Towards a Social Fabric for Pervasive Assistive Environments

Clare Owens[1]
University of Southampton
Building 32, Highfield
Southampton, SO17 1JB, UK
+44 (0)2380 597 208

clare@ecs.soton.ac.uk

David E. Millard
University of Southampton
Building 32, Highfield
Southampton, SO17 1JB, UK
+44 (0)2380 595 567

dem@ecs.soton.ac.uk

Andrew Stanford-Clark
IBM
Hursley Park Rd, Hursley
Winchester, SO21 2JN, UK
+44 (0)1962 815 225

andysc@uk.ibm.com

[1] Née Hooper.

ABSTRACT

The digital divide refers to a lack of technological access, part of which involves exclusion from a blooming arena of social interaction. People without mobile phones or PCs cannot access email, SMS or social networking websites; this includes many groups, such as the elderly, who can become vulnerable without good social contact. By enabling multimodal access to a variety of communication channels, including ubiquitous ones such as televisions and home telephones, this set of people can be included in such interactions. This paper describes a prototype pervasive messaging infrastructure for multimodal communications, and how it can be used as an assistive environment. Our eventual aim is to create a social fabric, a pervasive infrastructure layer to support more complex social experiences in the future.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia – *architectures, user issues.*

General Terms

Algorithms, Design, Human Factors.

Keywords

Messaging, multimodal communication, ambient social experiences.

1. INTRODUCTION AND MOTIVATION

A great variety of communication technologies are in day-to-day use: these include more traditional technologies such as email and landline phones, fully established tools such as mobile phones and instant messaging (IM), and newer items including social networking sites and blogs. Keeping track of data and communications through all of this can be problematic; additionally, a large chunk of society is excluded from this arena

of social interaction. For example, in general elderly people face a range of obstacles to the uptake of technology [16].

This issue is increasingly important, especially given the trend for ageing populations across the western world, and the geographical dispersion faced by many families. Technologies built without due consideration can exclude users considered 'non-standard', whether due to impairments, economic considerations or cultural aspects. Meeting these considerations when building any system requires a holistic outlook, and an inclusive approach to design [17].

Our goal is to connect these offline people through technology with which they are familiar. Opening up fresh communication channels for these people could help improve their general wellbeing. For example, contact with family might become easier (grandchildren could email grandparents regardless of whether the grandparents own a PC), and access to online content could be gained without requiring ownership of expensive computers or mobile phones, and knowledge of how to use web browsers. Another use of the system might be to view a weekly printed bulletin of updates about friends and family, and a daily bulletin with important social updates and prompts about the day ahead (appointments or jobs to do). In summary, many more people would be able to access the online content and communications facilities which so many of us take for granted.

To attain this goal, it is vital to decouple information from its original modality. For example, the content of a chatty email is the text, which as well as being displayed on a computer monitor could be printed out, displayed on a television screen or vocalised on a landline phone with text-to-speech technologies.

This decoupling of content from modality allows people much greater flexibility in terms of what information can be received when. It allows a user, Alice, to stream her voicemail to her PC if she has forgotten to bring her mobile phone to work, and means that she can email her grandfather Derek, even though he doesn't own a PC. Derek can access direct messages such as emails and SMSes, and also more ambient social information (e.g. microblogs or shared photographs) from his family members. Figures 1 and 2 show how a multimodal infrastructure can enable this decoupling.
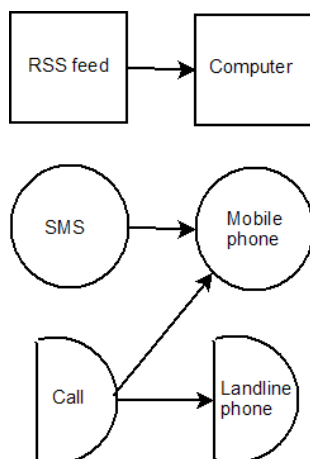
Figure 1. The current situation, where message modality constrains the devices upon which the message can be received.
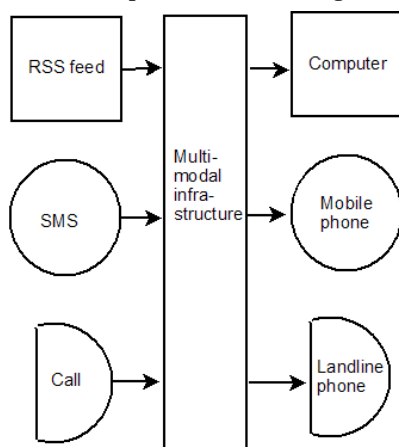
Figure 2. The vision, where message content is decoupled from its modality: content can be sent to any device.

