Godard: A Flexible Architecture for Audio/Video Services in a Media Space

Paul Dourish
Rank Xerox EuroPARC
Cambridge, UK
dourish@europarc.xerox.com

"It wasn't very civil of you to offer it," said Alice, angrily. "It wasn't very civil of you to sit down without being invited," said the March Hare.

1 Abstract

Research into the use of audio-video technologies for collaboration frequently raises issues of privacy and control for the users of these systems. Common problems are those of protecting individual privacy and providing mechanisms for awareness of network usage, while at the same time maintaining enough flexibility in the network for it to be useful. Godard is a flexible architecture for audio-video services, developed at Rank Xerox EuroPARC, which tackles these problems. This particular model has been in use at EuroPARC for approaching a year, and provides the users of an inter-personal audio and video network with mechanisms for protection, feedback and control. Informal usage evidence suggests that is has addressed these issues with some success. This mechanism is contrasted with other approaches, and particular design issues are highlighted, particularly with respect to the role of a technological solution within a usage culture.

2 Introduction

The RAVE project at EuroPARC, in common with groups at other research centres, is experimenting with the notion of a "media space", the integration of computer and audio-video technologies for workgroup collaboration [Stults, 1986; Buxton and Moran, 1990; Gaver *et al*, 1991]. Our media space links individuals through an audio and video network in addition to the more traditional computer network, and affords them flexible workstation-based control of inter-office connections. By attempting to recreate some aspects of physical co-presence, the media space provides an environment for collaboration, even though the collaborators may be separated by large distances—in different offices, buildings or countries.

One important feature of the EuroPARC RAVE media space, unlike some others, is that it is *ubiquitous* within the lab. All lab members have a "node"[1] on their desk; at the time of writing, we have 23 personal office nodes, with another 26 public devices (either nodes in public areas or devices such as VCRs) available. This emphasis contrasts markedly with some other setups, where connected individuals have specifically "signed up" for the technology and the social consequences that go along with it. A significant consequence of our approach has been that concerns of privacy and control, of less immediate concern to those who've "signed up", have been particularly important in the development of our media space; and indeed, such issues must be addressed if this technology is to move out of the research laboratory.

To attack these problems, we have designed, implemented and introduced an architecture for audio-video services which provides users with a flexible mechanism for controlling their degree of involvement and with feedback on the state of their node. This system is called "Godard". Effectively, it provides users with a mechanism for tailoring their presence in the media space by analogy with social presence in a physical space.

An implementation of this architecture has been in place at EuroPARC for about a year, providing all media space services. While we have little formal evidence, it seems in general to have been successful. Feedback mechanisms are much used, and within our user community there is generally little worry about the invasion of individual privacy or the difficulty in "knowing what's going on". We attribute much of this success to the specific connection model adopted, and in particular to the way in which it causes connections to be tagged with some indication of their intent. This paper discusses the design of the system, its introduction and

1. A node comprises a video camera, monitor, speakers and microphone in addition to the user's normal workstation. Nodes are typically associated with specific offices, public areas or shared devices.

consequences, and comparisons with similar systems in use with other media spaces. Before discussing the architecture itself, it's worth looking at the infrastructure which underlies it and discuss the specific motivations which led us to build the RAVE architecture above it.

3 IIIF, EuroPARC'S Media Space Infrastructure

The basis of EuroPARC's media space infrastructure is the Integrated, Interactive, Intermedia Facility, known as "iiif" [Buxton and Moran, 1990]. The hardware component of iiif consists of an analogue audio and video network throughout the building, linking nodes in offices and public areas through a central switching facility. The switch is controlled by a networked Sun workstation which provides a simple connection service to all workstations via an RPC interface.

Essentially, the iiif mechanism provides a *point-to-point connection service*, and in the early days of experimentation, that level of service was adequate for most purposes. Interfaces from UNIX and Xerox Lisp environments, making direct use of iiif's switching facilities, provided media space users with the ability to connect offices and public areas by video, audio or both. Iiif provides a simple notion of "ownership" for device-to-device connections, which prevents interruptions and allows users some primitive protection.

However, as our network expanded and usage increased, a number of user requirements surfaced which were not directly provided by the switching service:

1. Users typically like to be aware of the usage of their node. The iiif infrastructure does not provide feedback on connections or node status; connection information is only given on demand.
2. Users distinguish between connections in terms of their dynamic behaviour, but iiif does not. Hence, there's no way for iiif to distinguish for a user the difference between a short-term or long-term video connection made by another, even though those connections reflect different activities.
3. Users cannot confirm or deny specific connection requests; in iiif, the permission mechanism is unconditional.
4. Users need the infrastructure to perform resource management (eg contentions for scarce equipment resources, etc). Iiif does not attempt to manage all of these issues itself.

