

INTERACTING IN CHAOS



Photo/collage by Jack L. Moffett

DAN R. OLSEN, JR.

User interface software and interactive systems have reached a crossroads. After two decades of work on interacting through the desktop metaphor, we must now look into the future to find new problems whose solutions will equip users in fundamental ways. This paper is my personal view of what those issues are and where interesting research questions are to be found.

The design decisions made in 1984 still dominate our personal computing environments, but they no longer reflect the way in which computers are actually used. The fundamental assumptions of interactive computing must be reconsidered. We now use personal computers to deal with large amounts of information, many people across the Internet, and an increasing variety of computing platforms. The chaos induced by the diversity of information, collaborators, and interactive platforms is the main issue I will explore in this article. To resolve the problems created by such diversity we must focus on representations that are naturally convergent. Such convergence will be driven not by computational forces or mandates for standards but by the bounds of human sensory and motor capabilities. Although the size and diversity of computing will continue to grow, human capabilities to use such facilities will remain constant.

Information Chaos

The access, collection, organization, synthesis, and communication of information are the primary uses of most computers. The calculation tasks that characterized earlier computing still exist, but their relative importance has been sharply diminished. The tools that we offer the computer users of the future

must support and facilitate these information tasks.

Information Usage Cycle

The following cycle appears to apply to most information.

1. **Chaos:** Information is available in a raw, unorganized form created for purposes other than the current intent. Someone recognizes order and usefulness in this chaos and begins to laboriously sift, organize, and synthesize.
2. **Unification and codification:** Having recognized the usefulness of this information someone begins to codify and regularize the data so it can be more effectively applied to its newfound purpose.
3. **Exploitation:** As a unified and codified information resource is created it is exploited in as many ways as possible to spread the costs of creating the new information resource.

The process is then repeated.

Current computer science practice as well as user interface design and development practice thrive on Steps 2 and 3 of this cycle. The next challenge lies in Step 1—the sifting, organizing, and synthesizing of information whose usefulness and structure are not yet recognized and still being formed.



Dan R. Olsen, Jr.
Computer Science
Department
Brigham Young University
olsen@cs.byu.edu
+1-801-378-7655



Photo/collage by Jack L. Moffett

The Challenge of Chaos

Exponential growth is the driving force of the future of interactive computing. Moore's Law forecasts that any computer today will have 100 times more memory and more CPU cycles in a decade. Such growth puts computing resources on the desktop that far exceed what is necessary for word processing, e-mail, or spreadsheets. Exponential growth in computing capability puts vast capabilities into very small devices. This not only reduces the form factor for personal interaction but also creates a potential for multi-sized personal computers and interactive buildings. The Internet is growing even more rapidly than computing hardware, introducing new sources of information and new potential collaborators. Dealing with the chaotic diversity of information, collaborators, and interactive devices is the principal interactive challenge of the next decade.

From Models to Chaos

Computer science typically assumes that for a given problem, an abstract model of all information relevant to that problem exists. If you create such an abstract model, you can define computational processes that can manipulate information in that model. This fundamental assumption leads to a software design process as follows:

1. Study a problem,
2. Identify the essential pieces of information for the problem,

3. Construct an abstract model for that information,
4. Build a computing solution around that model, and
5. Conform work practice to the model.

This modeled information world requires careful study to predict all of the eventualities and provide for them in the model. Any eventuality not provided for is discarded as not sufficiently important.

The problem with a modeled information world is that the real world of work is simply not that clean. Two situations are rarely identical. There are always exceptions and variations. A modeled approach requires that before a computing solution can be applied to a problem, an abstract model of the solution must be created. This "modeling first" strategy is not mirrored in practice. The process that occurs in the real world is more like the following:

1. A problem arises.
2. The user locates the information artifacts relevant to the problem.
3. The user identifies strategies that have succeeded earlier for similar problems.
4. The user adapts and applies information and prior strategies to generate a unique solution.
5. The user makes procedures from strategies that are repeatedly useful.

Modeling—and all of the techniques that accompany it—only comes into play when Step 5 becomes large and important enough to justify the costs of professional programmers. In many cases, problems never reach the point of justifying professional programming. The interplay of independent people generating and synthesizing information in many cases maintains a steady state of chaotic variation in the work. This chaotic view of work that attacks and solves problems as they arise rather than predictively modeling their solutions before they arise will come to dominate interactive behavior in the future. The following examples will illustrate the forces that produce such irregular information.

A traditional computing application usually involves a known set of users that one can study. Typically this set of users is bound together by a common organization or similar

sets of tasks. If one is trying to produce a new information service on the World Wide Web (WWW), this strategy will not necessarily help because the set of potential users of a service is unknowable other than by statistical generalities and the users have only indirect influence on the providers. Unlike a traditional “accounts payable” style of applications, there is no well-defined group of users and providers around which models can be developed. The Web community of providers and users is open and diverse, with few controllable connections between information creation and use.

