# One-Hundred Days in an Activity-Centric Collaboration Environment based on Shared Objects

**Michael J. Muller, Werner Geyer, Beth Brownholtz, Eric Wilcox, and David R. Millen**
IBM Research
One Rogers Street
Cambridge, MA 02142 USA
{michael_muller, werner.geyer, beth_brownholtz, eric_wilcox, david_r_millen}@us.ibm.com
+1-617-693-4235

**ABSTRACT**
This paper describes a new collaboration technology that is carefully poised between informal, ad hoc, easy-to-initiate collaborative tools, vs. more formal, structured, and high-overhead collaborative applications. Our approach focuses on the support of lightweight, informally structured, opportunistic activities featuring heterogeneous threads of shared objects with dynamic membership. We introduce our design concepts, and we provide a detailed first look at data from the first 100 days of usage by 20 researchers and 13 interns, who both confirmed our hypotheses and surprised us by reinventing the technology in several ways.

**Keywords**
CSCW, Computer-mediated communication, Activity-centric collaboration, synchronous/asynchronous collaboration, User study.

**ACM Classification Keywords**
H5.3. Group and organizational interfaces. H4.3. Communications applications.

## INTRODUCTION

Ad hoc collaboration systems, such as email and chat are lightweight and flexible, and provide good dynamic support for short-term communication needs. However, for those collaborative activities which extend over longer periods of time, or over larger numbers of participants, these media rapidly become unmanageable. At the other extreme, structured shared workspaces provide good support for making sense of large corpora of messages and files. However, these environments are relatively labor-intensive to initiate, and discourage people from using them for small-scale or short-term collaborations.

The result is often that a single collaborative activity is managed with multiple stores of shared resources and media, such as email, chat, wikis, discussion databases, listservs, and document management systems. This diversity of shared resources means that people must monitor and participate in multiple shared venues, dividing their attention and their effort across multiple media. Even if they are successful in this divided attention and context management task, they face difficulties in choosing which medium to use for any new collaborative activity: Most crucially, people need to predict the scale of each collaboration upfront, because it is difficult to move resources from one environment to another. For example, a brief conversation that begins as a chat may need to be transferred into email, so as to enfranchise a larger number of participants, or for the convenience of people who are in different time zones. Moreover, if the number of people or shared resources increases, then email may prove a chaotic venue for large-scale collaborations, and it becomes necessary to transfer the resources again into a structured discussion space or a document management system.

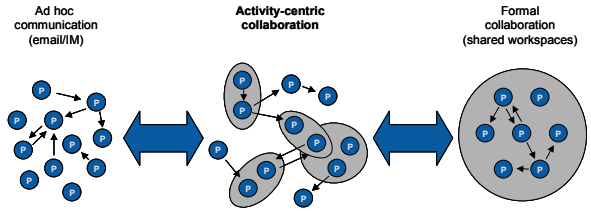



**Figure1: From ad hoc communication to formal collaboration**

Our research in the Instant Collaboration project [8, 9] investigates technologies that can help bridge these gaps between small vs. large numbers of participants, brief vs. extended collaborations, and informal vs. formal structures (see Figure 1). Our project has three goals:

- To provide a hybrid system that could, accommodate brief and informal interactions, but could also hold persistent resources that people could share and manage.
- To provide hybrid ways to structure collections of heterogeneous resources when needed, while allowing simple collaborations to proceed without the overhead of a formal structure.
- To support appropriate amounts of presence and awareness, so that collaborators have the social information needed to keep shared activities going,

without unnecessary intrusions into solitary activities, or interruptions from one collaborative activity into another.

This paper provides the first usage data from a system to support hybrid collaborative activities, whose design and implementation we reported earlier [9]. We provide a detailed look at the first 100 days of use by a community of 33 people. As will be seen, we have made some progress toward each of the aforementioned goals, and we have problems left to resolve for each, as well.

We begin our report with a summary of the underlying design principles of the client portion of our application, called "ActivityExplorer". We then describe the functionality of the system and we illustrate how people can collaborate in a lightweight, ad hoc manner but aggregate and organize different types of shared resources into larger collaborative activities with dynamic membership, hierarchical object relationships, as well as blended synchronous and asynchronous collaboration. The main part of this paper focuses on the user study and results, based on internal work with a group of 20 researchers and 13 student interns using ActivityExplorer over a period of three months. We present and discuss data collected from our server logs and from interviews with students.

## DESIGN PRINCIPLES

The design of our system was mainly driven by the desire to combine the lightweight and ad hoc characteristics of email and the rich support for sharing and structure in shared workspace systems. A more detailed discussion of the design can be found in [9].

