EXHIBIT H

Canard: A Framework for Community Messaging

Pascal R. Chesnais
MIT Media Laboratory
Cambridge, Massachusetts, USA
E-mail: lacsap@media.mit.edu *

Abstract

A portable and extensible framework for community messaging is described. It integrates off–the–shelf communications devices and protocols by converting them into a uniform message representation. Using personal databases, the relevancy of a message is evaluated and appropriate delivery channels selected. This framework is being tested with a group of forty students in a living group at MIT. They have been equipped with two–way pagers which can be carried at all times and work in concert with this communication framework. The framework's ability to autonomously manage communications, makes it a flexible model for the wearable community[1] where communications reliability is an issue.

1 Motivation

Canard is an on–going experiment at the MIT Media Laboratory examining the application of Computer–Supported Cooperative Work[2] (CSCW) and User Modeling [3] technologies in a social context. At the core of this project is a modular, extensible communications framework that creates a uniform message representation for off–the–shelf and custom built (e.g., wearable computers) technologies (see figure 2). It is intended to function a communications bridge between people, even when they are unable to handle the communications themselves. Inter–personal and inter–application communications can be quickly developed using this framework. It can exist as a suite of isolated applications, or work in concert with other applications and people where automated cooperation is permitted. Its openness allows a "constructionalist"[4] approach to developing communications solutions. This model is well suited for people with wearable computers who wish to simplify communications with other people through the use of many communications channels.

*This project is part of the ongoing News in the Future research consortium at the Media Laboratory directed by Walter Bender and is made possible with an equipment grant from Motorola.

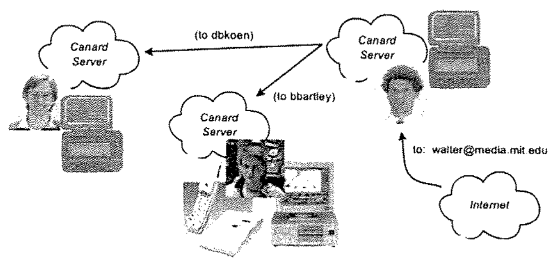

Figure 1: Canard allows for simplified, media rich communication between people. The user need only know the person with whom he is communicating, and not the method of transport. Canard negotiates the most economic channel for the message — which for the wearable computer user determines the use of a synchronous, expensive, wireless channel, or the use of a deferred land–line channel.

An on–campus living group of forty students is participating in this experimental framework. This group has intersecting demographics and needs that extend beyond the physical boundaries of their residence. Although they have a number of digital communication streams available (e.g., telephone, electronic mail (e–mail), etc.), it is not an integrated system that encourages group collaboration, nor do all the data streams generally travel with them (e.g., they are not wearable). Monitoring these digital communications, on behalf of the user and group Canard offers the ability to make inferences about member activities and whereabouts. For example, if a group member originates a message from a two–way pager, and no other recent observations were made, the system might infer that the person is away from on–campus communications. In the long run, wearable and environmental devices capable of making direct observations about their users will add strength to the user models we are building.