The fact that iiif doesn't itself provide these facilities is not unreasonable. Iiif is designed to provide a "virtual switch" capability, and hide the details of routing topologies, port namings, etc. By analogy with the ISO networking model, we can regard it as providing the "network layer" facilities; the higher levels, organised around user-oriented connections, are to be provided by some other mechanism.

As the privacy issue was obviously an important one which had to be addressed in the development of our media space, we began the work of designing an architecture which would provide a mechanism for feedback and control in a flexible way which would enhance our research. Essentially, the goal was to build a framework in which new audio-video services could be developed; the framework would provide a standard mechanism for feedback and control, providing a uniform protection interface to users of the various services.

4 The Service/Agent Model

The architecture we developed is based on two notions: audio-video "services" that a media space network provides to its users, and the use of a tailorable "agent" to handle connection requests on behalf of users. The combination of these two provides the mechanism which is now in use at EuroPARC.

Using the low-level iiif facilities, users would make requests for connections to other users in terms of the individual *device* connections; e.g. from a colleague's camera to my monitor, or from my microphone to some speaker[1]. In developing our framework, then, one goal was to move away from this view of the network entities to something closer to the users' requirements. For instance, in connecting to another user's camera, I usually have some other goal in mind, such as to see if that person is available to talk, or if they're in a meeting or busy working. Similarly, if I make the four connections involved in setting up a two-way audio-video connection, then I think of those connections as a unit which provide me with the ability to converse with my colleague. Using this new model, then, the user's interface to the network is described in terms of the sorts of facilities which the user wishes the network to deliver—glancing into another office, setting up a video-phone call, etc.

This higher-level view of audio-video services serves a number of roles within the media space. We can particularly highlight three, which help us to tackle the problems discussed above.

1. It introduces a *negotiation* phase, when service facilities are requested, during which the user can be informed of a/v node activity.

1. These connections were often "hidden" by the user interface, but the system view was still of individual device connections.

2. It provides an *abstract, task-oriented view* of the connections which are made further down in the system. A service-level connection can be supported by a number of iiif-level connections, which can now be addressed and handled as a unit. Resource contention is much more easily managed at this level.

3. Since the notion of service differentiates individual connections, it *partitions* the space of connections, allowing users some notion of selectivity. This means that our protection mechanism can be much finer-grained than would be possible purely at the iiif level. While at the iiif-level, all connections to my camera appear identical, at the service-level I can distinguish between connections which will only last a few seconds (which we call “glance” connections), ones which might last much longer (“watch”), and ones which will be reciprocal (“window”), etc.

A Godard service, then, is a collection of iiif connections, with pre-specified general characteristics, such as length, reciprocality (*i.e.* whether or not the connection is reciprocal), methods of initiation and termination and type of protection.

Although, from a technical perspective, any use of the audio-video system can constitute a “service”, we chose first to implement and provide services which were defined analogously to real-world behaviours with which users would be familiar (such as the glance) [Anderson and Moran, 1990]. We wished to exploit users’ familiarity with these services in usage, as well as to find appropriate tailorable parameters through the real-world model.

The heart of the protection mechanism is a tailorable audio-video “agent” which acts on the user’s behalf, fielding and responding to connection requests from other users. These requests are now couched in terms of specific named services, and the agent can use information provided by the user to decide whether to accept or reject the connection. This decision can also be informed by consultation of other information sources (such as a location system based on active badges [O’Shea *et al*, 1991]) and by the context of other connections currently in progress. For instance, this might allow the system to reject a video phone call because I’m not in my office, or to reject a glance because I’m currently using my camera to talk to another user of the system.

This model of service acceptance is very general, and introduces a number of complexities in the user interface. In order to be able to specify complex patterns which collate information from a number of sources, the user is effectively performing a programming task. Most of our users are not programmers, and we would not wish them to become programmers in order to use the media space. As a result, early releases of Godard used a simplified connection model which allowed unconditional acceptance or rejection for named users, and built such facilities as user-confirmed conditional acceptance directly into the service descriptions.

5 Godard: Implementation and Interfaces

We’ll first discuss the implementation of this system, and then talk about the interface which is provided to users for making and controlling service connections.

