous servers to support speech and audio applications. In *Proceedings of the ACM Symposium on User Interface Software and Technology.* ACM, November 1992.

[AS96] Harold Abelson and Gerald Jay Sussman. *Structure and Interpretation of Computer Programs.* MIT Press, Cambridge, MA, 1996.

[BB96] Victoria Bellotti and Sara Bly. Walking away from the desktop computer: Distributed collaboration and mobility in a product design team. In *Proceedings of Computer Supported Cooperative Work 96*, pages 209–218, Cambridge, MA, 1996. ACM.

[BCE+96] Walter Bender, Pascal Chesnais, Sara Elo, Alan Shaw, and Michelle Shaw. Enriching communities: Harbingers of news in the future. *IBM Systems Journal*, 35(3&4):369–380, 1996.

[Ber94] Stephen R. van den Berg. *Procmail: Mail processing Package.* RWTH-Aachen, Germany, October 1994.

[BH93] Liam J. Bannon and John A. Hughes. The context of CSCW. In K. Schmidt, editor, *Report of CoTech Working Group 4 (1991–1992)*, pages 9–36, February 1993.

[BL89] Tim Berners-Lee. Information management: A proposal. Technical report, CERN, March 1989.

[Blo91] Alan Blount. Bettyserver: More news than you can beat with a stick. Technical report, MIT Media Laboratory Electronic Publishing Group, Camridge, MA, 1991.

[Bro95] Darrick Brown. *LEDSign 2.7*, 1995.

[Bru97] Amy Susan Bruckman. *MOOSE Crossing: Construction, Community, and Learning in a Networked Virtual World for Kids.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1997.

[Che96] Pascal Chesnais. Application for approval to use humans as experimental subjects no. 2334. COUHES Experimental Protocol 2334, November 1996.

[Che97a] Pascal Chesnais. E-mail extension to the canard community messaging system. Proposed extension to COUHES Protocol 2334, February 1997.

[Che97b] Pascal Chesnais. Improving MIT's human subjects internal review board. A Report Submitted to COUHES, December 1997.

[Che97c] Pascal R. Chesnais. Canard: A framework for community messaging. In *The First International Symposium on Wearable Computers*, pages 108–115, Cambridge, MA, October 1997.

[CMS95] Pascal Chesnais, Matthew Mucklo, and Jonathan Sheena. The FishWrap personalized news system. In *Proceedings of the 1995 2nd International Workshop on Community Networking*, pages 275–282, Princeton, NJ, June 1995.

[Cou98] Kino Coursey. *MicroNoc Version 1.1 Reference Manual.* Motorola, 1998.

[Cro82] D. Crocker. Standard for the format of ARPA internet text messages. *Network Working Group Request for Comments*, (RFC822), August 1982.

[Dav91] James Raymond Davis. Let your fingers do the spelling. *Journal of the American Voice I/O Society*, 9:57–66, March 1991.

[DE98] John Driscoll and Ingeborg Endter. *Silver Stringers: community-centric news*, 1998. A website describing the Silver String project of the News in the Future consortium at the MIT Media Laboratory. Available http://stringers.media.mit.edu.

[DEF+88] C. Anthony DelloFera, Mark W. Eichen, Robert S. French, David C. Jedlinsky, John T. Kohl, and William E. Sommerfeld. The Zephyr notification system. In *USENIX Proceedings*, 1988.

[Dew38] John Dewey. *Experience and Education.* Collier Books, New York, 1938.

[Die95] Klee Dienes. Information architecture for personalized multimedia. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1995.

[Dig87] Digital Equipment Corporation. *DECTALK DCT101 Programmer's Reference Manual*, 1987.

[FB96] Ned Freed and Nathaniel S. Borenstein. Multipurpose Internet Mail Extensions (MIME) part one: Format of internet message bodies. *Network Working Group Request for Comments*, (2045), November 1996.

[FKK96] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The SSL protocol version 3.0. Technical report, Netscape Communications Corporation, March 1996.

[FPSK98] Geraldine Fitzpatrick, Sara Parsowith, Bill Segall, and Simon Kaplan. Tickertape: Awareness in a single line. In *CHI 98 Summary*, pages 281–282, Los Angeles, CA, 1998. ACM.

[Gav91] William W. Gaver. Sound support for collaboration. In *Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, Amsterdam, The Netherlands, September 1991.

[Gre88] Irene Greif, editor. *Computer-Supported Cooperative Work: A Book of Readings.* Morgan Kaufmann, 1988.

[Gru91] Jonathan Grudin. The development of interactive systems: Bridging the gaps between developers and users. *IEEE Computer*, 24(4):59–69, 1991.