As a second example of information chaos, consider the information needs of an investment adviser. The role of an investment adviser is to convert information into predictions of future market behavior and then to convert such predictions into buying and selling plans. If the adviser’s conversion of information into buy and sell strategies is successful, then she will become rich. Let us suppose that all stock advisers use the same information (just as all accounts payable clerks in a company or all users of the SABRE airline reservation system use the same information). If every stock adviser uses the same information and processes in the same way, nobody has an advantage. Any bright stock adviser will instead hunt for new useful information. By using new information that others are not exploiting, the adviser gains a competitive advantage. In such activities, the economic advantage goes to those who can find a new kind of order, regularity, and insight in the information chaos before them. Those who exploit known information in known ways have no such advantage. The power is in mastering the chaos.

Collaboration Chaos

The arrival of the Internet has converted personal computing from an individual chaotic activity to a collaborative one. Recent studies of Internet usage have shown that most time spent on the Internet involves communicating and collaborating with other people rather than accessing information [3]. This type of usage has large implications for the way in which interactive environments should be

structured. In today’s interactive environments, files and folders are easily manipulated on-screen entities. People, groups, and control of information sharing, however, are barely better than command-line interactions or buried in deeply nested dialogs and arcane e-mail naming schemes.

This lack of collaborative facility, of course, must change if we are to continue to make progress in providing services that meet the actual working needs of users. For the average knowledge worker, people are as important as or more important than files or documents. It is important that collaborative mechanisms such as messaging and file sharing become as important, visible, and convenient as files and folders are in the current environment. Collaboration must be:

- ◆ **Pervasive**, available for all information, at any time, with any other authorized user;
- ◆ **Uniform**, with standardized interactive mechanisms for messaging, sharing, discussion and change awareness that are similar for all types of information; and
- ◆ **Diverse**, never confining collaboration to special collaborative data types but rather supporting all information relevant to people.

Model-Based Collaboration

The essence of all collaborative software is the communication of shared information



Photo/collage by Jack L. Moffett

between two or more people. The form of that shared information is critical to the success of the collaboration. Many collaborative systems function by communicating the underlying model information of the application, as shown in Figure 1. This architecture is characterized by such activities as mailing MIME attachments and helpers or browser plug-ins. By treating the model data simply as a collection of bytes and shipping those bytes over the Internet, collaboration is possible for any application. Architectures for synchronous collaboration by replicating or providing centralized access to shared models have been implemented. In principle it would seem that such mechanisms will readily handle the diversity of information that one will find in the future. In practice, however, this approach is destined to fail.

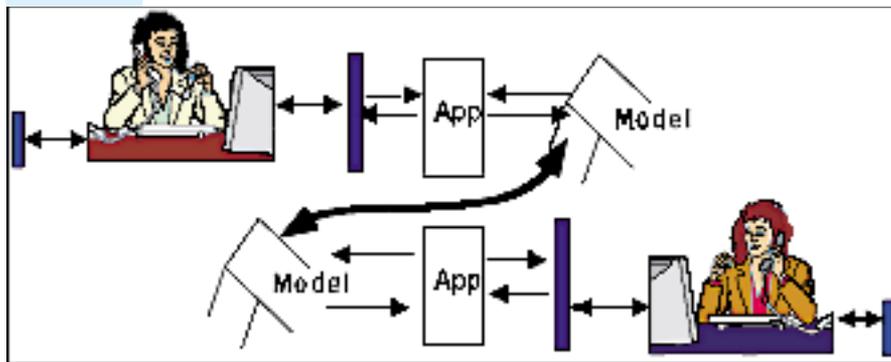


Figure 1. Model-based collaboration.

The principal deficiency of this architecture is the prerequisite that all collaborators have consistent versions of the software required for each piece of information. Without a viewer, browser, or editor that is consistent with the model format being shared, a potential collaborator is technologically shut out. In order for any new application to become collaborative, most of the potential collaborators must have a current copy. Whenever a large percentage of the desired collaborators do not have compatible software, most users revert to paper. In fact, studies show that paper dominates most interpersonal information exchange [8].

This incompatibility is inherent in the economics of software. No software vendor can stay in business without continually selling new features to the same customers. This

implies a continual round of new versions. As major versions change, file formats are no longer compatible with earlier versions. There is no economic alternative to this. In a large community using the ZorchTron product, any user that upgrades to ZorchTron version N+1 will produce messages incompatible with all users of ZorchTron version N. There are only three possible solutions. The first is to force all users in the community to upgrade at the same time. This is completely unrealistic in communities the size of the Internet. The second is to never change the ZorchTron file format. This forbids most progress in the ZorchTron product line. The third is to base collaboration among ZorchTron on something more uniform and convergent than the ZorchTron file format.