### Object-centric Sharing

A key design decision in our system was to allow sharing of resources in a fine-grained way. The basic building block for collaboration in our approach is a *shared object*. Shared objects hold one piece of persistent information, and they define a list of people who have access to that content. Examples for shared objects are a message, a chat transcript, a shared file, etc. Sharing on an object level was motivated by the difficulty of predicting the scale of a new activity upfront when people start collaborating. Collaboration might be very short-term and instantaneous and involve only little amounts of data to be shared, few people, and few steps of interaction, e.g., exchanging one or more files, setting up a meeting agenda with people, or jointly annotating a document. These activities might or might not become part of a larger collaborative work process. However, people usually do not create heavyweight shared workspaces to perform these tasks.

### Object-Level Awareness

Our design deliberately does not make a distinction between asynchronous and synchronous types of collaboration. Each shared object supports real-time notifications that update other users' view of this shared object (if they are currently viewing or editing this object) or notify other users about current activity on this object. The latter provides fine-grained object-level awareness, i.e. people get to know who's working on what. Awareness about who's currently accessing an object, can serve as a trigger for opportunistic collaboration and can help keep collaborative activities moving forward.

### Activity and Conversational Structure

In order to be able to add structure to their collaboration, we allow users to combine and aggregate heterogeneous shared objects into structured collections as their collaboration proceeds. We call a set of related but shared objects an *activity thread*, representing the context of a collaborative activity. We see this structure being defined by users' ongoing conversation, i.e. each object added to an existing object can be considered as a "reply" to the previous one. We intentionally did not require any particular object as a structural container for a collaborative activity. Each individual shared object can function as the parent object for an activity thread, or for a subthread within a thread.

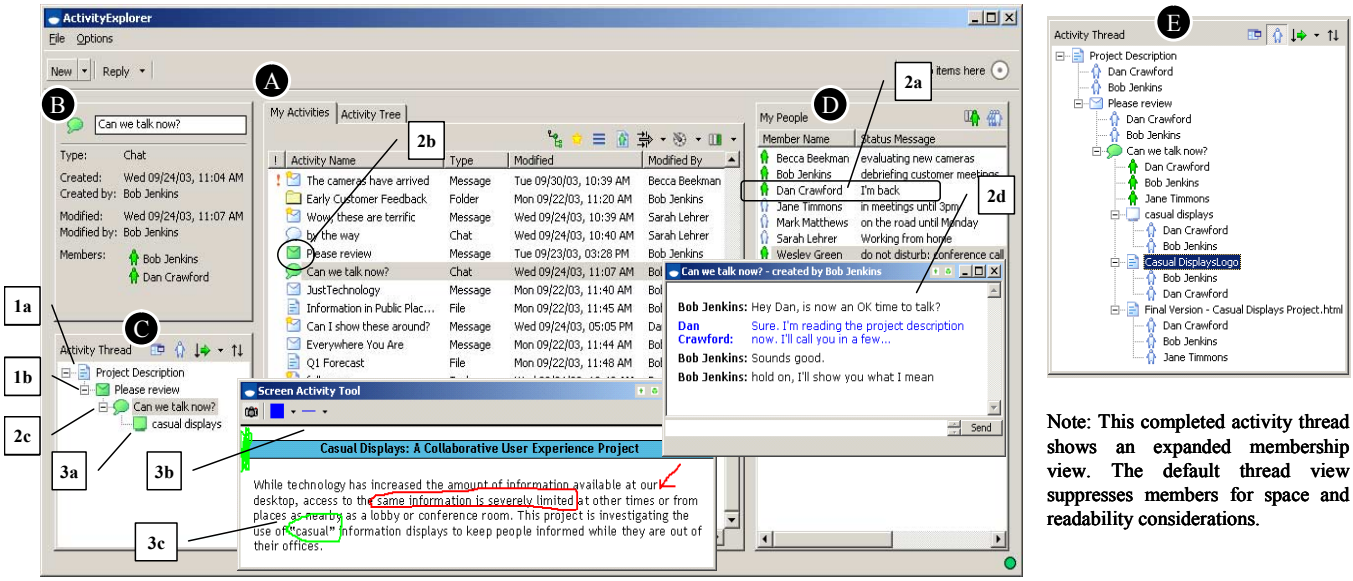
### Dynamic Membership

Membership in collaborative activities can be dynamic and heterogeneous. Activities often spawn side activities that involve a different set of people, or they might require bringing in new people or excluding people from certain shared resources [5]. For us, dynamic membership within an activity thread comes as a by-product of object-centric sharing, because each object has its own access control list. As collaboration proceeds, users may include new members, or exclude old members from selected shared resources in the thread.