[Gru94] Jonathan Grudin. CSCW: Its history and participation. *IEEE Computer*, 27(5):19–26, 1994.

[GS88] Irene Greif and Sunil Sarin. Data sharing in group work. In Greif [Gre88], chapter 17, pages 477–508.

[Haa96] Kenneth Haase. FramerD: Representing knowledge in the large. *IBM Systems Journal*, 35(3&4):381–397, 1996.

[ipu99] Ericsson iPulse^{TM} site. Available at http:www.oz.com/ipulse/, August 1999.

[Irv99] Larry Irving. Falling through the net: Defining the digital divide. Report, National Telecommunications and Information Administration – U.S. Department of Commerce, July 1999.

[Kay92] Erik A. Kay. MIRE: A Multi-user Information Retrieval Environment. Bachelor thesis, Massachusetts Institute of Technology, Cambridge, MA, 1992.

[KL98] Steinar Kristoffersen and Ljungberg. Mobicom: Networking in dispersed groups. *Interacting with Computers*, 10(1):45–65, March 1998.

[Koe96] Billy Vaughn Koen. Use of Internet Relay Chat (IRC) in distributed consensus forming. *Gulf Southwest ASEE Meeting*, 1996.

[LaM96] David Michael LaMacchia. The iflame client-based instataneous datagram communications substrate. Bachelor thesis, Massachusetts Institute of Technology, Cambridge, MA, 1996.

[Las99] J.D. Lasica. Citizens as budding writers and editors. *American Journalism Review*, page 92, July/August 1999.

[Lev94] Steven Levy. *Hackers: Heroes of the Computer Revolution*. Delta Books, 1994.

[LK96] Andrew Lippman and Roger Kermode. Media banks: Entertainment and the internet. *IBM Systems Journal*, 35(3&4):272–291, 1996.

[Man91] Sanjay Manandhar. Activity server: You can run but you can't hide. *USENIX*, pages 200–311, 1991.

[Mar95] Matthew Talin Marx. Toward effective conversational messaging. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 02139, June 1995.

[Mar98a] Stefan J. W. Marti. Active messenger. MIT Master of Science Thesis Proposal, December 1998.

[Mar98b] Stefan J. W. Marti. The two-way pager. MIT Media Laboratory MAS 714J final paper, December 1998.

[Mat99] Linda Matchan. it's, like, instant. *The Boston Globe*, April 1999.

[MGL+88] Thomas W. Malone, Kenneth R. Grant, Kum-Yew Lai, Rao Rao, Ramana, and David Rosenblitt. Semistructured messages are surprisingly useful for computer-supported coordination. In Greif [Gre88], chapter 12, pages 311–331.

[Mic97] Microtouch. *IBID Electronic Whiteboard User's Manual*, 1997.

[Mit99] William J. Mitchell. Etopia – Jim, there's life, but not as we know it. Draft of book, January 1999.

[Mot98] Motorola. *TalkAbout TA280SLK*, 1998.

[MP95] B. Moghaddam and A. Pentland. Probabalistic visual learning for object detection. In *IEEE Int'l Conference on Computer Vision*, pages 786–793, Cambridge, MA, 1995.

[MS96] Matthew Marx and Chris Schmandt. Clues: Dynamic personalized message filtering. In *Proceedings of Computer Supported Cooperative Work 96*, pages 113–121, Cambridge, MA, 1996. ACM.

[Mul91] Michael J. Muller. Pictive - an exploration in participatory design. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, pages 225–231. ACM, 1991.

[Nel93] Philip A. Nelson. *GNU Database Manager.* Western Washington University, 1993.

[Nor90] Donald A. Norman. *Design of Everyday Things.* Doubleday, New York, NY, 1990.

[one96] AT&T to shutdown PersonaLink; shift to Internet. AT&T News Release, July 1996.

[OR93] J. Oikarinen and D. Reed. Internet Relay Chat protocol. *Network Working Group Request for Comments*, (RFC1495), May 1993.

[Orw93] Jon Orwant. Doppelgänger goes to school: Machine learning for user modeling. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, September 1993.

[Orw96] Jon Orwant. For want of a bit the user was lost: Cheap user modeling. *IBM Systems Journal*, 35(3&4):398–416, 1996.

[Pap80] Seymour Papert. *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books, New York, 1980.

[Pap90] Seymour Papert. *Constructionist Learning.* MIT Media Laboratory, Cambirdge, MA, 1990.

[Pap91] Seymour Papert. Situating constructionism. In Papert and Harel [PH91], page 518.

[PH91] Seymour Papert and Idit Harel, editors. *Constructionism.* Ablex Publishing, 1991.