Application-by-application collaboration strategies also pose another problem. Suppose that you had an infinite software and support budget accompanying the perfect operating system. With this budget and operating system, the support staff would maintain a copy of every version of every piece of software on your personal machine without any failures. In this state you could receive any document in any format and bring it up in the appropriate application. Users in such a state, however, would hate the results. The problem is that with every new kind of message such users would face a new and unfamiliar user interface. The interactive chaos would be unbearable. This is akin to requiring knowledge of how to operate the sender's fax machine in order to receive a fax from him. Ultimately collaboration cannot be built around individual applications because it would overload human learning.

Device Chaos

Yet another complication in the future of interactive computing is the variability of future interactive devices. Computers of the future might be wall-sized, desk-sized, hand-sized, pocket-sized, or ear-sized. These configurations will also result in a wide diversity of interactive styles. We have become quite com-

portable with the screen, keyboard, and mouse configuration for interaction. This will change because of the plummeting cost of computing. When the PalmPilot or pocket pager of a decade from now holds 800 MB of RAM and runs at 1 billion instructions per second, the nature of interaction and collaboration will change. When 100-MHz computers with 64MB RAM cost \$10, the number of computers per user will change and with it all of the ways in which interaction and collaboration are performed. At \$10 each for a computer, the variety of such computers will be astounding. Throughout this chaos of devices users will expect integrated access to their colleagues and information. How is this to be done?

Mastering Chaos

The problem with such chaos is that the computing process itself thrives on uniformity and regularity. Without some regularity, computing solutions cannot be applied to the problem. Ultimately, the only power in computing is the ability to repeatedly perform complex manipulations. The power is in repeated use. Without some repeatable regularity, computers cannot help.

The Internet has shown that mandated standards should be used sparingly and be applied only where they enable a broad range of new communications and use. The path to regularity cannot restrict or confine the marvelous divergence in computing, but rather must be based on forces that will naturally produce convergence. By identifying the aspects of interaction that will naturally converge we discover fruitful targets for productive investigation.

The one constant throughout the history of computing is human capability. This includes the ability to perceive, assimilate, synthesize, and express information. Human eyes have not improved. Ears cannot hear higher frequencies. Fingers are neither smaller nor faster. Human memory has not increased. The number of spoken languages in common use has declined rather than expanded. The capabilities of human users define the fundamental convergent forces that can be exploited by future interactive systems to master the

chaos imposed by exponential growth. Human beings will ultimately reject any constraint on what they can and cannot do with computing, except constraints imposed by their own sensory and motor limitations.

Human-centric Information Is Not Truly Chaotic

The information people actually use is quite divergent in topic and content. However, the ways in which information is presented to people exhibits a high degree of regularity. Human-centric information not only tends to unify in format, but also in representation.

Convergence in Representation

One of the fundamental mechanisms for communication is human language. Whenever groups of people must regularly communicate, they converge on a common language. In the Middle Ages, educated people learned Latin, in addition to their native tongue. In the modern world the choice is English. In neither case was the unifying language particularly superior. The only requirement being that the language represents all of human



Photo/collage by Jack L. Moffett

thought and that one such language is chosen. The dominant constraint on language usage is the difficulty that most people have in learning languages. Only a relatively few gifted people can be fluent in more than two. The convergence to a lingua franca follows Speech Accommodation Theory [2], which states that communicating communities will tend to

unify their language in ways that make collaboration more efficient.

Uniformity also arises in the layout of documents and other information presented on the printed page. For a given writing system all information has a well-defined scan order (such as left-to-right, top-to-bottom). Indentation is widely used to represent hierarchies of topics and groups. Larger or bold type indicates important or high-level concepts. Objects are grouped by white space and by vertical and horizontal alignment. All of these techniques have converged (1) because of the way in which the human visual system naturally scans a scene and (2) because uniformity has improved communication among human beings. Schools regularly teach standard forms of information presentation, which facilitate communication among educated people. Modern-day windowing systems have shown that unifying the look and feel will simplify learning for new users.

As an example of how such regularity of information can be exploited, consider the table shown in Figure 2. This table shows the consumer price index (CPI) for the United States. This is a screen capture taken directly

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
1995	150.3	150.9	151.4	151.9	152.2	152.5	152.5	152.9
1996	154.4	154.9	155.7	156.3	156.6	156.7	157.0	157.3
1997	159.1	159.6	160.0	160.2	160.1	160.3	160.5	160.8

Figure 2. Tabular representation of consumer price index.

from the Web site of the Bureau of Labor Statistics. Based only on the screen capture (an image), any reader of this paper can immediately determine that the CPI for March 1996 is 155.7. We are all familiar with tables and how they represent information. Existing computer tools, however, cannot perform this same task. It is obvious that the

information is encoded in the geometry and characters of the image; however, our computing tools are currently incapable of extracting useful information from such an encoding.