In many regards our approach is similar to threads in email or discussion databases, or thrasks [1]. However, it is richer because (1) objects are shared (unlike in email or thrasks); (2) activity threads may contain different types of objects, not only messages like in email or discussion databases; (3) all objects are equal, unlike in email or discussion databases where attachments are subordinates contained in the message; (4) membership is dynamic and may differ within an activity thread from object to object unlike in discussion databases – and membership can be redefined after an object has been created, unlike in email; (5) objects support synchronous collaboration through built-in real-time notifications, providing rich awareness information.

## ACTIVITYEXPLORER

ActivityExplorer (AE), the client portion of our prototype, runs as a stand-alone desktop application. It supports fine grained access-controlled sharing of six types of objects: message, chat, file, folder, annotated screen shot, and to-do item. Users can create shared objects as a root object (the start of an activity thread) or as an addition to an existing thread (reply). As shown in Figure 2, activity structure and membership are managed by several UI components: **My Activities** (A) is a multi column "inbox-like" activity list, supporting sorting and filtering. Selecting a shared object in this list populates a read-only info pane (B). The **Activity Thread** pane (C), maps a shared object as a node in a tree representing an entire "activity thread." **Activity Thread** and **My Activities** are synchronized by object selection.




Note: This completed activity thread shows an expanded membership view. The default thread view suppresses members for space and readability considerations.

**Figure 2: ActivityExplorer**

Additionally, **My People** (D) lists members by name, email and status message. Users interact with objects or members, as displayed in these views, through right-click context menus. Representative icons are highlighted green to cue users of shared object access and member presence (2a, 2b).

The following scenario illustrates how shared objects as building blocks can be used to collaborate in an activity that starts from a document. Figure 2 is a snapshot of the activity in progress, shown from the perspective of one of the actors (Bob). The activity thread is built dynamically as the actors collaborate; see (E) for the complete thread.

***Bob** is a project lead who works with **Dan** on a project called "Casual Displays." **Jane** is a web designer in their company who is responsible for the external web site. The deadline for publishing new content to the site is approaching. Bob has written a project description to put on their web page and would like Dan's feedback. Bob glances at My People for Dan's name, and learns that Dan is offline (his name is grayed) and he will be available later (Dan's status message states he is at a midmorning appointment). From his desktop Bob drags the file on to Dan's name, starting a new activity thread (1a). The file is now shared, and Bob right clicks on the file object in his list to add a message asking Dan for his comments (1b).*

*A few hours later, Dan returns to his desktop (2a).A message in the system tray alerts Dan to the new activity. Clicking on the alert, he is taken to the activity thread. He opens the message. While Dan is reading it, Bob can see Dan is looking at the message because the shared object is lit green (2b). Bob seizes the opportunity; right clicking on the initial message to add a chat to this activity (2c). A chat window pops up on Dan's (2d). Bob refers to a detail in the project description; for clarity he wants to show Dan what he would like changed. By right clicking on the chat object, Bob creates a shared screen object (3a). A transparent window allows Bob to select and "screen scrape" any region on his desktop. He freezes the transparent window over the project text. The screen shot pops up on Dan's desktop (3b). Bob and Dan begin annotating the web content in real-time, like a shared whiteboard (3c). As they discuss a few changes, Bob also asks Dan to integrate a project logo into the web page. Dan agrees but is pressured now to run to a meeting. He assures Bob he will check with him the following day. Dan closes all his windows; subsequently his name turns gray throughout all of his shared objects displayed on Bob's client. Next, Bob drags the logo file from his local file system and the shared file object becomes part of Bob's and Dan's activity thread.*

*The next morning, Dan returns to his office and begins work on the changes. He includes the logo, which he sees that Bob had added to their activity. In a few hours, he has reworked the page. Noticing that Bob is online, Dan resumes their prior chat (which, as a persistent object, has remained part of their activity thread, ready to be re-opened for additional content). He tells Bob that the work is done. Aware of the pending deadline, Bob wants Jane informed. Within the chat, he selects **Invite** to add Jane as a member. On her client, Jane receives a pop-up invitation to join the chat and she accepts (note that Jane is a member of the chat only, and not of the other objects in the activity). In the last remaining steps of this activity, Dan shares his final versions of the content. Bob approves and Jane begins her work of publishing the new web page (E).*