To provide this functionality in an appropriate way, incoming information must be carefully managed. Any infrastructure able to choose the most appropriate modality must be able to account for a person's context – for example, their location, current activity and priorities (which may concern mode of communication and intrusiveness).

Several components are needed to realise this vision. Firstly, an underlying pervasive messaging model allows transport of information between locations, and translation between modalities: this system must understand channels such as email, IM and audio data. Secondly, a formal user information model (user proxy) captures a user such as Alice, her relationships with others, preferences for modality, and rules (such as not being phoned between 11pm and 7am). This paper introduces these concepts, with an eye to using this multimodal infrastructure as the basis for a social fabric. Section 2 provides a brief scenario, and Section 3 describes existing work. Section 4 outlines the envisioned system, and Section 5 discusses the prototype system in some depth. Finally, we discuss future work and evaluate our contribution.

2. SCENARIO OF USE

The following scenario reflects our vision:

Gerald is an elderly man, who lives alone in a flat. He has no interest in computers, but is happy to benefit from services enabled by technology. For example, he is very interested in seeing photographs from his grandchildren's first sports day at school. A digital photo frame is installed on his wall; when Gerald's son Matt uploads photographs from the sports day to a social networking website, the pictures are displayed in turn on the display.

Later that day, Gerald's grandson Billy writes him an email about the sports day. Gerald doesn't access his email account on a PC. Instead, the email is translated into an audio file and read to him via his landline telephone. Gerald listens to Billy's message quite late in the day, when Billy is probably in bed asleep. Gerald records a reply, a voice message for Billy to receive in the morning.

Gerald is also particularly interested in the building of a memorial in the north of France. He plans to travel to the site once the memorial is complete, and meanwhile his son has found a blog about the work in progress: Gerald receives updates from this blog, and also news stories from his local newspaper. He accesses both of these via a teletext[2]-style display on his television.

3. RELATED WORK

Pervasive computing involves the availability of many effectively invisible computers throughout the physical environment [23]: the technologies are so transparent to use that we do not notice them. Brown [4] describes 'calm' computing, where technology is not the focus of attention and people control technology, rather than being driven by it: our vision fits this description by allowing people to communicate and access information without concern over what devices are used.

Various proposed and implemented systems have looked at pervasive communication infrastructures. For example, the Mobile People Architecture [12] embodies a vision where people, rather than their disparate devices, are the endpoints of communications. A 'personal proxy' tracks user locations, and accepts, converts and forwards communications as appropriate. The concept of Universal Communication Systems, which combine various modalities of communication [1], is not a new one. Examples include: email notifications about voicemail [11]; a 'console' for group conversations via SMS, email, IM and the web [8]; and a proposed system to route emails and phone calls dynamically according to user context [9]. Similarly, Nakanishi et al have prototyped a system to redirect calls and emails based on people's schedule, location and available devices [14].

The Iceberg architecture [22] aims to integrate cellular telephony networks and the internet. The Universal Inbox [18] uses this to redirect communication based on pre-defined user preferences. Active Messenger [13] routes email to pagers, phones and faxes, based on calendar and other contextual information. It allows users to define preferences according to their location and the

---

[2] Teletext is a text-based television information retrieval service, which runs in the UK.

time of day. Another implementation, Mercury [10], integrates phones, IM, email and pagers.

Despite this work, progress in the real world has been slow, perhaps due to the challenges of implementing this technology in the wild. Turk [21] notes the need to integrate channels and address privacy issues, whilst Branco [3] raises questions such as what data helps ascertain user context, and how best to map content for impaired users.

Our work differs from existing designs and implementations in several respects. Firstly, our vision isn't only about routing direct communication. We cover broader information, some relating to the wider world (RSS feeds and sensor data, perhaps monitoring electricity usage or car mileage) and some more personal in nature (reminders and ambient awareness of friends' activities and wellbeing, achieved with text and photographs through social networking mechanisms such as Facebook).

Thus, the system routes two types of data:

1. personal data, to which access requires verification (e.g. direct communications such as email and text messages and ambiguous communication such as social networking data)

2. publicly available regularly-updated material, such as Twitter streams (see: http://twitter.com/), blog posts, sensor data and other items on RSS feeds

The system will allow different levels of notification, from viewing new information only upon explicitly logging into the system to being woken in the night when an urgent message arrives. We are aiming at a very broad audience: users may own PCs or mobile phones, but they need not. Anyone with a device which can interact with the system would be able to use it, including owners of older technologies such as televisions or landline telephones. Additionally, the emphasis on the assistive nature of this technology naturally leads to an approach of inclusive design, involving stakeholders where possible [17].