5.1 Process Organisation

Although the model that is presented to the user suggests that each user has a private agent dealing purely with their own requests, it’s possible to centralise and share some components to provide faster access to shared status information than would be possible with separate networked processes. This approach was taken in our implementation. Figure 1 shows schematically the distribution of processing between local and central components. In Godard, the primary agent mechanism is a single server comprising two cooperating UNIX processes running on a designated workstation, although individual agent components are also provided on users’ own workstations to provide essential local services and information. This centralisation is not reflected in the interface, however, which still gives the user the impression of controlling a private, locally-accessible agent. The user need make no distinction between local and remote components when dealing with Godard.

The central agent acts as a clearinghouse for all service requests, and communicates with clients via Courier remote procedure calls [Xerox, 1981]. The Courier interface provides the programmer with an abstract service interface; the details of services are stored internally to the agent and are referenced by Courier parameters. This means that new services can be provided without changing the programming interface. There are Courier procedures to request, suspend and close connections, and to query and update the internal database. Most of the work results from service requests.

5.2 Handling Service Requests

A service request has three principal components, which can be thought of as *who*, *why* and *what*. When a user requests a service, the `request` call specifies:

1. the identity of the user who is requesting the service connection;
2. the type of service being requested;
3. the device resources (e.g. cameras, speakers) required to fill this service request.

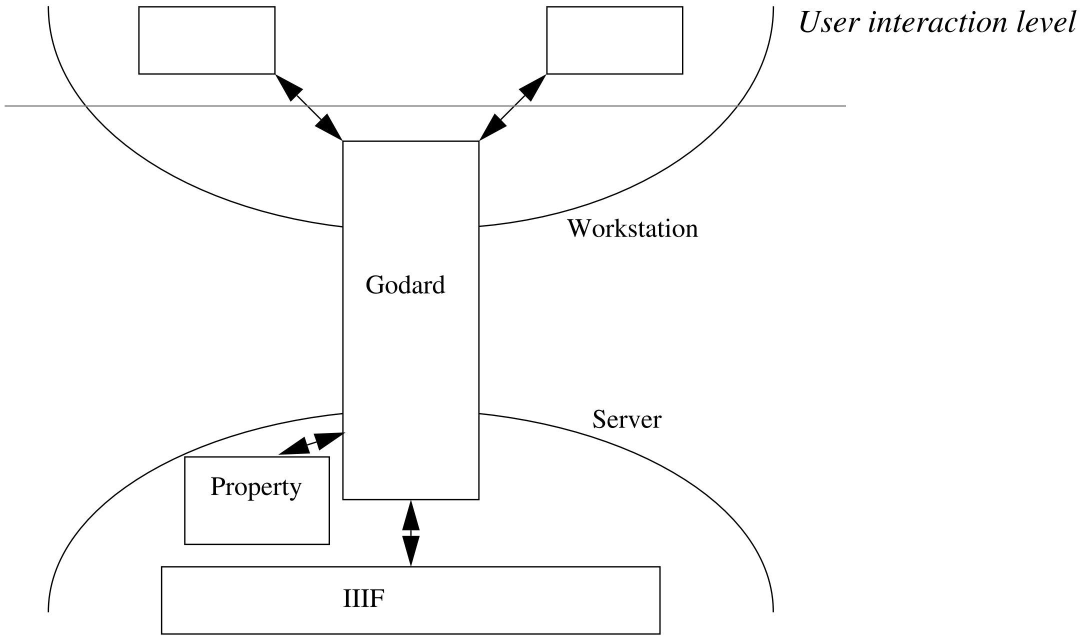

FIGURE 1. The agent mechanism is distributed between the workstation and a central server, as shown. The distinction between local and remote components is invisible to the user.

On the basis of these three parameters, a decision is made whether or not to accept a service request. Under most circumstances, the decision can be made based on information already provided by the user (see below), but sometimes it may require interactive input. For instance, device conflict resolution must be performed by the user, or services may specify an explicit acknowledgement of the request (e.g. video-phone calls, which must be explicitly "picked up" by the recipient). If the request is accepted, Godard instructs iiif to allow the connection to proceed and then passes a unique connection identifier back to the caller, to be used as a reference in all further interactions. If the request is rejected, however, then an optional rejection message is passed back to the caller, and the iiif-level permissions remain set so that the device connections cannot be made. By dynamically controlling the iiif-level protection mechanism, we ensure that Godard cannot be bypassed unless special permission has been given.

Once a service connection has been established, it remains in place until a `disconnect` signal is sent. Either party in a connection has the ability to send this signal. Disconnection causes both nodes to be restored to their pre-connection states.