## RELATED WORK

Our notion of activity-centric collaboration based on shared objects has been inspired by previous work in this area. This includes a variety of studies on email and activities, collaborative improvements to email, and shared workspace systems. The use of email has been widely studied and research indicates that email is a place where collaboration emerges [5, 6, 18, 25]. In [5], for example, Ducheneaut et

al. report on dynamic membership in email work activities and on informal activities evolve to more formal ones. Bellotti et al. [1] introduce the notion of a "thrask" as a means for better organizing email activities. Thrasks are threaded task-oriented collections that contain different types of artifacts such as messages, hyperlinks, and attachments and treat such content at an equal level. Along the same lines, Kaptelinin [14] presents a system that helps users organize resources into higher-level activities ("project-related pools"). The system includes not just email, but also any desktop application (see also [3, 4, 21] for the need to combine heterogeneous resources related to a single topic). While thrasks and Kaptelinin's work are similar to our activity threads, they are not shared and they lack any awareness of how others are manipulating the artifacts. By contrast, the Haystack project provides integrated presentation of diverse media that are assumed to be shared, such as email, IM, appointments, and web pages [21]; our approach goes further, adding objects that are usually *unshared* (files, to-do items), and showing *structured relationships* among those objects.

Our use of persistent chat derives in part from Haystack [21] and similar projects, and also from a long-term study of the use of multiple, topic-specific persistent chats in the Babble project [7]. Unlike simple persistent chats, we support the use of chats and other diverse resources, and we show structured relationships among those resources.

Several collaborative systems implement replicated shared objects and "collaborative building blocks" similar to our prototype. Groove [11] is a peer-to-peer system which features a large suite of collaborative tools and email integration, with collaboration centered on the tools placed in a dedicated workspace (e.g., all shared files appear in the shared files tool). In contrast, our system focuses collaboration around artifacts, which can be organized in a single connected hierarchy of different types. Kubi Spaces [16] also offers replicated shared objects, with a richer suite of object types, including to-dos, contacts, and timelines. Microsoft Outlook [19] has a server-based "shared email folder" capability. Lotus Notes [17] offers a replicated object architecture, although typically email objects are copies, not shared amongst different users. Collaboration in these "shared email solutions" is entirely asynchronous (e.g., users cannot work simultaneously on a whiteboard in real-time) with no real-time presence awareness.

## RESEARCH METHODS

The experimental prototype was used by 33 people, including 20 researchers and 13 interns, during summer 2003. We began our study by encouraging intern-mentor pairs to use AE as a tool to prepare their initial project proposal at the beginning of their internship, but we left it open to them to use it for all sorts of activities over the summer. The data for this report came from the period 1 April to 9 July 2003. We continue to use AE, and we anticipate further reports as we answer questions that were posed by the present analyses. We used two main sources of data: server logs and ethnographic interviews.

### Server logs

The first type of data was the server log for all users for the first 100 days of use of AE. The server logs gave us a holistic view of all activities by all users, and of their interrelationships. Our server log data described each object in each thread in terms of the following data attributes: **name** of object, **thread** in which the object occurred; **member**(s) of the object; **creation** (author, date, time); most recent **modification** (author, date, time); and most recent **reading** to the object (reader, date, time). Except for chats, each object functioned as a unitary contribution to a thread. Hence, for chat objects, we also extracted the number of **chat turns**.

### Ethnographic interviews with interns

Our second source of data was interviews with the student interns. We talked with seven of the 13 interns about their experiences with AE. These interviews were conducted by one member of the team, and took place in each intern's work area. Interns could illustrate their points by bringing up a particular activity thread or object on the screen, or by showing a resulting paper artifact.

## RESULTS

According to our design philosophy, we expected AE to provide a hybrid environment for collaborative work. Our quantitative and qualitative data support this claim. We begin our Results section with a general description of the nature of work in AE, including some analyses of the peculiarities of observing and analyzing collaborations in an environment that is characterized by heterogeneous contributions, interactions that can span minutes or months, and dynamic membership. We follow with a comparison to other collaborative environments, showing quantitatively and qualitatively how AE provided a mid-point between the very informal world of chat and email, vs. the relatively formal world of structured discussion databases. We then examine several emergent categories of collaborations that we have observed in this new environment.

### General Description

Table 1 provides a summary of usage at the conclusion of the 100-day study period. (Table 1 analyzes the data in terms of threads; by contrast, Table 2 analyzes the data in terms of objects – i.e., the components of the threads. The difference between these two analytical perspectives will prove important, below.) The server contained 1487 objects that had been created in 203 threads. A total of 33 people used AE during the study period.

#### *Activity Threads: Length, Duration, and Membership*

Analyzed as a group of 203 activity threads, the **length** of the threads was a mean of 7.02 objects (median = 1, SD = 18.58). The **duration** of the threads (i.e., the period during which a thread received on-going contributions) was a mean of 7.17 days (median = 3, SD = 10.00). The **membership** of the threads was a mean of 3.04 participants

| | Length [a] | Duration [b] | Membership |
|---|---|---|---|
| **Mean** | 7.02 (26.13) | 7.17 (11.43) | 3.04 |
| **Median** | 1 (5) | 3 (3) | 2 |
| **St.Dev.** | 18.58 (58.29) | 10.00 (18.43) | 18.43 |
| **Range** | 1-170 | 0-64 | 1-28 |

**Table 1. Activity Threads in AE**

a. **Lengths** without parentheses are based on a count of *objects*, counting each chat as a single object. Lengths within parentheses are for *turns*, counting each object and each chat input as a turn.

b. **Durations** without parentheses define duration as first-write to last-write. Durations within parentheses define duration as first-write to last-read.

in each thread (median = 2, SD = 18.43).