Ethical implications are very important in this domain, particularly given the use of potentially invisible technology and sensitive information relating to personal location [20]. Another important aspect is intrusiveness, which is a noted issue [19]: providing the right information on the right device at the right time and with the right level of intrusiveness has been raised previously [5]. An important aspect of our ongoing work is a full investigation into these dimensions.

4. ENVISIONED SYSTEM
We envision a system where users may browse or search a list of publicly available items such as RSS feeds and Twitter streams, and also subscribe to private streams such as email accounts and text messages (SMSes). To subscribe to private items, users must provide verification, such as a username and password for email and IM accounts, and a text from the relevant mobile phone for SMSes. Non-PC users would enter this data in novel ways. A teletext-like interface on televisions could offer one method of system configuration, while voice recognition technologies would enable configuration via landline telephones.

The envisioned system will use whatever available technology there is to determine user location: this might include the current

cell of a mobile phone, the wifi network used by a PDA, or sensor data (e.g. a broadcast car location). Additionally, users may explicitly notify the system about their current location. At the moment, locations and subscriptions are initially set up in an XML file, and may be edited via the GUI.

Users may have one of three levels of linkage with any given location. They may choose to be 'offline', in which case messages are not routed to them, but queued until they return online; they may be 'guests', and logged into the location – but their information is discarded when they leave; finally, they may be durable guests, in which case their preferences are remembered for future visits.

A more advanced system will carry out some transformations so as to deliver the data in an appropriate format. For example, most emails won't fit into a 160-character SMS, but a text message detailing the sender, subject line and first n characters of content might be appropriate. The method and nature of transformations is an area requiring further attention, although existing work has made inroads in this area: for example, Nagao et al have discussed content adaptation based on available devices [15].

Current preference lists, which rank modes of communication, are linked to people's locations. Later, they may have additional constraints relating to time, people and events. For example, Alice may not want to receive phone calls between 11pm and 7am (time); unless they are from her partner, Bill (people); but if she is at a doctor's appointment (event), she may not want to receive any calls at all.

Users may in future also list 'trusted contacts', who can receive limited information from the envisioned system about the person's context. For example, it is useful for Bill to know that Alice has received a text notifying her of his 1000-word email, but Alice has to list him as a trusted contact in order for him to be told this.

5. PROTOTYPE SYSTEM
We have developed a working prototype based on IBM's Lotus ® Expeditor micro broker. This is coupled with a simulation environment, which demonstrates that the underlying logic and model are sound and enables us to explore different usage scenarios.

The system receives incoming information (which in future will be from websites, email inboxes, sensors etc.) and delivers it to an appropriate end point. End users shouldn't have to concern themselves with the type of a message: whether the content was sent as an SMS, email or Tweet need not affect when and how it is received. Currently people may choose a different modality for message sending according to message priority. For example, if Alice's friend Bob is in the cinema, she may send him a text message but not phone him. This system aims to eventually remove that load.