A service also has to be able to deal with interruptions from other services which may require the same resources, either suspending or terminating the connection which currently holds them. If the pre-empted service is to be restored, any resources which will be needed on resumption of the connection are reserved with the agent before they are released to the interrupting service.

5.3 Control Parameters

So how do users control the behaviour of Godard? Directions for agent behaviour are recorded via a property list mechanism in the agent process.This is based upon a defined set of parameters for each service which govern the behaviour or appearance of that service for a given user. Parameters named `allowed` (a list of people granted permission to make connections of this sort to this node) and `reject` (a message which will be returned if a service request is rejected) are defined for all services. Individual services can also define other parameters which may, for example, control feedback mechanisms for connection acceptance/rejection, or list other conditions that must be met by initiators before they are accepted. The user controls the action of her agent by changing the values of these parameters. It's worth pointing out here, then, that the agent mechanism only implements features as specifically requested by a user or by a service. There are no overriding mechanisms (such as reciprocity of video connections) enforced by the system[1].

1. In our environment, reciprocity may be enforced as part of a specific service but not as a general rule for all services. *Video-phone*, for example, is inherently reciprocal, but *glance* is not.

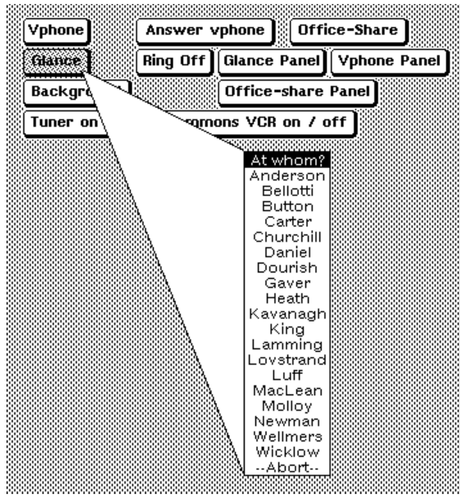

FIGURE 2. Glancing at another user with the Glance button. Buttons provide the primary interface to services, and buttons providing other Godard services are shown. The services are described in more detail in section 6.

5.4 Interfaces

At the time that Godard was introduced, the primary computing environment at EuroPARC was Xerox Lisp. So, while implementation of the central components was in C on a Sun machine, the interfaces, both for requesting services and for controlling the agent, were written in Xerox Common Lisp. Service request interfaces were provided using virtual on-screen Buttons, a tailorable interface mechanism previously developed at EuroPARC [MacLean *et al*, 1990] (see figure 2). Buttons provide a convenient mechanism for encapsulating both state and computing activity, and it was very valuable to have our users already familiar with the mechanisms for controlling, customising and sharing Buttons. The tailorability mechanism of the Buttons was used to present users with fairly general control facilities which could be tailored to provide more specific mechanisms which they used frequently.

The control interface by which users defined Godard's behaviour was based on the InterLisp FREEMENU mechanism, and took the form of a control panel which presented the parameters to users as mousable objects (see figure 3). Where possible, menu and help mechanisms were provided to assist the users in choosing appropriate options for the various parameters. One particularly useful facility is the ability to "buttonise" control panels. This facility allows users to capture the state of control panels in Buttons. Effectively, this "situated creation" mechanism provides users with a powerful macro facility that allows common agent configurations to be captured and accessed quickly. Common configurations might record default states for "generally accessible", "in a meeting", "on holiday", etc; users can simply use these Buttons rather than setting the control panel parameters individually.

Since the intent was to provide a framework in which new services could develop, we wished to make it as simple as possible for programmers to provide new services. To this end, the interface facilities for both requesting and tailoring services were not coded directly; instead, a *service compiler* was provided which automatically generates interfaces from a description of the service. This facility, together with the abstract service interface presented by the Courier library, allows us to build new services easily and quickly within the service/agent framework, which is essential for our use of the media space as a research tool.

6 Use of Godard at EuroPARC

The initial release of Godard at EuroPARC included support for six services.

1. *Background*—a long-term video connection which provides an image displayed on the video monitor when no other activity is taking place. Typically, this is used to watch the coffee room, the in/out board or the park outside.