Although seemingly obvious, these simple statistics underestimate two critical attributes: the length and duration of threads. In our initial statistics, we calculated thread **length** by counting each chat as a single object. This approach was consistent with determining the thread length in email discussions or databases. In this perspective, each object is seen as a single turn in an on-going conversation of messages, files, tasks, and so on. But of course chats contain their own internal structure of turn-taking. We therefore reanalyzed threads that contained chats, counting as a "turn" each *non-chat object*, and also each *separate message in the chat*. By these measures, the thread length in AE is more accurately described as a mean of 26.13 turns (median = 5, SD = 58.29). Counting chat messages as turns extends our estimate of thread length by about 19 turns.

The second underestimate concerns our analysis of thread **duration**. In our initial statistics, we treated a thread as active, as long as people were *contributing* to it (i.e., adding new objects or modifying existing objects). However, a second measure of duration uses the last time that an object in the thread was *read*. Objects in threads were read days after the last update. Measured from first-write to last-read, the mean thread duration was 11.43 days (median = 3, SD = 18.43). Calculating from first-write to last-read extends our estimate of thread duration by about 4 days.

All of these distributions were very positively (rightwardly) skewed (the skewness was the source of the large standard deviations). Visual inspection showed that these distributions were multivariate multi-modal, with a tight cluster of many small threads (few objects, brief duration, small number of members) and a more diffuse spread of larger threads (many objects, days, and/or members). As we will show, these multi-modal distributions are evidence of a diversity of uses of IC – some of which surprised us.

### *Objects: Duration and Membership*

The preceding description was focused at the relatively macro-level of analysis of activity threads – i.e., collections of objects organized by temporal sequence and by parent-child relationships. We now pursue a more micro-level analysis of individual objects within the activity threads.

| Type of object | *n* | First-Write to Last-Write (days) | First-Write to Last-Read (days) | Member-ship (persons) |
|---|---|---|---|---|
| **Chat** | 174 | x=1.81 m=0 | x=9.20 m=1 | x=2.61 m=2 |
| **File** | 348 | x=1.55 m=0 | x=16.02 m=7 | x=7.75 m=3 |
| **Folder** | 65 | x=2.43 m=0 | x=15.78 m=10.5 | x=7.35 m=3 |
| **Message** | 753 | x=3.73 m=0 | x=22.24 m=22 | x=12.01 m=6 |
| **Screen** | 67 | x=.45 m=0 | x=10.28 m=6 | x=4.84 m=2 |
| **Task** | 80 | x=23.15 m=22 | x=32.03 m=30 | x=7.43 m=10 |
| **All Objects combined** | 1487 | x=3.83 m=0 | x=18.99 m=13 | x=9.14 m=4 |

**Table 2. Objects in AE (x=mean; m=median)**

Activities were composed of up to six types of objects. Table 2 provides a summary of the usage of objects across the 203 activity threads. Messages were by far the most frequently used type of object, and (with tasks) were among the two object types that were used the longest. Messages also had the largest membership of any object type. Thus, much of the work done in AE was done via messages, and in fact there were both long and short threads that were composed entirely of message objects. However, there are several important qualifications to this observation:

- **Files** were the second most-often used object type, and had the second largest per-object membership. Much of the non-message work of AE was done via shared files.
- **Tasks**, despite their low usage rate, were used the longest on a per-object basis – this reflects their on-going value for coordination of work.
- **Chats** were used in a diffuse but pervasive way (see "Objects within Activity Threads," below). Much of the communication or articulation work of AE appears to have been done with chats.

Interestingly, folders were used least, and fell in the middle of the range of use-duration and membership. We had originally thought of folders as non-sequentialized containers for other types of objects. However, any object could be used as a parent (similar to a container) for other objects, and our users may have decided that folders were not needed in this inherently tree-structured environment.