From the table in Figure 2 you might want to transform the information into some other more appropriate form, such as the graph in Figure 3. To create the graph I typed all of the information from the table to Excel (a human-based transformation) and then requested Excel to automatically generate the graph. Once the information was encoded in the rows and columns of the spreadsheet, the Excel Chart Wizard could perform most of the transformation. However, the transformation from image to sheet is not complicated. All of the information required is encoded in the image. We just do not yet have the tools for extracting structured knowledge from pictures. Instead, our programs rely on special application-specific encodings as the basis for computing solutions. If our tools instead used geometric encodings, as people do, a much larger set of information sources becomes open to automatic processes. Some of the first steps toward this goal are in Tom Moran's Tivoli system [4] and Wilensky's work with multivalent documents [7].

The kind of wizard used to create the graph has its problems. One problem is that current technology requires all such wizards to be hard coded. It is difficult for ordinary people to make new wizards that will work on new kinds of informa-

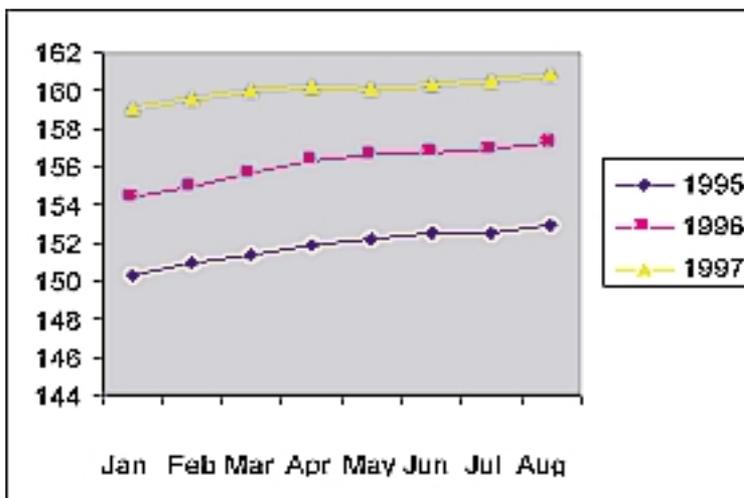


Figure 3. Graphical representation of consumer price index.

tion. Such wizards need to draw on programming by demonstration technology [9] to facilitate the bottom-up information usage described earlier.

People recognize structure in text as well as geometry. Figure 4 shows a screen capture from the online card catalog at Carnegie Mellon University (CMU). Although most readers have never seen this application, they can immediately identify the titles, editors, call numbers, and publication dates for the items presented. The information structure exists in a readily recognizable form. However, we do not have adequate interactive tools to perform transformations such as extracting this information into an EndNote database.

In Figures 2 and 4, the images that were the source of the information were not scanned from paper. They were extracted from Web pages. Obviously scanned images could yield similar results provided the image processing worked correctly. An important point, however, is that most information bearing pictures need not be scanned at all. The vast majority of our paper-based information was printed onto paper from a computer. In most cases that computer was connected to the Internet. If such information is already on the Internet, why is it being communicated on paper? The answer lies in the compatibility problem. Rather than risk the unreliable receipt of electronic information, people reduce information to paper, which always communicates. Paper has become the default interface between incompatible computer systems. Despite this situation we do not have the tools to exploit this obvious resource of digital information.

Lessons from UNIX

The UNIX environment deals very effectively with recognizing structure and new uses for information encoded in ASCII text. The underlying philosophy of UNIX is that all computable information can be encoded in a stream of bytes. In the

UNIX environment all commands are expected to read ASCII text from standard input and write ASCII to standard output. By unifying everything around ASCII text it was possible to build a wide range of pluggable tools that would store, extract, search, edit, and transform text. Because programs output readable text, users could readily see how some other program could manipulate such output for purposes not considered by the creators. This recognition of potential new uses in information, coupled with standard tools for text manipulation, is very powerful.

The UNIX model leads to a style of programming by which small components read text, manipulate it, and write text. This style of programming greatly facilitates the bottom-up exploitation that characterizes chaotic information. In some circles the “excessive” printing and parsing in UNIX were criticized as highly inefficient. One program would efficiently convert encoded binary data into text only to have the next program in the chain parse that text back into a binary form. The key point, however, was that because the data passed through a human-centric format (text) that was readily manipulated, it was possible for users to discover and exploit the information. Every UNIX system could display and edit text. UNIX also provided portability and encoding independence that were not possible in specialized encodings. People cannot exploit what they cannot see.