5.1 Using the Prototype
A screenshot of the current simulator can be seen in Figure 3, below. The simulator provides a listing of information from the world as modeled by the system: this list describes the time, devices, people, locations, data sources and events within the world. The right hand side contains controls for changing the state of various items within the world: it is possible to change a person's location and subscriptions, to send messages, and to
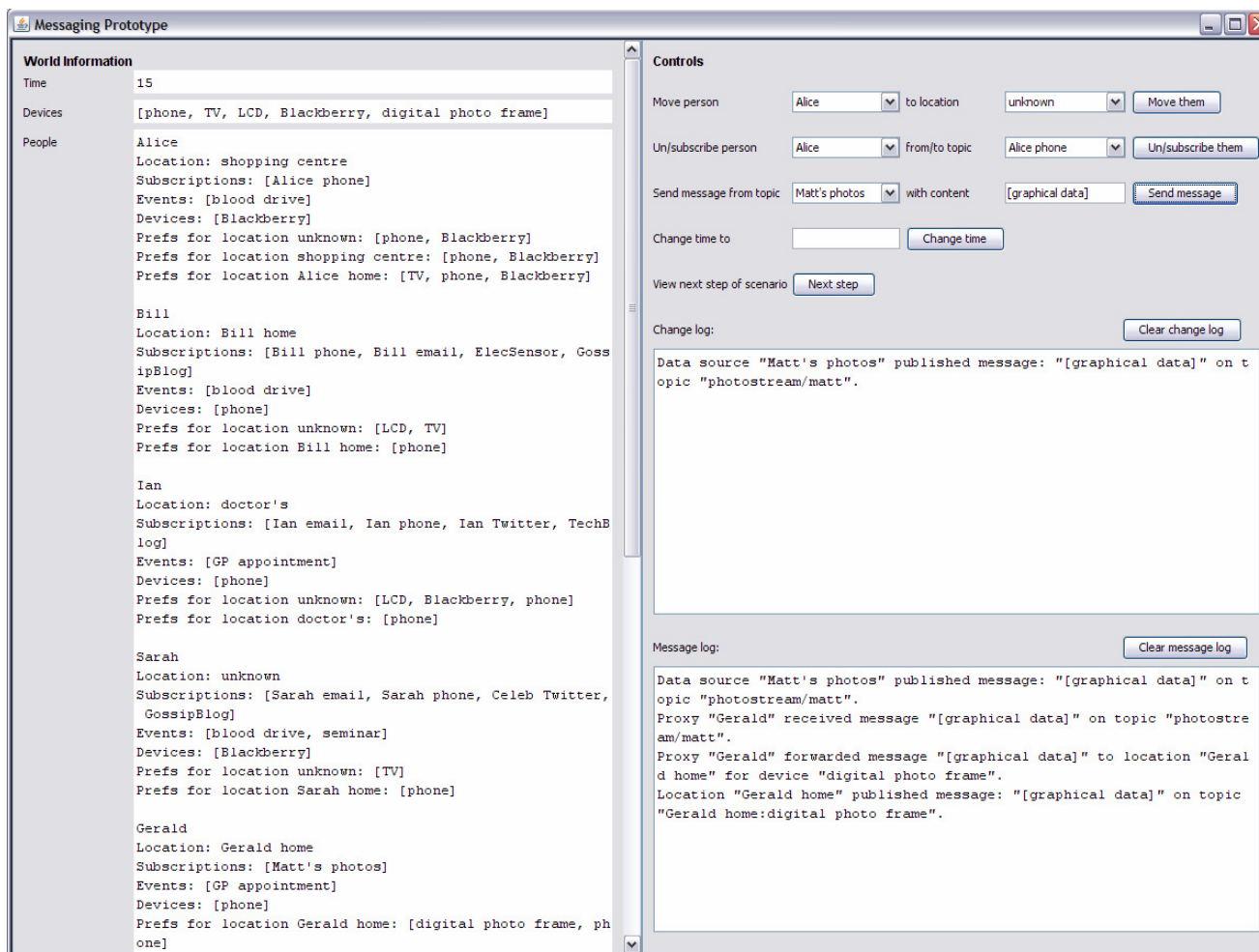
Messaging Prototype

World Information

Time	15
Devices	[phone, TV, LCD, Blackberry, digital photo frame]
People	Alice

```
Alice
Location: shopping centre
Subscriptions: [Alice phone]
Events: [blood drive]
Devices: [Blackberry]
Prefs for location unknown: [phone, Blackberry]
Prefs for location shopping centre: [phone, Blackberry]
Prefs for location Alice home: [TV, phone, Blackberry]

Bill
Location: Bill home
Subscriptions: [Bill phone, Bill email, ElecSensor, Goss
ipBlog]
Events: [blood drive]
Devices: [phone]
Prefs for location unknown: [LCD, TV]
Prefs for location Bill home: [phone]

Ian
Location: doctor's
Subscriptions: [Ian email, Ian phone, Ian Twitter, TechB
log]
Events: [GP appointment]
Devices: [phone]
Prefs for location unknown: [LCD, Blackberry, phone]
Prefs for location doctor's: [phone]

Sarah
Location: unknown
Subscriptions: [Sarah email, Sarah phone, Celeb Twitter,
 GossipBlog]
Events: [blood drive, seminar]
Devices: [Blackberry]
Prefs for location unknown: [TV]
Prefs for location Sarah home: [phone]

Gerald
Location: Gerald home
Subscriptions: [Matt's photos]
Events: [GP appointment]
Devices: [phone]
Prefs for location Gerald home: [digital photo frame, ph
one]
```

Controls

Move person [Alice ▾] to location [unknown ▾] [Move them]

Un/subscribe person [Alice ▾] from/to topic [Alice phone ▾] [Un/subscribe them]

Send message from topic [Matt's photos ▾] with content [graphical data] [Send message]

Change time to [   ] [Change time]

View next step of scenario [Next step]

Change log: [Clear change log]

```
Data source "Matt's photos" published message: "[graphical data]" on t
opic "photostream/matt".
```

Message log: [Clear message log]

```
Data source "Matt's photos" published message: "[graphical data]" on t
opic "photostream/matt".
Proxy "Gerald" received message "[graphical data]" on topic "photostre
am/matt".
Proxy "Gerald" forwarded message "[graphical data]" to location "Geral
d home" for device "digital photo frame".
Location "Gerald home" published message: "[graphical data]" on topic
"Gerald home:digital photo frame".
```

Figure 3. Screenshot of simulator.

change the time in the world. Below these controls are a change log (which records button presses) and a message log (which records the result of publishing messages).