### *Objects Within Activity Threads*

However, these analyses of objects have been presented without regard for their position within the context of activity threads. As we discovered, this kind of element-wise analysis can obscure the important role played by the less-frequently used chat objects.

| | Length | Duration | Membership |
|---|---|---|---|
| **AE** | 7.02 (26.13) | 7.17 (11.43) | 3.04 |
| **Email [15,20]** | 2.8 | ~ 3 | ? |
| **Instant Messaging [12,13]** | 14.31 | < 1 ? | 2 ? |
| **Text Messaging[10]** | 3.58 | < 1 | 2 ? |
| **ChatRoom [23]** [b] | 10 | n.a. | n.a. |
| **Discussion DB [22,24]** | 4.3 / 4.86 | 33 | n.a. |

**Table 3. Comparison of AE with other collaborative environments (means)**[a]

a. We report our figures in this table as means, despite the skewness in our data, because the data from the available studies was expressed as means.

b. Note: ChatRoom results are from experimental sessions, not long-term use.

When we analyze types of objects in terms of their inclusion in activity threads, a different pattern emerges. These two contrasting patterns are shown in Figure 3. The top portion of the Figure repeats the frequency-of-use data from Table 2, showing that messages were used more frequently than any other type of object. By contrast, the bottom of Figure 3 shows the percentages of the 203 threads that contained each type of object. More threads contained chats than they did any other type of object. Thus, we begin to see chats as either (a) important modes of sharing on their own (e.g., 63% of single-object threads are chats), or (b) as important explanatory or articulatory modes to make sense of other objects. These statistical conclusions receive further support in the discussions of "Heterogeneous Collections" and "Conversations," below.

### User Experience: Hybrid of Informal and Formal

In [9], we hypothesized that AE would occupy a middle range between more formal, structured environments, and less formal, unstructured environments. Our experiences forced us to re-evaluate that hypothesis: AE had become much more versatile than a point solution in this space.

Table 3 compares a summary version of the statistics for AE activity threads with similar statistics from other collaborative environments. The length and duration comparisons put AE into a middle position among the other collaborative environments. The membership data (where available) appear to put AE into a more extreme position – although we suspect that membership for email and discussion databases, if available, would be larger than for AE, thus returning it to a middle position.

As we noted above, the heterogeneous, compound nature of AE threads makes it important to consider multiple analytic perspectives. Although the object-by-object perspective on the length of AE threads falls in the middle of the length dimension of Table 3 (7.02 objects), the turn-by-turn perspective places AE in a more extreme position on the length dimension (26.13 turns).

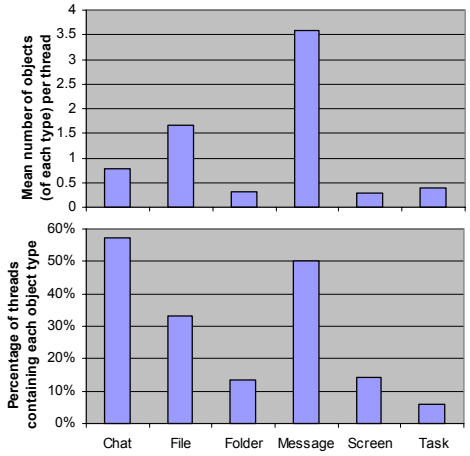



**Figure 3. Frequency of use of objects, and frequency of inclusion of objects within threads**

We also note the large variance in the measurements for AE along the length, duration, and membership dimensions (see Table 1 for standard deviations). These high variances, coupled with the extreme rightward skewness of all three distributions, suggests that AE covers a very broad *range* of values along each dimension. Thus, we appear to have come close to our goal [9] of designing an *average* experience between established formal and informal environments. However, the high variances suggest that AE has become a *hybrid* environment, partaking of some of the attributes of both informal and formal environments. The next subsections explore some of the sources of that variance in the diversity of uses that people found for AE.

### Living in AE

Ducheneaut and Bellotti described the user experience of "email as habitat" [6]. The interns in our study had a similar experience that might be called "AE as habitat."

Each intern was a temporary member of IBM Research. Interns had available the standard communications tools of our site, including Lotus Notes for email, file sharing, and discussions, and Sametime, IBM's Instant Messaging (IM) product for chats and e-meetings. Without giving the matter much thought, we anticipated that interns would work with other interns in the same way as in previous summers. That is, we expected interns to develop their own email distribution lists, and to form an interns' email-based and IM community for social and organizational purposes. Instead, we found that the interns had taken over AE, where they constructed several very long threads that served them as community resources. One of these threads, "Interns Tips and Tricks," contained 74 objects by the end of the study period, and continued to grow after the close of the study.

Interns' self-reports during the interviews were consistent with the server log data. One intern remarked, *"I never have used less email."* Other interns told us about their ongoing use of AE as a reliable, always-on medium: *"I kept it on all the time to coordinate with [my mentor]." "It's sometimes interesting to see what people are reading about" "I liked the address book – the people list and their*

*photos."*[1] Several interns said that they had never loaded the company-standard IM client, feeling that AE gave them all of the workplace chat capability that they needed.