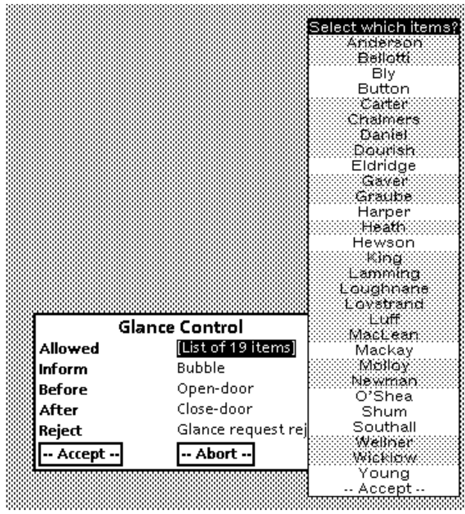

FIGURE 3. Setting the "allowed" parameter in a glance control panel. Control panels allow users to tailor the behaviour of their agents for individual services.

2. *Glance*—a short-term, one-way video connection made to a user's office. This is generally used to see if a colleague is busy, if a meeting is in progress, etc.
3. *Video-phone*—a longer-term, two-way audio and video connection between two offices. Videophone requests are signalled with a ringing telephone sound, and must be explicitly "picked up".
4. *Office-share*—a service very like video-phone, except that the initiation sequence is less complex. Office-share connections are typically of much longer duration and probably less engaged than video-phone calls, and are intended to be more like physically sharing an office.
5. *Watch*—a long-term, one-way audio and video connection, for watching devices like VCRs.

These were mostly facilities that had been available in the iiif-based environment and which, since they had gained some degree of popularity amongst our users, were deemed useful. A first set of parameters and defaults were chosen for each of these services, based on extensions of typical user behaviour. The most highly-developed service, in terms of the tailoring mechanism, was probably the "glance" facility, so we'll examine that one in particular, before and after the introduction of Godard.

6.1 Glance Service Pre-Godard

"Glance" was a facility which had been implemented in the days of direct iiif-based media space usage at EuroPARC, using various mechanisms to provide a facility whereby users could obtain a short-duration (typically a few seconds) one-way video connection into another office, and then return to whatever was previously on their screen. However, in common with other facilities which were implemented directly in terms of iiif, there was no mechanism by which users could obtain feedback about glances being made to their node. The consequences of this silent behaviour were felt not only by the recipients of glances, who were unaware that they were being observed at particular moments, but also by the initiators, who sometimes felt that they were invading the recipients' privacy.

These problems had been partly addressed by an early mechanism which watched continuously for iiif-level connections and then provided feedback to the user. This feedback was used to indicate *all* connections, irrespective of their origin, and was provided in the form of non-speech audio cues. This form of non-workstation based or *environmental* feedback has proved particularly effective for managing information in our environment [Gaver, 1991].

6.2 Glance Service Post-Godard

Godard improved on this arrangement in four ways:

1. the notion of a glance service allowed users to distinguish between glance connections and other connections to the local video camera;

2. the `allowed` parameter gave users the opportunity to select who would be allowed to glance;
3. the introduction of an `inform` parameter allowed users to be informed of the *identity* of the glancer. This information could also be gained through audio, using digitized speech rather than an on-screen message;
4. since Godard knows of connections before they're made, the feedback mechanism was properly synchronised to the video connection. This had not been true of the previous mechanism, which only noticed new connections a second or two *after* they had been established.

The effect of introducing the Godard-based glance facility was interesting. The first observation was that the existence of a correctly synchronised feedback mechanism made users feel much more confident about the system. Users who had previously felt uncomfortable about the system now participated more fully. Users who were unhappy with video cameras had previously pointed them out windows, or used lens caps to ensure their privacy. After Godard was introduced, our users began to feel considerably more comfortable about privacy issues, and most attributed this far more to the presence of the feedback and inform mechanisms than to the ability to restrict the people who could glance at them. Indeed, it remains the case at EuroPARC that most permissions are set so to allow all users to glance at personal nodes, as long as their activity is reported to the node's owner.

Interestingly, the Godard-based glance service also introduced differences for the *initiators* of glance connections. Once audio feedback and features such as digitised speech to "announce" glancers was introduced, there was an awareness that the recipient of the connection would know of the glance, and therefore that glancing at someone was essentially a *communicative act*. The process of glancing at another user could then become a way of saying, "Hello". Users would develop patterns of glancing at others when they arrived at work, partly to gain information about the others' states, but also to announce their own presence without interrupting the others' work, since a glance connection does not demand interaction in the way that a phone call or video-phone connection does.[1]

6.3 General Consequences of Introduction

We can say, then, that in terms of allaying people's concerns about privacy and control in our media space, the service/agent mechanism was quite successful. It must be stated, however, that in the long run the software did *not* cause an increase in the usage of the system, due mainly to performance considerations. Depending on the particular hardware in use, making a glance connection could take 10 seconds or more. The effect was to reduce the glance service's impact as a low-cost, lightweight connection facility. Changes in EuroPARC's computational infrastructure since the first introduction of this software should reduce this problem; the latest release of the software based on new switching technology and UNIX/X interfaces should largely eliminate the performance problems.