These controls allow users to see how messages traverse the world. By changing the subscriptions and location of a person, one changes which messages they will receive, and where. For example, as shown in the screenshot, it is possible to walk through aspects of scenarios, such viewing what happens when Matt publishes photos on the stream to which Gerald is subscribed.

Note that aspects such as the current time and events can affect message receipt. When the system doesn't know where a person is, it checks to see whether they are currently attending an event: if so, it reasons they are at the event's location, and tries to send the message accordingly. If no suitable devices are available upon attempted message delivery, the proxy of a given person will queue the message for later delivery.

5.2 Messaging Paradigm and Technology

We use IBM's micro broker middleware for message transfer [7]. Middleware provides connectivity between networked applications and software, while micro broker is a publish and subscribe (pub-sub) message broker appropriate for a variety of applications, especially in mobile and pervasive domains. Messages travel between brokers, which determine which recipients receive which messages. Pub-sub is one of two approaches to describing destinations in pervasive messaging:

1.   Point-to-point messaging: publishers specify message recipients, and place messages on those recipients' queues. This routing does not take advantage of common paths, and becomes inefficient when there are many subscribers [2].

2. Pub-sub messaging allows delivery of one message to many subscribers. Subscribers may register interest in a 'topic' (message destination or queue), and then receive messages sent to this topic.

Given the inclusion of one-to-many data sources such as sensors, blogs and RSS feeds, the pub-sub paradigm is most suited to our model.

The prototype system is written in Java™ and uses IBM's micro broker middleware. It models a real-life implementation in which broker instances deal with subscriptions and publications. Brokers can handle many connections at once. For example, an instance of micro broker can handle around 2000 connections at any given time: for our purposes, this system is scalable.
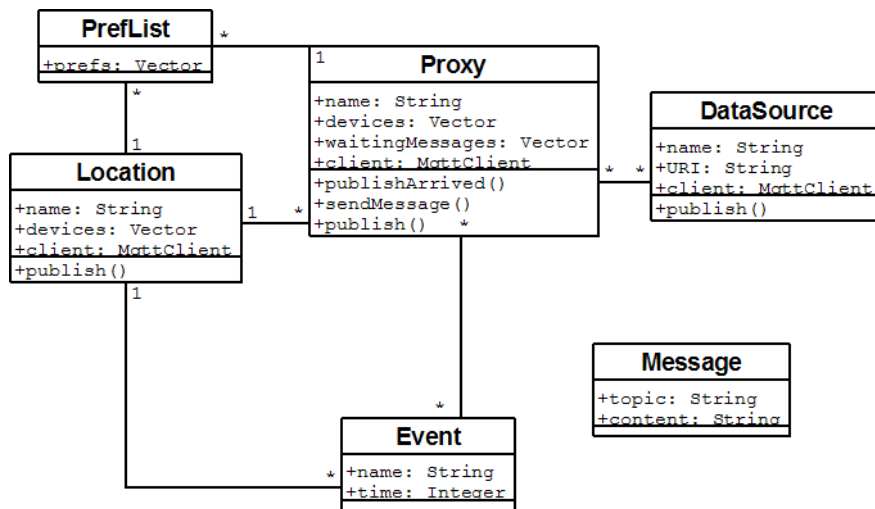
Figure 4. Class diagram.

The user proxy is an application subscribed to the individual's streams of information and relevant notification channels (e.g. regarding location and available devices).

5.3 System Model and Logic

Figure 4 shows a class diagram of the prototype system. As can be seen, the `Proxy` class is central: instances thereof represent an individual end user. This class has various properties, including preference listings (where each `PrefList` denotes preferred communication channels for a given location), a current location, any events the person plans to attend, and a list of `DataSources`, the items to which the person is subscribed. This class also contains a listing of devices on the person, and any queued messages for the person.

Users can have as many `PrefList` items as they want. It is recommended that every user has one `PrefList` with no specified location: this is used as a default, if they are in a location for which no `PrefList` has been defined, or their location is unknown. If someone doesn't wish to receive information via a certain delivery mechanism, they simply do not list relevant output devices.