The problem with UNIX is that text alone is not sufficient for effective communication with human beings. Graphical user interfaces (GUI) replaced text with pictures as the primary interactive medium. Pictures, however, did not come with the same set of powerful search, storage, and recognition tools that had

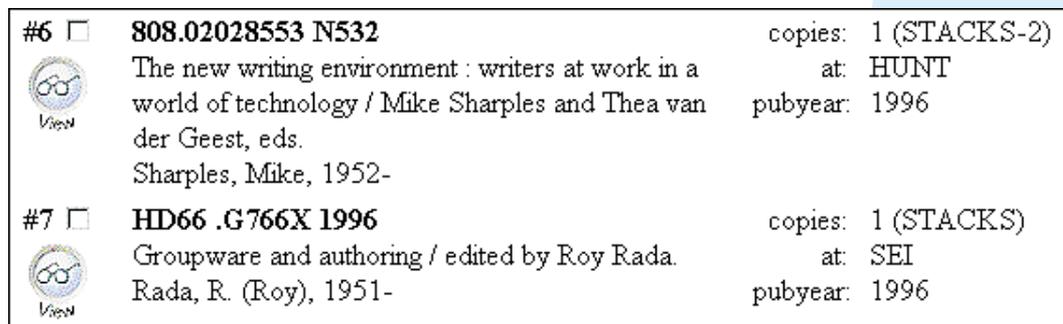


Figure 4. Example of CMU library catalog.

been developed for text. When GUIs were first invented, the processing power and memory required for manipulating pictures were not available. Moore's Law will resolve the resource problem; however, we do not yet have the necessary software technology. Applying the lessons of UNIX to human-centric data types other than text is a key research opportunity.

Convergence in Format

When considering the UNIX experience it is also instructive to look at the history of human-centric information. When UNIX was invented ASCII was by no means the only standard for encoding text. EBCDIC dominated the IBM world and DEC had its SixBit standard, which would reduce the amount of space required for a document by 25 percent. As memory and disks got cheaper and intercommunication became more important, all encodings except ASCII and its ISO-Latin extension died out. Similarly, there was a plethora of international encodings for efficient representation of non-Latin text. Gradually these are being replaced by UNICODE. UNICODE is less efficient in space than all of the others, but space for text is not an issue.

The main insight here is that diverse encodings of human-centric information survive only when the base technology is incapable of covering the full range of human capability. Since ISO-Latin covered the text needs of all Europeans, all other encodings for European text died. When the space overhead of 8 bit vs. 16 bit was important, many international encodings existed. When Moore's Law rendered the space question irrelevant, all but UNICODE died.

Consider audio files as an example. The dynamic range that the human ear can hear has a fixed limit. This means that more than a given number of bits per audio sample will be irrelevant because humans cannot hear the difference. The number of frequencies that the human ear can detect also has a fixed limit. Frequency and amplitude limits dictate the maximum number of bits required to represent 1 second of any sound that the human ear can hear. Exceeding those limits provides no competitive advantage for an alternative

encoding. If the base technology (disk, memory, and processor speed) cannot meet those limits, various compromise encodings will flourish to satisfy different needs and exhibit different competitive advantages. Once one or more alternatives exceed human limits in an economically viable way, the number of encodings quickly collapses to one or two. If human beings cannot discern the difference, nonstandard encodings become a liability rather than an advantage and lead to the demise of nondominant alternatives.

In seeking the natural convergence around which the information tools of the future can be built, the following **five** data types are of interest:

- ❖ **Text**, which represents human language;
- ❖ **Audio**, which represents hearing and speaking;
- ❖ **Two-dimensional (2-D) pictures**, which subsume text and add new pictorial and geometric visual relationships;
- ❖ **Video**, which is perceptibly different from still images; and
- ❖ **Three-dimensional (3-D) environments**, which make human perception of surroundings relative to one's own body different from simple video.

In terms of resolution, each of these representations has clear sensory limits. Today's technology has surpassed the limits of the first three data types, and their encodings are rapidly converging. Technological limitations still prevent video and 3-D environments from achieving convergence. Of all these data types, only text has achieved the level of power tools for manipulation that is needed to exploit chaotic information. Creating power environments based on the other four is a compelling opportunity.

Other sensory-based data types are excluded from this list, such as touch, smell, or direct stimulation of the nervous system. These are excluded because we do not yet have useful technology based on these senses. Experimental devices exist but they are in no way effective for representing and communicating information. Without significant breakthroughs they cannot influence the information and communication chaos of the next decade.

Human-human collaboration need not be chaotic

The chaotic problems of collaboration have many of the same characteristics as those described for information. However, collaborative technology has yielded several successes that support the focus on human-centric data. The most successful collaborative technology in history is the telephone. Unlike most computer applications, the telephone has undergone several decades of radical technological upgrade and change without damaging the collaborative fabric or leaving anyone technologically isolated who was previously connected. The fundamental medium for telephony is speech-grade audio. As long as speech-grade audio connections are ensured, the underlying technology can continue to change without users' concern. The telephone can be used to discuss almost any topic of human interest. It is completely independent of applications and supports chaotic content while rigidly holding to audio delivery standards. Unlike model-based collaborative applications, making a telephone call never requires the caller to know the kind of equipment owned by the person on the other end. Each party is free to purchase equipment as simple or as exotic as she wants without impairing the ability to communicate.