Since the first introduction of Godard, other services have been defined within the framework. Most of these services relate to system facilities, where connections are not directly initiated by users and the information gathered is presented in other ways (see, for example, Borning and Travers, 1991). In this way, the Godard acts as an integral part of EuroPARC's infrastructure, providing facilities not only direct to users but also to other experimental systems.

7 Other Interfaces and Approaches

EuroPARC is by no means the only laboratory investigating the use of a media space for collaboration within workgroups. While implementation aspects of the various experimental media space setups vary greatly, issues such as privacy and control are universal. It is instructive, then, to look at the solutions adopted by other research groups, and contrast them with the approach we have taken.

Before comparing our approach to those found elsewhere, however, it's worth spending some time on the issue of interfaces to such systems. Although the approach we have adopted has employed interfaces which have a very direct mapping onto the underlying operations (button/menu selection for services and form-filling for parameter setting), such an approach is by no means a requirement of the underlying architecture. The agent is essentially characterised not by the interface it presents to the user, but rather by the network protocol interface which it presents to the programmer building audio-video services. Other user interfaces which have been used with audio-video networks, such as map-based browsers and graphic systems employing analogy with face-to-face interactions can certainly be used to implicitly request Godard-style services. Indeed, we are currently in the process of changing standard interfaces at EuroPARC as our computational infrastructure changes, and it seems likely that a variety of interfaces, with very different styles, will emerge. It is important to remember this interface-independence when considering the facilities of the agent in comparison to protection and control mechanisms implemented in other media spaces.

1. It was suggested that we capitalise upon this by creating a special two-way glance (or "Buongiorno") service for this function. This has not been implemented to date.

7.1 Toronto's CAVECAT

Members of the CAVECAT group at the University of Toronto are experimenting with a media space which is very similar to ours at EuroPARC; indeed, they share a common origin [Mantei *et al*, 1991]. Like EuroPARC, the Toronto group has encountered the privacy issue in the development of their media space, and has generated a number of mechanisms by which users can control access to their office nodes. One of the most interesting mechanisms from our point of view is a rule-based approach [Louie *et al*,1991]. In this system, users provide rules which determine whether connections are to be accepted or rejected based on a pattern-matching approach. Patterns describe properties associated with connections. Various properties can be used in the pattern side of a rule:

1. duration. Connections can be *short*, *long* or *indefinite*. A glance would be a short connection; a longer-term connection of known duration (e.g. delivery of a two-minute video mail message) is long; and connections which will be terminated only by user action (e.g. video-phone) are indefinite
2. medium. Connections can be *audio*, *video* or both.
3. identity of connector.

Users use these rules, together with other facilities provided by an enhanced version of iiif, such as non-speech audio feedback (which they refer to as "earcons") and enforced reciprocal connections, to determine the behaviour of the system in response to requests from other users.

It is interesting to contrast this with the service/agent approach. One major difference is that properties associated with connections are *implicit* in the EuroPARC service model, but *explicit* in Toronto's. In other words, named services at EuroPARC provide a mechanism by which connection properties can be distinguished, but do not refer to those properties directly; instead, they deal with a more abstract notion of the *mode of use* of the connection. Toronto's approach is more general, in that new systems can be introduced and rules will already exist to cover them; EuroPARC's system, on the other hand, avoids limiting the space of possible connection types by using named services which may have different intents although their descriptions along Toronto's dimensions would be the same. For example, the Toronto dimension-based system cannot distinguish between EuroPARC's video-phone and office-share connections, nor between glance and "sweep" (a quick scan of video sources from around the building), since these connections differ in their intent, rather than in properties of the device connections themselves.

If we further investigate the notion of mode-of-use, then another important distinction arises. In the service/agent model, the use of named services places a responsibility on the initiator of a connection to identify the mode-of-use; she is making this statement when selecting the service. In the Toronto system, on the other hand, the responsibility for determining mode-of-use is placed upon the recipient of the connection, who must program her workstation to determine mode-of-use from the properties of the connection, and then act accordingly. This, indeed, is a crucial distinction. If we intend the use of named services to elicit some notion of the intent behind the connection, then only the recipient is in a position to make this clear. Toronto's system cannot provide the initiator with a mechanism for this, without providing a new "intent" or "mode-of-use" parameter for the initiator to state when making the connection. This is why there are connections in the EuroPARC environment based on distinctions which cannot be made in Toronto's. This is bought at the expense of the generality of individual rules in the Toronto scheme.