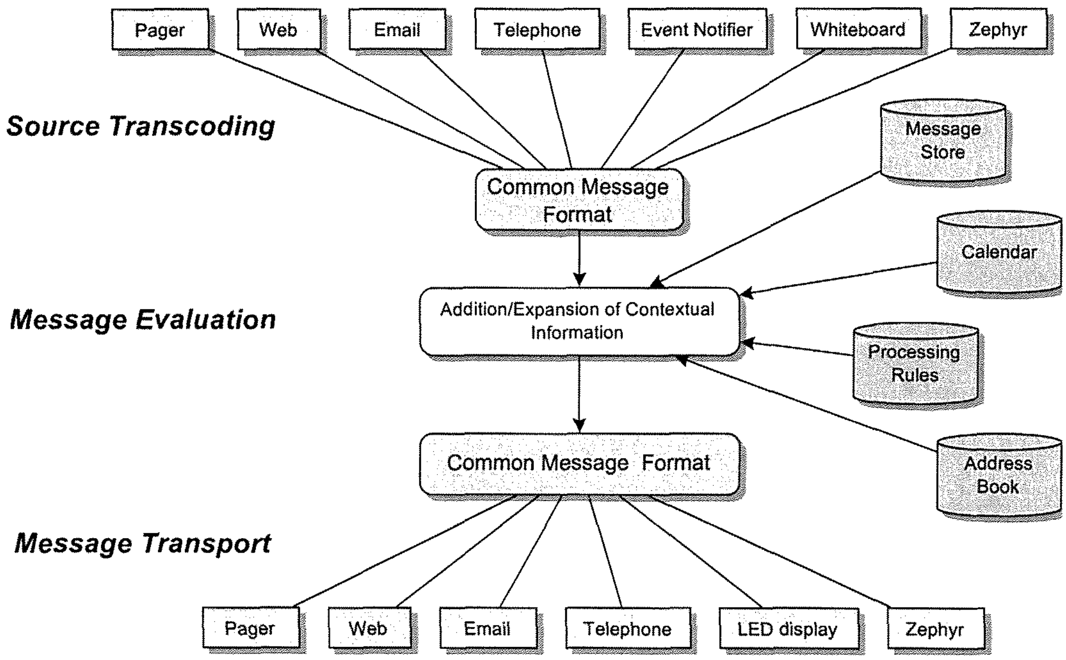

Figure 2: Canard messaging model is a three layer approach: representation, evaluation, and transport. First, source material is analyzed and converted into a uniform representation. Next, at the evaluation layer, one or more programs can be used to analyze the message using personal databases to evaluate its importance, and ultimately, its delivery mechanism. Finally, at the transport layer, a message is transcoded for delivery on a particular channel — stripping the message of unusable material (i.e., stripping video data to a text only device and instead sending a textual description, if available, of the footage).

2 Design Issues

Mitchell's City of Bits [5] outlines the urban planning issues of a society in a digital world. Canard's focus is on the communications infrastructure of a group space that straddles both a physical and an electronic environment. New communications protocols and devices, both synchronous and asynchronous, need to be easily integrated. Messages need to be transcoded effectively from one communication medium to another.

2.1 Computer–supported Cooperative Work

The field of CSCW provides some insight to the structure needed for the Canard system. Gavers [6] describes how "shared work involves the fluid transitions among focussed collaboration, division of labour, serendipitous communication, and general awareness". The transitions from awareness, communications and collaboration increases in formality and planning required to cooperate on a task.

Group Awareness The Canard system makes observations, through the use of personal communication sensors, in order to present group activities and whereabouts. Donath [7] suggests a way to tailor the manifestation of group activities and whereabouts to capabilities of available communications devices.

Serendipitous Communications Rather than burden group members with explicitly choosing which communication channels to employ, Canard offers a simplified representation of correspondents, such that the individual needs only to indicate the destination and urgency of the message. The negotiation of a communication channel is a function of the sender/recipient relationship and message urgency.

Focussed Collaboration Canard provides a structure for group members to offer services in the form of programs that are executed through known protocols and that may reside on foreign machines.

2.2 Communication Space Considerations

In a group environment, there are shared communications spaces. Using them for meaningful personal communications, without sacrificing privacy is a challenge for the system. Take, for example, a bulletin board: private messages are traditionally not left in a public space, however an indication of how important a message is may be left (i.e., "call your mother") without fully disclosing its nature.

Group members use the Canard messaging system 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 other group members may be present, but not necessarily aware of the members communications activity (e.g., using a pager in the dormitory's lounge area).

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

The physical scale of the communications interface is an important design issue. Weiser describes three interface scales in his Scientific American article on Ubiquitous Computing [8] from which parallels can be drawn to communications devices used by Canard.

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

Pads a letter–sized display (i.e., computer monitor). This device has a 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 (i.e., electronic whiteboard plus projection). This type of device is intended for a rich information display and is not likely to be easily transportable. It can be shared with a large group.

2.3 Automatic Message Addressing

A major concern in the Canard project is the ability for an individual to be relieved of the burden of deciding which communications channel to use to deliver a message. The actual message transport is negotiated between computers, acting on behalf of the sender and recipient as a function of their relationship, the recipient's activities, and the perceived urgency of the message.

Device dependency Aliases exist in the Canard system to handle the limitation of the source communications device. For example, it is extremely difficult to compose a message with the six buttons on a two–way pager. Using single letter aliases speeds up message composition time.

Urgency and confidentiality In addressing a message, some level of urgency or confidentiality can be associated with the message at the time of composition. This could signal the use of a particular communication channel: In the case of two–way pagers, we might use punctuation in conjunction with the address to indicate level of urgency (i.e., "pascal!" to mark an urgent message for "pascal").