E-mail is one of the most successful asynchronous collaborative technologies. To the extent that it is confined to textual messages, the likelihood of successful delivery is approaching the levels offered by the telephone. However, when e-mail strays from simple text, the collaborative fabric develops numerous holes. When e-mail moves away from being a fundamental textual data type into being an application-specific data type, the divergent effects of chaos set in. E-mail is instructive because of the semiautomatic techniques that have been built around e-mail messaging. E-mail, like the telephone, imposes few restrictions on content or on the

tools used by participants.

The Web defines a collaborative medium based on hypertext markup language (HTML) and GIF and JPEG graphics formats, each of which is designed for communicating content virtually among people. Semiautomatic processes for generating content on the server side and for filtering and searching for content on the client side have extended the power of the Web beyond what humans can do. The hypertext transfer protocol (HTTP) is an inefficient communication mechanism between programs, but it is repeatedly used because its open and human-accessible nature fosters the discovery of new uses for the information and their rapid exploitation. Exploiting data in its human-centric forms has opened the door to a whole range of new automated information processes and techniques. The lessons of UNIX appear again.

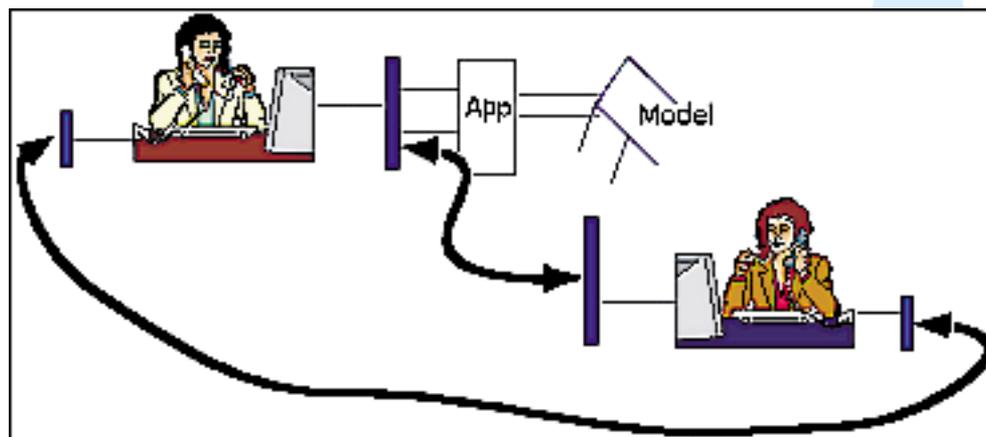


Figure 5. Surface-based collaboration.

Surface-Based Collaboration

Insights from successful collaboration in a chaotic world of information lead to an alternative to model-based collaboration called surface-based collaboration. The collaboration is based on surface, human-centric information rather than on deep, computer-oriented model information. In Figure 5 we see collaboration in terms of audio (over the phone) and pictures (replicating the displays).

For "what-you-see-is-what-I-see" (WYSIWIS) collaboration, products such as Net-Meeting, CuSeeMe, and Timbuktu are surface-based. They support collaboration of

screen or camera images and are completely application-independent. Their only failings are somewhat clunky early interface designs and Internet-induced latency and bandwidth problems. The interface design problems will work out with time, market success, and competition. The bandwidth and latency problems will succumb to Moore's Law and its network corollaries. Model-based synchronous collaboration will have competitive advantage only as long as Internet latencies are high and bandwidth is scarce. The synchronous collaborative solution is at hand. All we are waiting for is a sufficiently powerful Internet.

Asynchronous collaboration, however, is another matter; a good collaborative architecture does not yet exist. The outline of a surface-based collaboration architecture is shown

original application.

This architecture poses numerous technical challenges, including

- ★ Creating editors that can manipulate the surface information with less effort than raw pictures or audio,
- ★ Devising appropriate surface representations that encode some connection between the surface and the underlying application mode,
- ★ Developing techniques for representing multiuser changes to the surface so that the various collaborative versions can be understood and integrated, and
- ★ Developing semiautomatic techniques for incorporating surface changes back into the original application.