The introduction of such protection facilities within CAVECAT is at an early stage; it will be particularly enlightening to compare the usage of the two systems when both are in regular use.

7.2 Bellcore's Cruiser

Cruiser [Root, 1988] is a system developed at Bellcore to provide informal browsing in a media space environment. Users walk along "virtual hallways", encountering other system users to facilitate informal interactions. In terms of the EuroPARC system, the user sees a sequence of "glance"-style connections (a "sweep"), as well as having the opportunity to make direct connections to individuals ("video-phone").

The Cruiser designers suggest protocols by which recipients of connections can indicate their degree of "availability for interaction". These indications take the form of virtual equivalents of such real-world behaviours as closing your office door to indicate that you should not be disturbed. On-screen video overlays indicate these "availability states" to a connector; so, for instance, computer-generated graphics virtual "blinds" may obscure the view into the office. The determination of a user's availability state for a given connection is determined on the basis of a profile in which the user defines states appropriate for particular virtual visitors.

Users of Cruiser have the ability to tailor their presence in the media space, but do not have facilities to control the "appearance" of the service, in the way that we can using feedback parameters, etc. The Cruiser system is also less flexible in terms of the services it can offer to users. It is designed for a particular application (simulation of social browsing), and so some properties of that application, such as reciprocal video connections, are built into the system. Further, because there's effectively just a single application, there is no notion of a partitioning of the connection space, and different access rights granted to users on the basis of the sort of connection they request. While the facilities offered to users are perfectly adequate to support this social-browsing application, there is little flexibility for expansion of the system to provide other additional services.

7.3 PARC's Media Space (ELSE[1])

In the Systems Science Laboratory of Xerox PARC, there is another media space, known as "ELSE" or just "the Media Space", which in many ways is the intellectual predecessor of our own at EuroPARC. This system links up around 15 people, many AV devices and a few public areas. Unlike the situation at EuroPARC, it's not (currently) possible to connect all PARC staff via the Media Space, and so there's only a small subset of the possible users who are actually connected. The consequence of this is that those users who have Media Space nodes in their offices have explicitly "signed up" for the service. In doing so, they are expressing an interest to participate in the environment and accepting the social behaviours which have developed with the system at PARC. In the context of this report, the important feature of this system is that there is no user mechanism for protection or control. Connections can be established between any two nodes, one-way or two-way at any time, and the owners of those nodes need give no permission nor will they be informed directly. The owners of Media Space nodes know the social codes of the space, and so etiquette ensures that privacy is not invaded. At the same time, the node owners know that it's possible for anyone to watch them at any time; that's part of the code that they signed up for when they requested their node.

As part of our joint investigation of media spaces, we spent some time investigating the differences between our systems, particularly with regard to the appropriateness of introducing some mechanism like EuroPARC's service/agent architecture to PARC. While this was certainly technically feasible, the decision was that it would not be worthwhile. The considerations, rather than technical, were social and cultural.

A media space, such as that at EuroPARC or PARC, while an interesting technical creation, cannot be regarded or judged from a solely technical point of view. As a mechanism for enabling social interactions, the social structure of the group which it connects and their shared culture are much more important components of the system. The culture of the PARC group was such that the introduction of EuroPARC's infrastructure (iiif and Godard) would not have been appropriate, as it would have introduced restrictions into an environment where social control was the accepted determiner of behaviour. At EuroPARC, however, there was no "sign-up" process for nodes. It was necessary to fix control at a point much more towards the technical side of the technical/social dimension.

1. The relation between the names "iiif" and "else" is not entirely coincidental.

8 Conclusions

> *"Speak English!" said the Eaglet. "I don't know the meaning of half those long words, and, what's more, I don't believe you do either!"*

We have described a software architecture for control and feedback in a media space, which has been in use at EuroPARC for approaching a year at the time of writing. This architecture provides a flexible mechanism by which a uniform system of control and privacy can be made available to users of a range of audio-video services. Its flexibility is based on an abstract notion of service and controllers or agents; the power of the model is in the use of these services to add a declared "mode-of-use" to a connection in the media space. Our experience has shown that the use of feedback mechanisms, particularly using generated environmental audio cues, can be particularly effective in addressing the issues of privacy and awareness in these systems. However, a media space is essentially a social system, in which the prevailing culture and social norms are more important factors in the design of a control system than technical possibilities. Indeed, we found that our solution, although very effective in our own environment, would have been inappropriate for a similar research system.