A `Location` has a name and a list of devices which are present, while an `Event` is a combination of a location and time, with a name. Example events include clinic appointments, tea dates and film screenings. `DataSources` have a name and URI, and can publish messages to that URI. `Messages` have a topic (the URI to which the message was published) and content. Messages may be written and published via the GUI.

Note that `Locations`, `DataSources` and `Proxies` all have `MqttClients`. `DataSources` and `Locations` use these to publish messages. `Proxies` also publish with their `MqttClients` (to the topics of devices located upon the person they represent), and use a `publishArrived` method to receive incoming messages.

Proxies carry out the logic of running through a person's preferences for modality, and poll the person's current environment for available options, sending the information as appropriate. When a message is received, the proxy's

`sendMessage` method is called. In this method, the proxy works through several steps:

1. If the current location is unknown, check events for this person: if the person should currently be at an event, set their current location to that event's location.

2. Try to find a preference listing for the current location; if there isn't one, use the default preference listing (for the 'unknown' location). If there is no default preference listing, throw an error.

3. Iterate through the preference list: look up the most preferred device. Check if an instance of the device is available, either on the person or in their current location. If so, send the message to that device's URI; otherwise, check the next most preferred device. If a message cannot be sent (no preferred devices are available), add it to the Proxy's `waitingMessages` Vector.

As described, `Proxies`, `Locations` and `DataSources` run instances of `MqttClient` in order to publish messages and subscribe to topics. A broker can handle many connections at once; an instance of micro broker can handle around 2000 connections, which is scalable for our purposes.

5.4 User Preferences

Users can build lists to rank communication channels in order of preference. For example, Alice's list `[speakers, TV]` means that Alice prefers to hear information from loudspeakers; if these are unavailable then she wants to access it on a television.

Preference lists generally relate to specific locations, meaning that users may build multiple lists: it is likely that Alice wishes to receive notifications about a friend's activities in different ways depending on whether she is at work or at home. A 'default' preference list applies if Alice is in a location for which no specific list exists.

5.5 Sequence Diagram of the Scenario

Figure 3 showed the simulator demonstrating the beginning of the given scenario. The following sequence diagram shows the beginning of this transaction.
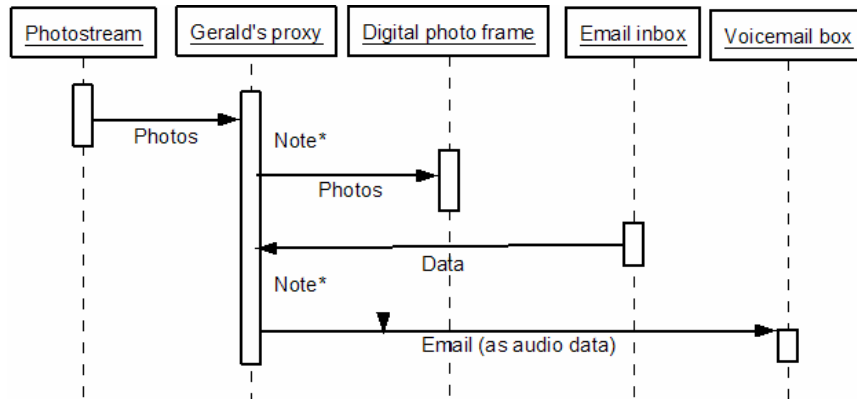
*Note: At each of these points, the proxy checks Gerald's preferences, and reasons that, based on Gerald's current location and preferences, it is appropriate to send the data on to the sources shown.

Figure 5. Sequence diagram showing Gerald's proxy transforming and routing incoming material based on his preferences.

6. FUTURE WORK

We are planning a more sophisticated system that will use the same framework as the prototype we have presented. The first stage will be to advance the messaging infrastructure to include more sophisticated preference data and logic. We then intend to build on this to create a pervasive infrastructure with a social fabric, allowing the easy integration of pervasive social tools.

6.1 Advancing the Messaging Infrastructure

The key extension to the infrastructure would be to improve the `Proxy.sendMessage` method, the reasoning engine of the system, so that it can handle more complex logic. To support this, classes that represent objects in the world also need to be augmented. For example, the Message class may be augmented with a priority flag, and the Location class may gain a list of adjacent locations. More complex classes will be used to represent Devices and Subscriptions, which are currently represented in the system as strings. However, no changes would impact the system framework.

Environments will themselves have preferences: for example, it may be appropriate to suppress the ring tone on mobile phones in cinemas, or meeting rooms in current use. In contrast, an open-plan office environment may allow soft beeps but nothing above a certain volume.

Events could also have preferences. Were incoming messages blocked, perhaps during a film screening, provision of an emergency phone number may be appropriate. For example, if Alice's mother is taken into hospital whilst Alice is at the cinema, a phone call or SMS will not reach her. However, it may be possible to have a cinema employee find Alice in person.