The Information Collaboration Interaction

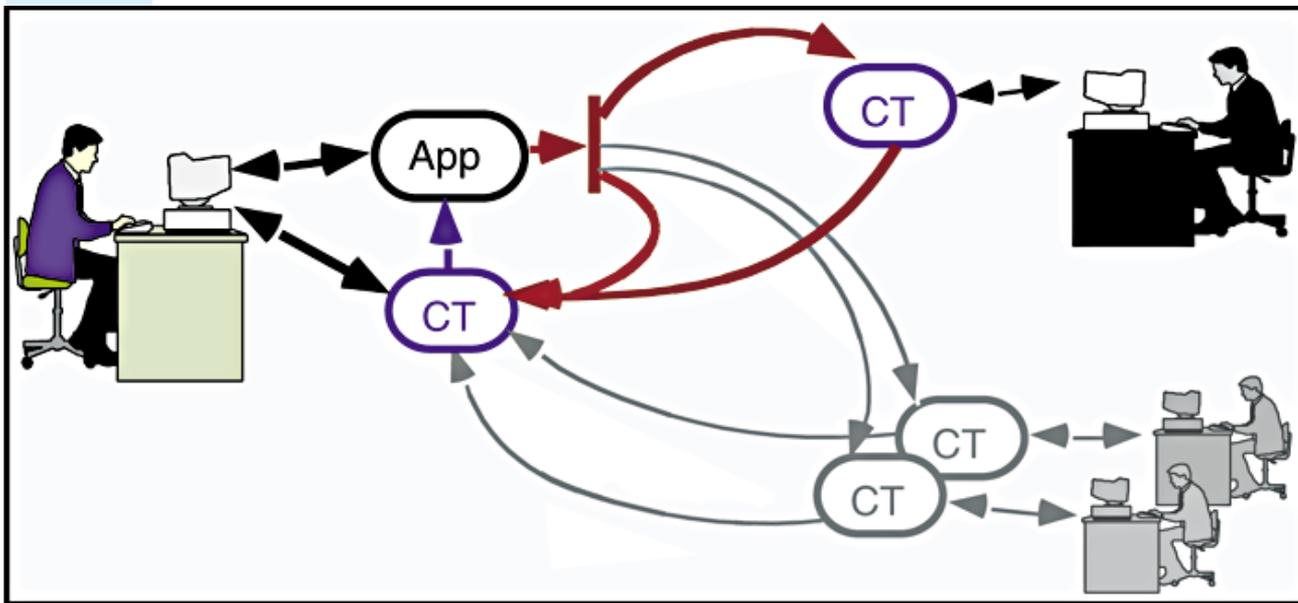


Figure 6. Surface-based asynchronous collaboration.

in Figure 6. In this architecture an application generates an information artifact and renders it into a surface representation. Adobe Acrobat's portable document format (PDF) is a possible candidate for such a surface. The surface representation is then transmitted to all participants. Each of the participants has his own collaborative tools (CT), which can browse, store, manipulate, and annotate the surface. The modified surfaces are returned to someone who must use her own collaborative tool to review the changes and comments of the participants and merge them into the

Environment (ICIE) project [6] at CMU has produced initial prototypes of this collaborative architecture, but there is still a great deal to be done. It is also possible to use model-based techniques when compatibility does exist among collaborators and use surface-based collaboration as a fallback position. Model-based solutions can provide more effective interaction. Experience with paper, however, shows that users will accept degraded interactivity to ensure the ability to collaborate with anyone, at any time, using any application. Surface-based approaches can provide such ubiquity.

Integrating Diverse Interactive Devices

The third contributor to interactive chaos is the potential diversity of forms that computers in the future can take. Many of these computing devices, such as those found in the modern automobile, serve a special purpose and perform their function with computer assistance but independently of other devices. For such independent devices, chaotic diversity is not a problem. The problem with diversity arises when such devices become part of the information space of an individual or community. Then the problems of integration, interoperation, and mastery of information and services become an issue.

The need to control the chaos arises on two fronts: information and interaction. People will use a variety of computing devices because they serve their information needs in a variety of situations. The information devices that will serve while walking down a hallway, sitting at a desk, or inspecting an automobile are different because of the physical situations in which they must be used. It is unacceptable, however, that each such device defines an isolated and independent information space. I recently watched my auto body repairman inspect the dents in my minivan. After inspecting the damage in the parking lot he then entered his shop where his computer was. While entering the damage information into the computer on his desk, he repeatedly jumped up and peeked out the window to look at the vehicle to make sure he had accounted for all of the damage. The fundamental problem was that the information space that produced his estimates and billings could not leave his desk and reach out to where the crumpled van was parked. The computing device that he needed while inspecting my van would not include a keyboard and a mouse but must fit within the same information space as his estimation and billing tools.

The development of new forms and shapes for computers will mandate new interactive styles. Consider a pager-sized device with one gigabyte of information inside and the whole Internet available through a radio. The size is dictated by what will comfortably fit in a pocket or purse.

This size, however, also forbids a screen of any reasonable size, no keyboard, and few buttons. The physical size will dictate a new style of interaction. Similar arguments can be made about room-sized computers. An interactive room allows multiple occupants to be positioned anywhere in the room. Again, a new interactive style is required. This problem is further complicated when several people with their own pocket devices enter an interactive room and want to work together while fully exploiting all of the available technology. Chaos emerges again.