External directories Available external directory databases are used whenever possible. For example, the address lookup sequence for a message sent from a two–way pager is:

- Pager specific alias
- Address book nickname entry
- Address book formal name
- A person listed in the MIT on–line directory

The project's goal is to add as much descriptive information as possible about the message that is independent of the communication channels used. This allows for decisions about delivery methods to be adapted to the recipient's current environment.

3 Communications Framework

The Canard framework is inspired by Jef Poskanzer's Extended Portable Bitmap Toolkit[9] (PBM). In the PBM library, images in different file formats (i.e., GIF, TIFF, TARGA, etc.) are first converted to an intermediate representation. Successive image transformations are performed (e.g., gamma correction, rotation, scaling, etc.) on the image in its intermediate representation. Finally, the transformed image is transcoded to its final file format. This allows developers to write image transformation functions that only need to know about 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 transcoding functions which convert from a source representation to a common format. Once in the common format a message's importance is evaluated against

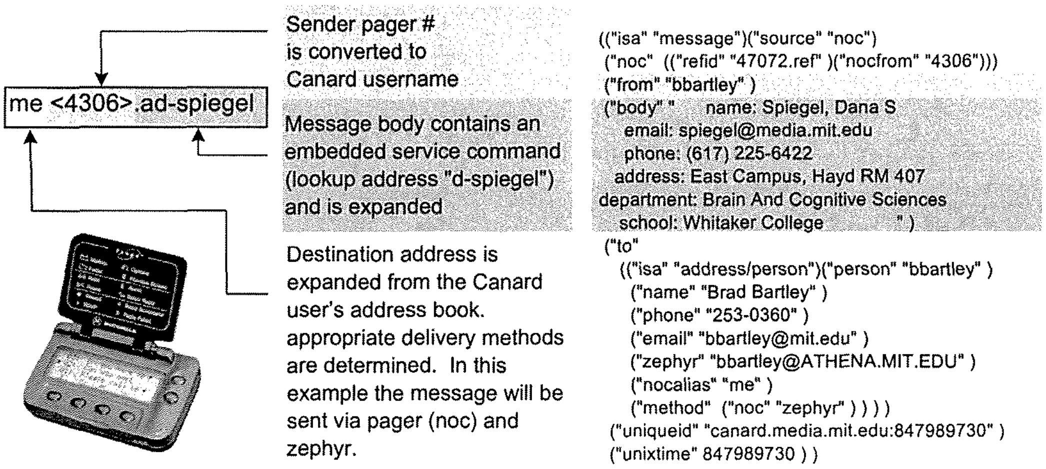

Figure 3: In this example, a message from a Motorola Tango two–way pager (on the left) is converted to a message object (on the right). Salient features from the original message are preserved. Delivery mechanisms are computed by consulting personal databases (in this case, the address–book).

personal databases. At delivery time, the message is converted into the format needed for delivery through a specified channel.

Figure 3 is an example message to illustrate how messages are handled by the Canard system.

3.1 Source Transcoding Layer

Messages entering the system come in different formats. The goal is to extract as many of the salient features of the message, while preserving the richness of the original format. In the case of Internet mail format, the sender information, message subject and body is extracted while storing the original headers for later use. Similar features can be extracted from other streams (i.e., caller identification (caller–ID) from incoming phone calls.)

Figure 3 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 message object. The first feature of the message is "me", which is the destination of the message. The second feature is "4306", which is the pager identification number. The third feature, ".ad-spiegel", is the message body composed by the user. The pager number is used to look up the owner of the pager, which will determine the personal database to use. The destination is looked up in the pager owner's address–book. In this example, "me" is a pager alias for the owner. Prior to message evaluation, the common message object for this message looks like:

```
(("isa" "message")
 ("source" "noc")
 ("noc" (("refid" "47072.ref")
 ("nocfrom" "4306")))
 ("from" "bbartley")
 ("to" "bbartley")
 ("body" ".ad-spiegel")
)
```

3.2 Message Evaluation Layer

Once a message is in a common form, it can be processed by one or more filters. These filters are much like the PROCMAIL[10] electronic mail filtering package, except, rather than solely operating on electronic mail messages, they evaluate anything in any field in the message object, and can leverage off of personal databases. This evaluation processing determines which delivery mechanisms to use.