The interaction of preferences is another area for future work, particularly if environments and events also have preferences. If two people meet, how do their preferences affect one another? Alice may not want interruptions, but Bob may be happy to receive these; meanwhile, Carol may not want to receive messages from Debra if she is with Evan. It is likely the logic to deal with this would reside with the proxy of each person: for example, Carol's proxy may be aware that messages from Debra should not be propagated if Evan is present, and Carol may have marked

meetings with Evan so that her proxy knows when to delay Debra's messages.

The system might also consider computer ownership and usage (enabling reasoning about location based on whether a laptop is on or off); message sensitivity; intrusiveness; and connections between topics and events (allowing prioritization of notifications relevant to a current event).

Proxies need to handle issues of privacy and intrusiveness. People are unlikely to want personal messages to appear onscreen while giving a presentation, but it may be appropriate to display a message relevant to everyone in the meeting.

6.2 Creating a Social Fabric

With the prototype messaging system in place, we can consider what next steps are required to create and support the social layer.

The multimodal messaging infrastructure is effectively the underlying framework that will support the social fabric: it routes and transforms messages appropriately. The social fabric, by contrast, comprises the interface and social model supported via this multimodal framework. Clean separation of these two aspects allows appropriate focus on each part: the underlying messaging infrastructure can easily be used in other domains, while the social fabric can be developed as a discrete concern, with due consideration given to issues such as how it will be used, the interface and HCI aspects, and ethical considerations.

Experiences which superficially appear to be the same may manifest differently in different media. This is demonstrated by Dix's work on Christmas crackers [6]. A cracker is made up of an inner tube wrapped in brightly colored paper. When pulled by two people, it splits into two uneven parts, making a bang as it does so (caused by a small chemical mechanism called a cracker snap). Crackers generally contain a paper hat, a small plastic toy and a motto or joke.

Dix wanted to create virtual crackers, available on a website. Rather than trying to directly emulate real crackers, Dix succeeded in capturing aspects of the experience of pulling crackers, and translating those to the medium of the web. He did this by deconstructing the experience of pulling a cracker, and then reconstructing it in the new medium. By deconstruction, Dix

refers to "taking apart, teasing out the strands that make something what it is … and, in this context, especially those that make something 'work' as an experience or as a designed artefact."

There are two aspects to the deconstruction, which are consideration of surface elements and experienced effects. An example surface element of Christmas crackers is that they are traditionally 'cheap and cheerful': thus the webpage for virtual crackers was simple, with cheerful graphics. An aspect of the experience of pulling a Christmas cracker is the shared nature of the experience. To incorporate this, the virtual cracker system would not allow the sender to see the contents of the cracker until the recipient had 'pulled' it (by clicking on a link).

Dix's approach to Christmas crackers involved deconstructing an experience in the real world and reconstructing it in a digital context, the web. We intend to deconstruct a digital experience (using the web for communication and ambient social awareness of friends' activities via social networking sites) and reconstruct it in a different digital context, by providing that information and interaction via the modalities previously described.

A first step towards this process of deconstruction has been taken: the authors have surveyed the functionality offered by a range of social networking sites, and analyzed which functionality is key across this range. The next step is to examine the surface elements and experienced effects of this functionality, in order to abstract them to new media.

7. CONCLUSIONS

We have presented a design that will allow provision of fluid multimodal information to its users, based on context, priorities and preferences. This is motivated by the ideal of allowing people without use of digital technologies to access (and return) communication and social information which originated with these.

This system will be capable of delivering information in appropriate formats according to various factors such as personal preference, time of day and location. The system releases people from the burden of choosing a communication channel based on what they imagine is best for recipients, such as worrying over whether to send an SMS to someone who may be asleep, or whether to call someone who could be at the cinema.

As observed in Section 3, there exists prior work looking at multimodal communications, much of which is from at least several years ago. Recent developments in social communication, such as microblogging and instant photo sharing, have introduced new requirements to these systems, and we are still in the process of understanding user attitudes and behaviour on popular platforms such as Twitter, and social networking websites, such as Facebook or MySpace. Unlike email, SMS and IM, these technologies are not primarily about direct messages, but more ambient awareness, and so they must be treated differently in the context of multimodal communications.

In parallel with this work, the authors have been considering what motivates people to make use of emerging social networking websites, what it is these offer, and how that experience might be transferred to a pervasive environment. By using Dix's approach to deconstructing experience, it is possible to analyse the experience of using social networking sites and translate it to new

modalities. This access to social experiences forms the core of the social fabric which will be supported by the messaging infrastructure presented in this paper: combining the two layers results in a fully-fledged multimodal social fabric.