Human limitations again bring convergence. The limitations of human learning will force a convergence of interactive styles to a relative few. Nobody will learn a new language for every speech-based appliance they buy. Nobody will learn a new style for every drawable surface they encounter. The physical form factors for interactive devices are limited by the physical characteristics of human beings. People are incapable of interacting directly with any object that is more than about 10 feet away. Upper and lower bounds exist on the sizes of objects that fit in the hand, pocket, or ear. Upper bounds also exist on what can be carried. Books and paper do not come in an infinite variety of sizes and shapes. They come in quite standard forms and shapes and have been shown over time to be effective for human use. The same will be true of computers. The factor limitations on physical form will dictate particular styles of interaction. For example, it is difficult to write anything useful on a pager-sized object, and a mouse will not be effective in an interactive room. However, the range of such styles will be bounded.

People convey information directly by body movement, gaze, and voice and indirectly through writing or drawing. The range of possibilities is quite limited relative to the potential effectors available to a computer. Although this range of possibilities is much larger than that offered by a screen, keyboard, and mouse, these limitations will still constrain the interactive chaos.

Interconnecting the information spaces of new devices can be partly addressed using collaboration and information techniques. The

exception is that the variability of devices will mandate robust transformation and cross-interaction among the human-centric data types and interactive styles. Interactive situations in which the eyes must do something else (like drive) will dictate a speech-based access to information originally encoded in pictures or text. Variability of devices and situations will force us to cross representational boundaries. Collaboration in shared spaces using personal devices will produce interoperability problems. We will need new algorithms and interactive techniques that support such multirepresentational interaction. The Pebbles [5] and Hybrid Paper/Electronic Interfaces [1] projects point to the beginnings of such solutions. The good news is that the set of basic representation and interactions is fundamentally limited.

Summarizing Chaos

In summary, exponential growth in just about everything related to computing will produce a level of information and communication chaos that is beyond the vision of the Xerox Star or its commercial descendants. The interactive environment and software strategies embedded in today's windowing systems and their applications cannot scale to handle the growing diversity. The application software market is fundamentally divergent and will only add to this chaos.

The only convergent forces are the fundamental limitations of human beings. By focusing on the five human-centric data types and the basic expressive behaviors of people, we can develop an essentially convergent set of technologies. We need to develop a level of usable power tools for audio, pictures, video, and 3-D environments that have previously graced the textual world. When we can search, extract, parse, store, index, manipulate, and interconnect these other human-centric data types with the same ease as text we will have

taken a major stride toward mastering information and collaboration chaos. When we have developed coherent strategies for other interactive modalities as we have for the screen, keyboard, and mouse, we will have mastered the chaos of multiformed computing devices.

References

1. Arai, T., Aust, D., and Hudson, S. PaperLink: A technique for hyperlinking from real paper to electronic content. In *Proceedings of Human Factors in Computing Systems (CHI 97)*, March 1997, pp. 327–334.
2. Giles, H., Coupland, J., and Coupland, N. *Contexts of Accommodation*. Cambridge University Press, England, 1991.
3. Kraut, R., Scherlis, W., Mukhopadhyay, T., Manning, J., and Kiesler, S. HomeNet: A field trial of residential Internet services. In *Proceedings of Human Factors in Computing Systems (CHI 96)*, April 1996, pp. 284–291.
4. Moran, T., Chiu, P., and van Melle, W. Pen-based interaction techniques for organizing material on an electronic whiteboard. In *Proceedings of ACM Symposium on User Interface Software and Technology*, October 1997, pp. 45–54.
5. Myers, B., Stiel, H., and Gargiulo, R. Collaboration using multiple PDAs connected to a PC. In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW 98)*, ACM, November 1998.
6. Olsen, D., Hudson, S., Phelps, M., Heiner, J., and Veratti, T. Asynchronous surface collaboration. In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW 98)*, ACM, November 1998.
7. Phelps, T. and Wilensky, R. Toward active, extensible, networked documents: multivalent architecture and applications. In *Proceedings of the ACM Conference on Digital Libraries*, 1996.
8. Sellen, A. and Harper, R. Paper as an analytic resource for the design of new technologies. In *Proceedings of Human Factors in Computing Systems (CHI 97)*, March 1997, pp. 319–326.
9. Watch What I Do: *Programming by Demonstration* (Cypher, A. ed.), MIT Press, 1993. ☞

PERMISSION TO MAKE DIGITAL OR
HARD COPIES OF ALL OR PART OF THIS
WORK FOR PERSONAL OR CLASSROOM
USE IS GRANTED WITHOUT FEE
PROVIDED THAT COPIES ARE NOT
MADE OR DISTRIBUTED FOR PROFIT OR
COMMERCIAL ADVANTAGE AND THAT
COPIES BEAR THIS NOTICE AND THE
FULL CITATION ON THE FIRST PAGE.
TO COPY OTHERWISE, TO REPUBLISH,
TO POST ON SERVERS OR TO REDIS-
TRIBUTE TO LISTS, REQUIRES PRIOR
SPECIFIC PERMISSION AND/OR A FEE.
© ACM 1072-5220/99/0900 \$5.00