Revised Thesis brainstorming ideas:

Fundamental Assertion:

Tools that support user productivity must be based on the dynamic interaction between the user and the information, rather than just the content of the information.

Outline:

1. Introduction and fundamental assertion
2. Definitions
3. Theoretical framework: User context
4. Empirical findings: initial interviews about patterns of email/phone/meeting use. Cite other studies.
5. Resulting software design and new findings: Designs of Common Lens and Argus. Key findings with users.
6. Experimental tests???
7. General design implications
8. Conclusions: moving toward co-adaptive systems. Look at cooperative work implications -- making tradeoffs based on the behavior of two or more people.

Other relevant stuff:

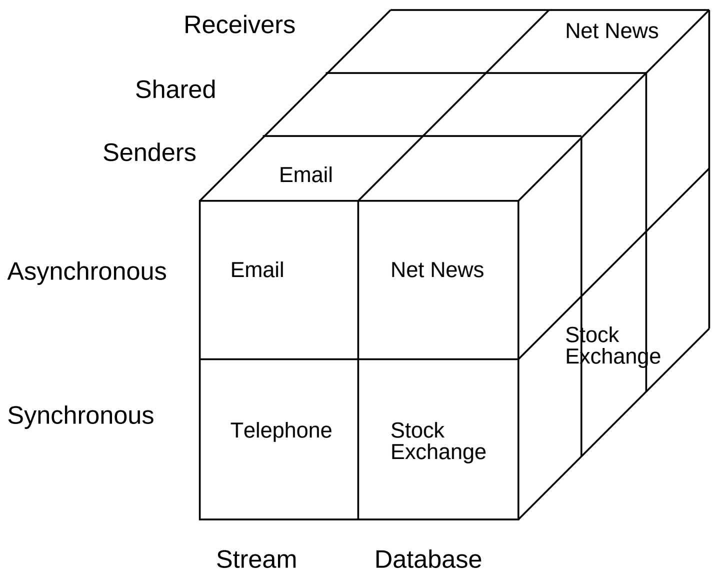

Figure 1: A User-oriented View of Electronic Messages

Particularly suited to communication systems, because they are very dynamic, very diverse from user to user, reflect overall work patterns.

Other systems, such as lens, are based on the idea that the content of the information/message is sufficient for rules. Also, it's rule based.

Challenge both. When are rules important, when are neural nets? When does user control, when does system, when is it shared? When does the appropriate action depend solely on content, solely on user's current state, and on a combination? How to detect patterns, embody them, make trade-off decisions/strategize, evaluate effectiveness and design tools. Predict the coming of co-adaptive systems. Focus on optimizing trade-offs based on user's preferences (which may well be influenced by corporate or other external pressures), not on a single optimal solution.

How does Argus reflect this difference in perspective?

Implications:

Design systems that capture the user's process of use over time and feed that information back to him/her. Allow the user to then modify the tools based on that information. Result: a coadaptive system...

What are the characteristics of Co-Adaptive systems?

Definitions

Finding examples of them

Creating them

Observing them

Predicting what happens with them.

Argue that a key characteristic is that users can modify them -- either by design or through accident. Argue that feedback loops are important. Also, dynamic situations will increase speed of co-adaptive evolution.

Argue that you can't design your software to be co-adaptive, explicitly. Standard UI development techniques still act as though the final software is static -- a snapshot in time. They're better than not doing it, because they provide a view of the process of a single user using the software over a discrete period of time under specified conditions. But what if the user's use is changed due to other users, changed understanding, new kinds of situations, changes in related technologies, etc.

Can we predict what kinds of systems are most likely to be co-adaptive? Management Information Systems. Electronic communication systems. What else? Educational systems? Anything that has users operating with a complex, flexible system, that can be modified.

The myth that software is ever "done" -- it's not like a book, that can be placed on a shelf and still "work" years later. Many dependencies, on things that are based on shifting assumptions.

But, concentrate again on systems in which the content of the information exchanged changes the use of the system -- and thus results in co-adaptivity. Example -- you can't force people to use email. What will cause people to use email is if: a. it's easily accessible and b. it carries information that's considered to be of value. If one or the other condition isn't met, people won't use it. Once it is assumed to be used, the determination of "value" will change -- since the user has already absorbed the initial cost of accessing the information, he/she may be more willing to deal with other kinds of infomration over that channel.

Implication for videomail system -- make sure that there's a very good reason for people to use it (eg SIGGRAPH or neuroanatomists -- people who rely on images as part of their daily work). This will develop along unpredictable lines.

Can we draw an evolutionary analogy from S.J. Gould? The idea that there are a limited number of possibilities for vertabrates to evolve into, because there are switches on the chromosomes. Similarly, if you have underlying assumptions in your system, it will affect the range of possible adaptations. Thus, users of supercard will use it in some predictable ways -- and it will evolve as influenced by the applications it's designed to support. The clash between application assumptions and supercard assumptions should provide some "innovations".

List of Figures