In the example in figure 3, the message body starts with a '.', indicating an embedded service request. Embedded services are structured messages [11] 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). This embedded service model is particularly useful for consulting databases that exist outside of the system. In addition to the address lookup, direction assistance is offered, which will give directions from one street address to another. Certain assumptions can be made at this point: The starting address may be omitted and the starting point can be inferred as the recipient's current location.

3.3 Message Transport Layer

Determined at this stage are the message elements to deliver and their form. For example, a phone message has a number of elements — caller–ID and directory information, a sound file, and a description of the sound file (length and time recorded). To deliver this message to a text pager, the sound file is omitted, and instead the message: "Dana left a 36 second voice message" is delivered.

4 Implementation

4.1 Dtypes

Dtypes [12] provide an extremely flexible, platform–independent representation for complex data structures: integers, floating point numbers, strings and lists. Dtypes have an external ASCII representation which allows them to be used inter–operably between 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 [13] knowledge representation system and have been used to build a number of distributed servers as well[14] [15]. Dtype implementations exist in C, C++, Scheme, LISP, Java, and Perl on a wide variety of platforms.

Dtypes were used extensively for the fishWrap personalized news system [16] [17]. A modular system, with Dtypes as the interface, was used to link independently developed user modeling, knowledge representation and database servers.

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.

In the case of fishWrap, a parser was developed that allowed a text file to be applied against a contextual Dtype. This facility allowed presentation of data to be encoded outside the core software. For example, you could extract the headline from an article object and render it in a different style based on who is reading it.

The fishWrap parser can validate Canard database objects with text from documentation text file. 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 4). This allows the system to adapt 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 examples, there are a number of special object environments defined in the Canard framework:

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

uniqueid a unique string identifier for the object.

4.2 Canard Objects

A number of fundamental objects within the communications framework are implemented using Dtypes.

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

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 will determine how the message is processed and delivered.

Address An address object has the information needed to deliver a message. It may be as simple as a string which represents a server (to consult on how to deliver the message), or a complex

```
...
<TR BGCOLOR="#DDDDDD">
<TD>
<B>Duration: </B><!--GLUE: (print duration)--> minutes<BR>
<B>Lead time: </B><!--GLUE: (print leadtime)--> minutes
</TD>

...
```

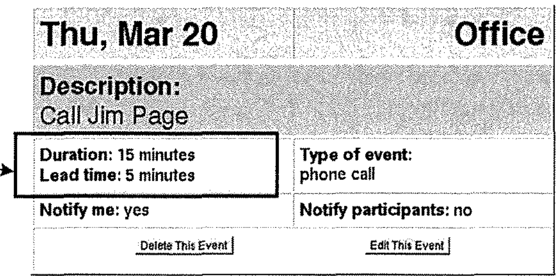

Figure 4: On the left is an excerpt from the style sheet used to render calendar event objects. On the right side is the resulting WWW hypertext presentation. Dtype evaluator commands are embedded within "<!–GLUE: –>" mark–up tags. This allows for WWW browsers to ignore the embedded commands.

Dtype with all the information needed by a transport mechanism to deliver the message. Address objects currently represent 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 [18] allows us to communicate when a particular activity is occurring. It may result in the automatic creation of a message object at a given time.

4.3 Communications Devices

At the time of this writing, a number of devices have been integrated into this framework and the implementation of others is being worked on. All of these interfaces are off–the–shelf technologies, albeit some more expensive than others.

Motorola Tango two–way pagers have been provided to the experimental subjects. These pagers have the capability of originating free form messages. The pager's discrete form factor makes it extremely portable and likely to accompany the subjects almost everywhere. 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. Mainly, if the pager is being used to originate messages, the person is away from desktop communications access. Pagers send an acknowledgment back to the system, allowing us to know if the message has been successfully delivered. Future pagers will have the ability to do simple processing of messages on the pagers themselves, allowing us to know whether or not the message was actually read. As the local processing power of pagers increases, they will assume some of the roles of wearable computers.

Telephones are probably the most ubiquitous communication devices available. A touch–tone based interface with a synthesized speech unit similar to Schmandt's PhoneShell [19] is available to the subjects.. Through this interface, the member uses the touch tone keypad to compose a message, as well as a speech synthesizer to read back text messages. In addition, MIT's telephone system provides us with caller–ID. This is important since the system can infer the caller's location if they are calling from a phone in a fixed location (as opposed to a cellular phone). Building a database with telephone locations is a simple and powerful addition to message analysis.