As the use of media space technologies increases and they are introduced into a range of working environments, systems to maintain adequate privacy and control for the inhabitants of the space will be of prime importance. Once a range of such systems exist, it will perhaps be possible to generalise our notions of what constitutes adequate protection of individual privacy. Comparison of our work with that of others suggests that control based on the mode-of-use abstractions give users the opportunity to more flexibly tailor their presence in the media space, and therefore to ensure that the system delivers no more information about them than they would wish. By increasing the ability of users to tailor the system in these ways, we increase participation and therefore utility for both the initiators and recipients of connections in a media space.

9 Acknowledgements

Details of the service/agent architecture did not appear fully-formed, but rather emerged over a period of time. During this time I was greatly helped by enlightening conversations with Kathy Carter, Mik Lamming, Lennart Lövstrand, Allan MacLean, Tom Moran and Pierre Wellner. Subsequent

conversations with Victoria Bellotti, Sara Bly, Bill Gaver, Steve Harrison, Wendy Mackay and Tom Milligan have also helped greatly in understanding the significance of some elements in the Godard's design and usage patterns, and in making this paper readable and useful (I hope). Ian Daniel, Mike Molloy and Richard Southall contributed greatly to the project by maintaining the video network so well that problems were often fixed before most users even noticed they had arisen. The iiif code which underlies everything was written by Tom Milligan, now at the University of Toronto.

10 References

[Anderson and Moran, 1990] Robert Anderson and Thomas Moran, *The Workaday World as A Paradigm for CSCW Design*, in Proc. Conference on Computer-Supported Cooperative Work CSCW '90, Los Angeles, October 1990.

[Borning and Travers, 1991] Alan Borning and Michael Travers, *Two Approaches To Casual Interaction Over Computer and Video Networks*, in Proc. Conference on Human Factors in Computing Systems CHI '91, New Orleans, April 1991.

[Buxton and Moran, 1990] William Buxton and Thomas Moran, *EuroPARC's Integrated Interactive Intermedia Facility (IIIF): Early Experiences*, IFIP Conference on Multi-User Interfaces and Applications, Herakleion, Crete, September 1990.

[Gaver, 1991] William Gaver, *Sound Support for Collaboration*, Proc. European Conference on Computer-Supported Collaborative Work ECSCW '91, Amsterdam, September 1991.

[Gaver *et al*, 1991] William Gaver, Thomas Moran, Allan MacLean, Lennart Lövstrand, Paul Dourish, Kathleen Carter and William Buxton, *Realising a Video Environment: EuroPARC's RAVE System,* Proc. Conference on Human Factors in Computer Systems CHI '92, Monterey, May 1992.

[Louie *et al*, 1991] Gifford Louie, Marilyn Mantei and William Buxton, *Making Contact in a Multi-media Environment*, to appear in Proc. European Conference on Computer-Supported Collaborative Work ECSCW '91, Amsterdam, September 1991.

[MacLean *et al*,1990] Allan MacLean, Kathleen Carter, Thomas Moran and Lennart Lovstrand, *User-Tailorable Systems: Pressing the Issues with Buttons*, in Proc. Conference on Human Factors in Computing Systems CHI '90, Seattle, Washington, April 1990.

[Mantei *et al*, 1991] Marylin Mantei, Ronald Baeker, Abigail Sellen, William Buxton, Tom Milligan and Barry Wellman, *Experiences in the Use of a Media Space*, Proc. Conference on Human Factors in Computer Systems CHI '91, New Orleans, April 1991.

[Milligan, 1991] Tom Milligan, *IIIF Design Report CAVECAT Revision 1.0*, Internal Report, Dynamic Graphics Project, University of Toronto, Toronto, 1991.

[O'Shea *et al*, 1991] Tim O'Shea, Mik Lamming, Matthew Chalmers, Nicolas Graube, Pierre Wellner and Geoff Wiginton, *Expectations and Perceptions of Ubiquitous Computing: Experiments with BirdDog, a prototype Person Locator*, in BCS/IEE ITaP '91, February 1991.

[Stults, 1986] Robert Stults, *Media Space*, Technical Report, Xerox Palo Alto Research Center, Palo Alto, California, 1986.

[Xerox, 1981] Xerox Corporation, *Courier: The Remote Procedure Call Protocol*, Xerox System Integration Standard XSIS-038112, Stamford, Connecticut, December 1981.