### Heterogeneous collections

We had hypothesized that people would make use of AE's ability to store heterogeneous objects together in a single, threaded activity representation. For activity threads of more than one object ($n$ = 90, or 44% of all threads), 73 threads (81%) contained more than one type of object, and in fact the modal number of *types of objects* per thread was 3 types of objects. Thus, for those threads of length>1, people overwhelmingly chose to use heterogeneous objects.

Intern's self-reports again supported the quantitative results from the server logs: *"We would store chats with the objects that they were about." "In the [development folder], messages and chats on a related topic or subtopic appeared together with screen-shares… useful, especially in UI development, showing updates…. Screen shares usually followed a message, 'I've updated this and that.'"*

### Conversations through Objects

We had also hypothesized that people would make use of diverse objects in a conversational manner – e.g., a question posed in a message might be answered with a file; a shared screen might be commented on in a message; or a planning document might give rise to a list of tasks. Again, the data supported our hypothesis. We analyzed threads of length>1, for unique sequences of objects within parent-child-grandchild-great-grandchild-etc. chains. Within these chains, we analyzed all sequential *pairs* of objects (737 pairs). 337 object-pairs occurred *between different users* (i.e., were conversational), as contrasted with 400 object pairs that occurred *from one user to the same user* (e.g., a person would post a file and then post a message commenting on the file; or a mentor would create a folder, and then put readings for her/his intern into the folder). Of the 337 conversational pairs (46% of all object-pair sequences), 158 pairs (47%) involved *different types of objects* within the pair. Thus, roughly half of the object-to-object sequences within the long threads were conversational, and roughly half of these conversational sequences were comprised of objects of differing types.

Interns' self-reports again provided convergent evidence: *"I wrote a set of slides and put them [as a file] in [AE]. [My mentor] put comments in a message. I wrote a new set of slides." "I created my [slides] and sent them to [my mentor]. I got his comments… in a reply message [a child object of the slides], and I made changes."*

Most other collaborative environments restrict people to a single type of object (e.g., an email or a discussion entry or a chat text), and require that other types of objects be attached or enclosed within that type of object. Our AE experiences show that people readily mix object types when that is possible, and suggest that this equal-prominence of object types should be a feature of other collaborative environments (see [1, 3, 4, 21] for convergent concepts).

### Awareness Issues

AE provided several awareness features. Icons were provided for both users and objects. When users were running AE, their icons' appearance would change to green to indicate their presence. When someone was accessing an object, the icon for that object would change to green. Thus, it was possible to see people enter and leave AE, and to see objects being used and then falling out of use.

Each access to an object was sent as a notification to all AE clients, and was shown as a status-update message in the system tray. Interns' reports of these features were mixed. Positive comments included: *"I like being able to know who is looking at something right now. It's cool to see that an object is green." "[my mentor] is my code source… I had three persistent chats with [my mentor]. It's faster, simpler in [AE] – I can see his little green penguin."*

However, we had designed the alerts for use by relatively small groups. We had not anticipated that the interns would create large threads with many members. These large groups led to a sometimes constant stream of alert messages, which could become a problem: *"I turn alerts on for a couple days max after I post something, and then I turn them off." "I wanted to be informed only for new postings." "I would like to select what I would be informed about…"* These critiques and requests eventually led to the design of new features that helped people to control the nature and the volume of the alerts that they received.

## CONCLUSIONS AND LESSONS LEARNED

Conducting this study was an extremely encouraging experience for the research team. While the quantitative and qualitative data collected during our study confirmed that we made substantial progress towards our goal of providing a new hybrid environment between informal, unstructured and more formal, structured collaborative tools, we were pleased to see how readily the interns made this new environment their "habitat". This was unexpected, in particular in the presence of other tools such as email or IM.

However, "living" in this environment also pushed its boundaries. In addition to using AE to coordinate their summer projects with their mentors, the interns also created large community threads in the spirit of a discussion group. As we reported in "Awareness Issues," above, our alert-generation strategy had been designed for small groups, and turned into an annoyance for members of large threads. We eventually provided better features to control these alerts.

On the other hand, object-level awareness also proved to be quite useful for smaller threads. One of the interns worked in a development team that used a source control system in

[1] The address book was a spontaneous intern project, actively promoted by one of the interns. It contained digital photographs and biographies of the interns -- a kind of "who's who" database used by both interns and others.

addition to AE to coordinate their development work. In addition to using AE for brainstorming and discussing design issues, they would also coordinate their "check-ins" into the source repository with messages in AE that triggered alerts. This helped them in particular when work became highly parallel towards development deadlines.