The World Wide Web provides participants a consistent interface across the many computer platforms available to them. Given this rich environment, a WWW interface for the subjects to use to manage their communications was specifically crafted. This interface allows them to read and compose messages, as well as consult and edit their calendars and address books.

From the very beginning, the desire was to integrate the Canard system into the public space of the residence. An electronic whiteboard, capable of digitizing the markings on its surface, is being investigated as a replacement to traditional bulletin boards. Coupled with a projection system, this interface makes an ideal connection to the group. Material captured on the whiteboard can be sent to the group's WWW page as an image file, or sent to handwriting recognition software to be transcribed to text form. The projection system can place configurable templates on the surface for formatted entry. In the long term future, a digital camera on the whiteboard will be able to take a snapshot of the person when they press the "publish" button, and using face recognition software [20] [21], attribute whiteboard data to the author. In addition, low cost scrolling LED displays can be placed in other public locations around the residence. 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 on the LED display his name followed by e–mail icon in red. This would tell him that an important message is waiting.

4.4 Integrated Applications

Dtypes are used to represent myriad objects in the Canard system. Common routines can be created for manipulation. Viewing objects becomes a simple matter of specifying a desired database and matching it with an appropriate style sheet for presentation (see fig. 4). A number of Dtype manipulation programs have been implemented for the subjects use.

This framework allows for the integration of messaging systems. MIT's Zephyr [22] system has been incorporated. Zephyr is a distributed window–gram messaging system commonly used at MIT for messaging other users on the Athena computer system without knowing at which machine the recipient is located. Zephyr can be used to find individuals on the system as well. Electronic mail is another rich source of information when coupled with structure multimedia message formats like the Multipurpose Internet Mail Extensions (MIME) [23]. Today, merely the text portion of e–mail is moved into the Canard system, but multimedia messages will be integrated in the future. These message bodies may be represented using Dtypes as follows:

```
((isa "message/mime")
 (mime-version "1.0")
 (mime-type multi alternative)
 (body(multi/alternative (
        (text/ascii "Hello World!")
        (image/gif  (4947 3846 6139...)))
...
```

A unified address book database is critical to the usability of the system. It provides a mechanism for the simple addressing of messages, which is independent of the communications channels. It provides pointers to other address servers that can authoritatively disclose how to communicate with someone.

Calendars can be rich in context — who we will be with, what we will be doing, and where we will be. Calendar information comes from many streams: E–mail is routinely used to communicate meetings and WWW based calendars contain schedules of activities. Having a unified representation allows us to create software capable of negotiating appointments automatically.

4.5 External Services

Important is the ability for participants to author collaborative services. Rather than limit the participant's creativity by imposing a single algorithmic language to author services, an interface with which they specify an address book entry for others to use is provided. Using known protocols, such as the hypertext transfer protocol (http), the service is accessed as if it were just another message being delivered (in this case to the particular service). The difference is that a reply is immediately composed by the service, and, depending on the situation, is sent as either part of the body of a message to another destination (in the case of an embedded service), or as a complete message sent back to the person who made the request.

5 Security and Privacy Issues

This paper describes the representation and evaluation on a "trusted system." The processing can reside on a centralized system, as it does in the current implementation, or can be distributed at the fringe of a network.

The use of "secure" transport layers to deliver the messages is not affected by where the computation takes place. For the WWW interface, we use a Secure Socket Layer [24] http server which provides a secure interface that is widely available in current browsers. As new encryption technology emerges, it can be folded into the framework of this project. Public keys can be maintained within the address book database and used to assure secure channels with other correspondents. The plan is to leverage off of available authentication services wherever possible (i.e.,Kerberos[25]).

Messages that are received in encrypted form using public keys are not automatically decrypted. This does not mean that they are ignored, since many of the features for evaluating the message are present in clear text form. In fact, the use of encryption can be a feature to infer the urgency of the message.

6 Conclusion

The Canard framework allows for rapid development of collaborative communications projects for wearable computing community. Its open architecture allows components to be developed independently in the programming language familiar to the developer. Its autonomous management of communications makes it an ideal server environment for wearable computing, where reliability is an issue.

It is too early to determine how the residents are using the system (while it is being developed), but initial interviews with the subjects indicates that they are receptive to this approach. The experimental group has a diverse background: Some are more technically savvy than others. It is unclear whether this framework is simple enough to encourage all the residents to be authors of collaborative services.