8. REFERENCES

[1] Andrews, C. 2001. Unified Communication Systems. Crossroads (Fall 2001)

[2] Banavar, G., Chandra, T., Strom, R. and Sturman, D. 1999. A case for message oriented middleware. In 13th International Symposium on Distributed Computing.

[3] Branco, P. 2001. Challenges for Multimodal Interfaces Towards Anyone Anywhere Accessibility: A Position Paper. In Workshop on Universal Accessibility of Ubiquitous Computing: Providing for the Elderly.

[4] Brown M., Weiser, R. and Gold, J.S. 1999. The origins of ubiquitous computing research at PARC in the late 1980s. IBM Systems Journal. 38, 693-696.

[5] de Roure, D.C., Hey, T. and Trefethen, A.E. 2005. Where the Grid meets the Physical World - Research Issues in Grid and Pervasive Computing, http://www.semanticgrid.org/documents/gridperv3.pdf

[6] Dix, A. 2003. Deconstructing Experience – pulling crackers apart. In Funology: From Usability to Enjoyment, M Blythe, K. Overbeeke, A. Monk and P. Wright, Eds. Kluwer, Dordrecht, the Netherlands, 165-178.

[7] Gale, M. 2007. End-to-end integration with pervasive messaging and IBM Lotus Expeditor micro broker, http://www.ibm.com/developerworks/lotus/library/expeditor-pervasive/

[8] Heyer, C., Brereton, M. and Viller, S. 2008. Cross-channel mobile social software: An empirical study. In CHI'08, pp. 1525-1534.

[9] Kamioka, E. and Yamada, S. 2004. Environment-adaptive personal communications realizing ubiquitous computing networks. Electronics and Communications in Japan. 1, 87, 34-47.

[10] Lei, H. and Ranganathan, A. 2004. Context-Aware Unified Communication. In IEEE International Conference on Mobile Data Management.

[11] Liscano, R., Impey, R., Yu, Q. and Abu-Hakima, S. 1997. Integrating Multimodal Messages Across Heterogeneous Networks. In ENM - First IEEE Enterprise Networking Mini-Conference, pp. 45-53.

[12] Maniatis, P., Roussopoulos, M., Swierk, E., Lai, K., Appenzeller, G., Zhao, X. and Baker, M. 1999. The Mobile People Architecture. Mobile Computing and Communications Review. 1, 2, 1-7.

[13] Marti, S. and Schmandt, C. 2001. Active Messenger: filtering and delivery in a heterogeneous network. Human Computer Interaction, 20, 1-2, 163-194.

[14] Nakanishi Y, Takahashi K, Tsuji T and Hakozaki K. 2002. iCAMS: A mobile communication tool using location and schedule information. International Conference on Pervasive Computing (Zurich, Switzerland, August 2002), 82-88.

[15] Nagao, K., Shirai, Y. and Squire, K.: Semantic annotation and transcoding: making web content more accessible. 2001. IEEE Multimedia. 8, 2, 69-81.

[16] Namazi, K.H. and McClintic, M. 2003. Computer use among elderly persons in long-term care facilities. Educational Gerontology. 29, 535-50.

[17] Newell, A.F. and Gregor, P. 2004. Design for older and disabled people - where do we go from here? Universal Access in the Information Society. 2, 1, 3-7.

[18] Raman, B., Katz, R.H. and Joeseph, A.D. 2000. Universal Inbox: Providing Extensible Personal Mobility and Service Mobility in an Integrated Communication Network. In Workshop on Mobile Computing Systems and Applications (WMSCA'00).

[19] Ramchurn, S.D., Deitch, B., Thompson, M.K., de Roure, D.C., Jennings, N.R. and Luck, M. 2004. Minimising intrusiveness in pervasive computing environments using multi-agent negotiation. In Proceedings of the 1st International Conference on Mobile and Ubiquitous Systems.

[20] Stone, A. 2003. The Dark Side of Pervasive Computing, IEEE Pervasive Computing, 2, 4-8.

[21] Turk, M. 2005. Multimodal Human Computer Interaction. In Real-time vision for human-computer interaction.

[22] Wang, H.J., Raman, B., Chuah, C. et al. 2000. ICEBERG: An Internet-core Network Architecture for Integrated Communications. IEEE Personal Communications (Special Issue on IP-based Mobile Telecommunication Networks).

[23] Weiser, M. 1989. The computer for the 21st century. ACM SIGMobile Mobile Computing and Communications Review. 3, 3-11.