Our data indicate that the other design principles of activity-centric collaboration worked well. We observed that people made frequent usage of shared objects in heterogeneous threads, and that they used these objects in a conversational, turn-taking structure. In addition to the aforementioned large public threads, activities performed in AE ranged from many smaller threads involving few objects used to collect feedback, gather information, or solve small problems, to larger threads that served as project spaces for intern-mentor pairs. However, the data from the first 100 days was insufficiently detailed to test our prediction that people would use dynamic heterogeneous membership in activity threads. Anecdotes among researchers (but not among interns) suggest that people used this capability; our on-going studies use a richer design of the server log to allow us to investigate dynamic membership in greater detail.

Previous research indicates that collaboration can be an emerging process, from highly unstructured work to more formal, structured workflows [2, 5]. Although we were not looking into this aspect in particular, we have found AE to be an intriguing platform to gain more insights into how activities emerge: AE lets people start collaborating in an ad hoc way but also supports gradually adding structure by combining shared objects into larger threads. We are currently examining our data to learn more about how people structured their threads and if there were any patterns present in the use and transition of shared objects.

Users accomplished a great deal in AE, despite the limitations of only six object types and the burdens of an over-active alerting mechanism. They showed us ways to support of diverse activities, from quick questions to long-term community resources. We look forward to learning more as we extend AE concepts into new architectures.

## REFERENCES


1. Bellotti, V., Ducheneaut, N., Howard, M., Smith, I., "Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool," *Proc. CHI 2003*.
2. Bernstein, A., "How Can Cooperative Work Tools Support Dynamic Group Processes? Bridging the Specifity Frontier," *Proc. CSCW 2000*.
3. Boardman, R., "Workspaces that Work: Towards Unified Personal Information Management," *Proc. HCI2002*.
4. Churchill, E.F., Trevor, J., Bly, S., Nelson, L., Cubranic, D., "Anchored Conversations: Chatting in the Context of a Document," *Proc. CHI 2000.*
5. Ducheneaut, N., Bellotti, V., "A Study of Email Work Activities in Three Organizations," working paper, PARC, http://www.sims.berkeley.edu/~nicolas/
6. Ducheneaut, N., Bellotti, V., "E-mail as Habitat: An Exploration of Embedded Personal Information Management," *ACM interactions*, 9(5), 2001, 30-38.
7. Erickson, T. & Kellogg, W. "Social Translucence: An Approach to Designing Systems that Mesh with Social Processes." *ACM TOCHI*. Vol. 7, No. 1, pp 59-83, 2000.
8. Geyer, W., Cheng, L., "Facilitating Emerging Collaboration through Light-weight Information Sharing," *Conference Supplement ACM CSCW 2002.*
9. Geyer, W., Vogel, J., Cheng, L., Muller, M, "Supporting Activity-Centric Collaboration through Peer-to-Peer Shared Objects," *Proc. Group 2003*.
10. Grinter, R.E., Eldridge, M., . "Wan2tlk?: Everyday text messaging," *Proc. CHI 2003.*
11. Groove Networks, http://www.groove.net.
12. Handel, M., Herbsleb, J.D., (."What is chat doing in the workplace?," *Proc. CSCW 2002.*
13. Isaacs, E., Walendowski, A., Whittaker, S., Schiano, D.J., Kamm, C., "The character, functions, and styles of instant messaging in the workplace," *Proc. CSCW 2002*.
14. Kaptelinin, V., "UMEA: Translating Interaction Histories into Project Contexts," *Proc. ACM CHI 2003*.
15. Kerr, B.J., "Thread arcs: An email thread visualization," *Proc Infovis 2003*.
16. Kubi Software, http://www.kubisoft.com
17. Lotus Notes, http://www.lotus.com/notes
18. Mackay, W. E., "More Than Just a Communication System: Diversity in the Use of Electronic Mail," *Proc. CSCW'88*.
19. Microsoft Outlook, http://www.microsoft.com/outlook/.
20. Moody, P. [Personal communication, 7 July 2003]
21. Quan, D., and Karger, D.R., "Haystack: Metadata-enabled information management. *Proc UIST 2003*. See also http://haystack.lcs.mit.edu/
22. Schoberth, T., Preece, J., Heinzl, A., "Online communities: A longitudinal analysis of communication activities," *Proc HICSS 2003*.
23. Smith, M., Cadiz, J.J., Burkhalter, B., "Conversation trees and threaded chats," *Proc CSCW 2000*.
24. Yates, J., Orlikowshi, W.J., Woerner, S.L., "Virtual organizing: Using threads to coordinate distributed work," *Proc HICSS 2003*.
25. Whittaker, S., Sidner, C., "Email Overload: Exploring Personal Information Management of Email," *Proc. CHI'96*.