[Poi99] PointCast,Inc. *PointCast Network 2.6 Online Help*, 1999. Web site http://www.pointcast.com.

[Pos91] Jef Poskanzer. *Extended Portable Bitmap Toolkit*, December 1991.

[Pro98] Pro-Lite. *Pro-Lite User's Manual*, 1998.

[Ray98] Eric S. Raymond. The cathedral and the bazaar. Widely circulated critique available at http://www.tuxedo.org/ esr/writings/cathedral-bazaar/, November 1998.

[RBP98] J. Ramsay, A. Barbesi, and J. Preece. A psychological investigation of long retrieval times on the world wide web. *Interacting with Computers: the Interdiscplinary Journal of Human-Computer Interaction*, 10(1):77–86, March 1998.

[Res94] Mitchel Resnick. *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds.* MIT Press, Cambridge, MA, 1994.

[Rho97] Brad J. Rhodes. The wearable rememberance agent: A system for augmented memory. In *The First International Symposium on Wearable Computers*, pages 123–128, Cambridge, MA, October 1997.

[RMSS96] Mitch Resnick, Fred Martin, Randy Sargent, and Brian Silverman. Programmable bricks: Toys to think with. *IBM Systems Journal*, 35(3&4):443–452, 1996.

[Roa98] Chris Roast. Designing for delay in interactive information retrieval. *Interacting with Computers: the Interdiscplinary Journal of Human–Computer Interaction*, 10(1):87–103, March 1998.

[Rog95] Everett M. Rogers. *Diffusion of Innovations*. Free Press, New York, NY, 1995.

[RR96] Mitch Resnick and Natalia Rusk. The computer clubhouse: Preparing for life in a digital world. *IBM Systems Journal*, 35(3&4):431–439, 1996.

[RS94] Stefanie Ruther and Michael Stupperich. Electronic meeting assistance. In *Conference Companion CHI 94*, pages 87–88, Boston, MA, 1994. ACM.

[RT98] Pete Russo and Richard Tarabour. Unified messaging and enhanced services. White paper, Bellcore, 331 Newman Springs Road, Red Bank, NJ 07701, 1998.

[Sch93] Christopher M. Schmandt. Phoneshell: the telephone as computer terminal. In *Proceedings ACM Multimedia 93*, pages 373–382, New York, 1993. ACM.

[Sha95] Alan Clinton Shaw. *Social Constructionism and the Inner City: Designing Environments for Social Development and Urban Renewal.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1995.

[She94a] Jonathan Sheena. Mini-cell database. Can be found on FishWrap web site, 1994.

[She94b] Jonathan Sheena. The server abstraction. Can be found on FishWrap web site, 1994.

[She96] Jonathan Sheena. About time for news. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1996.

[Sil97] Silverman. Computer reminders and alerts. *ACM Computer*, 30(1):42–49, January 1997.

[SMRL97] Thad Starner, Steve Mann, Bradley Rhodes, and Jeffrey Levin. Augmented reality through wearable computing. Technical Report 397, MIT Media Laboratory Perceptual Computing Section, Cambridge, MA, 1997.

[SNS88] Jennifer G. Steiner, Clifford Newman, and Jeffrey I. Schiller. Kerberos: An authentication service for open network systems. In *USENIX Proceedings*, 1988.

[Sou98a] Mike Souryal. Evaluating a unified messaging service. White paper, Bellcore, 331 Newman Springs Road, Red Bank, NJ 07701, 1998.

[Sou98b] Mike Souryal. Unified messaging: Removing the barriers between traditional messaging services. White paper, Bellcore, 331 Newman Springs Road, Red Bank, NJ 07701, 1998. White paper found on their web site.

[Sta96] Thad Starner. Human-powered wearable computing. *IBM Systems Journal*, 35(3&4):618–629, 1996.

[Sun98] Suncoast Systems, Inc. *Computerfone Touch-Tone – Voice Out User's Manual*, 1998.

[TP91] Matthew Turk and Alex Pentland. Eigenfaces for recognition. In *Journal of Cognitive Neuroscience*, volume 3, pages 71–86, 1991.

[Tuf90] Steven Tufty. Watcher. Bachelor's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1990.

[Uni91] University of California at Berkeley. *Vacation*, December 1991. Part of the UNIX Berkeley Standard Distribution.

[Vyg78] L Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA, 1978.

[Wei91] Mark Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, 1991.

[Won91] Chi Chong Wong. Personal communications. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1991.

[Zim91] D. P. Zimmerman. Finger user information protocol. *Network Working Group Request for Comments*, (RFC1288), December 1991.