Focus group studies and individual interviews with the participants will be held to evaluate the effectiveness of this approach. The residents will be questioned about the types of services that they authored using this framework, and the processing rules they used to filter communications.

Future work will focus on: transcoding of communications from one stream to another; the tracking of people in the community; and a security model for personal access control.

7 Acknowledgment

The author wishes to thank Walter Bender for his support of this project as part of the News in the Future research consortium at the Media Laboratory. The author also wishes to thank Jim Page and David Morgan of Motorola.

Dana Spiegel, Matthew Mucklo, David Lamacchia and Dan Gruhl contributed to the design and implementation of the system. Jonathan Sheena and Nathan Abramson developed Dtype library and utilities which this system is built upon.

Canard has a publicly accessible WWW page located at http://canard.media.mit.edu/

References

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

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

[3] Jon Orwant, "For want of a bit the user was lost: Cheap user modeling," *IBM Systems Journal*, vol. 35, no. 3&4, pp. 398–416, 1996.

[4] Mitch Resnick, Fred Martin, Randy Sargent, and Brian Silverman, "Programmable bricks: Toys to think with," *IBM Systems Journal*, vol. 35, no. 3&4, pp. 443–452, 1996.

[5] William J. Mitchell, *City of Bits: Space, Place, and the Infobahn*, MIT Press, Cambridge, MA, 1995.

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

[7] Judith S. Donath, *Inhabiting the virtual city: The design of social environments for electronic communities*, Ph.D. thesis, Massachusetts Institute of Technology, 1996.

[8] Mark Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

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

[10] Stephen R. van den Berg, *Procmail - Mail processing Package*, RWTH-Aachen, Germany, October 1994, Sources are available at ftp.informatik.rwth-aachen.de as pub/packages/procmail/procmail.tar.gz.

[11] 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 [2], chapter 12, pp. 311–331.

[12] Nathan S. Abramson, "The dtype library or, how to write a server in less time than it takes to read this manual," Tech. Rep., MIT Media Laboratory Electronic Publishing Group, Camridge, MA, 1992.

[13] Kenneth Haase, "Framerd: Representing knowledge in the large," *IBM Systems Journal*, vol. 35, no. 3&4, pp. 381–397, 1996.

[14] Alan Blount, "Bettyserver: More news than you can beat with a stick," Tech. Rep., MIT Media Laboratory Electronic Publishing Group, Camridge, MA, 1991.

[15] Andrew Lippman and Roger Kermode, "Media banks: Entertainment and the internet," *IBM Systems Journal*, vol. 35, no. 3&4, pp. 272–291, 1996.

[16] Pascal Chesnais, Matthew Mucklo, and Jonathan Sheena, "The fishwrap personalized news system," in *Proceedings of the 1995 2nd International Workshop on Community Networking*, Princeton, NJ, June 1995, pp. 275–282.

[17] Walter Bender, Pascal Chesnais, Sara Elo, Alan Shaw, and Michelle Shaw, "Enriching communities: Harbingers of news in the future," *IBM Systems Journal*, vol. 35, no. 3&4, pp. 369–380, 1996.

[18] Irene Greif and Sunil Sarin, "Data sharing in group work," In Greif [2], chapter 17, pp. 477–508.

[19] Christopher M. Schmandt, "Phoneshell: the telephone as computer terminal," in *Proceedings ACM Multimedia 93*, New York, 1993, ACM, pp. 373–382.

[20] Matthew Turk and Alex Pentland, "Eigenfaces for recognition," in *Journal of Cognitive Neuroscience*, 1991, vol. 3, pp. 71–86.

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

[22] 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.

[23] Ned Freed and Nathaniel S. Borenstein, "Multipurpose internet mail extensions (mime) part one: Format of internet message bodies," *Network Working Group Request for Comments*, , no. 2045, November 1996.

[24] Alan O. Freier, Philip Karlton, and Paul C. Kocher, "The ssl protocol version 3.0," Tech. Rep., Netscape Communications Corporation, March 1996.

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

Pascal Chesnais is a Ph.D. candidate at the MIT Media Laboratory. He is a MIT/Motorola Fellow. Mr. Chesnais received the B.S. degree from Hofstra University in 1985. He spent two years at the Centre Mondial Informatique et Ressource Humaine in Paris, France. Mr. Chesnais received the M.S. degree in visual studies from MIT in 1988. A member of the Laboratory's Electronic Publishing group, Mr. Chesnais is the publisher of FishWrap.