A Distributed Location System for the Active Office

Andy Harter,* Andy Hopper*†

*Olivetti Research Limited
Old Addenbrooke's Site
24a Trumpington Street
Cambridge
CB2 1QA
United Kingdom

†University of Cambridge
Computer Laboratory
Pembroke Street
Cambridge
CB2 3QG
United Kingdom

November, 1993

Computer and communications systems continue to proliferate in the office and home. Systems are varied and complex, involving wireless networks and mobile computers. Mobility itself introduces many new issues [Weiser93]. However, systems are underused because the range of control mechanisms and application interfaces is too diverse. It is therefore pertinent to consider what mechanisms might allow the user to manipulate systems in simple and ubiquitous ways, and how computers can be made more aware of the facilities in their surroundings.

Knowledge of the location of people and equipment within an organisation is such a mechanism. Annotating a resource database with location information allows location based heuristics for control and interaction to be constructed. This approach is particularly attractive since location techniques can be devised which are physically unobtrusive and do not rely on explicit user action. This article describes the technology of a system for locating people and equipment, and the design of a distributed system service supporting access to that information. The application interfaces which are made possible by, or benefit from this facility are presented.

1 Location Technology

The topology of a fixed network of wireless receivers is the basis for a location system. The position of a mobile wireless transmitter is determined by the identity of the receivers within the network. Increasing the density of receivers allows the transmitter power to be reduced and the granularity of location to be refined. Location boundaries become well-defined if the wavelength is such that transmissions do not pass through walls. This property is exploited in the ORL infra-red network.

1.1 Room scale location

Infra-red base sensors occupy fixed positions within a building. Sensors are connected by a wire network which provides a communication path to the network driver, and distributes low-voltage power (Figure 1). Sensors are eight-bit addressable, and contain buffering to simplify protocols on the wire network. A single sensor may also be connected to the serial port of a convenient workstation.

A baseband modulated incoherent infra-red carrier of wavelength 900nm is the basic medium. Pulse-position modulation is used and a physical layer protocol defines how discrete pulse sequences form packets. A data link layer protocol defines messages and supports different devices using the infra-red network [Harter93]. A low data rate of 9600 baud allows transceivers to be small and low-powered, which is important for mobile devices.

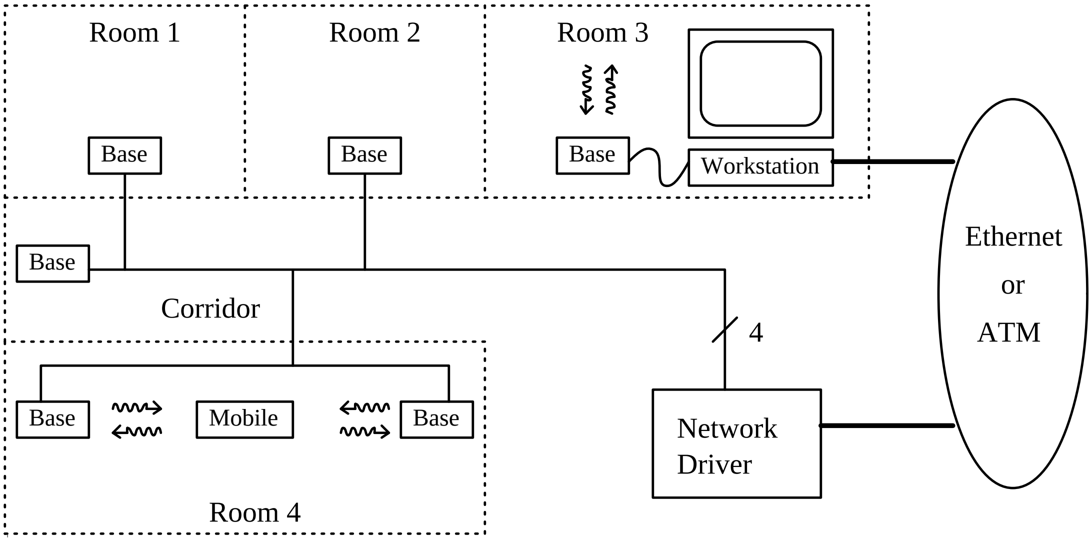

Figure 1: Infra-red sensor arrangement

1.2 Locating people

People are located to the room scale by wearing a personal infra-red transponding computer, the Active Badge[1] [Want92a,b]. Each badge periodically transmits an infra-red message containing a globally unique code. The range of the system is around thirty metres, and line-of-sight is not necessary. Pressing one of two pushbuttons causes the badge to transmit immediately and act as a primitive but ubiquitous signaling device. The badge also has a receive capability and can interpret a range of messages. A small speaker and

[1] Active Badge is a trademark of Ing. C. Olivetti & C., S.p.A.

two visible LEDs provide a basic paging facility. A small amount of internal state can be set and subsequently read.

The power budget is managed by only transmitting every ten seconds, which is in keeping with the rate at which a person might move between zones. A light dependent component increases the transmission interval in darkness, for example when a badge is in a drawer or pocket. In addition, the receiver is only enabled for a period after each badge transmission. Base sensors are able to store-and-forward messages at this rendezvous (Figure 2). Badges run continuously on conventional batteries lasting around one year.

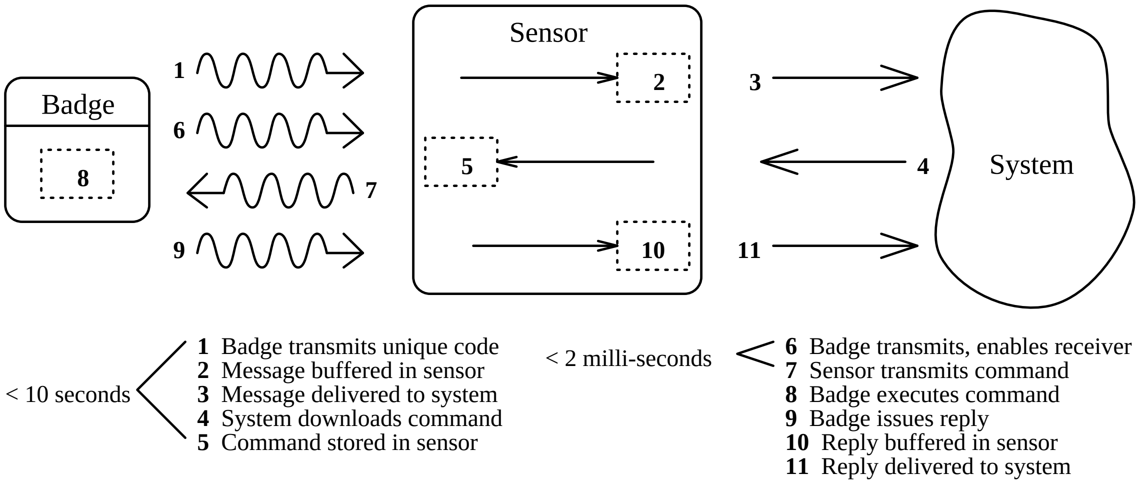

Figure 2: Active Badge communication sequence

1.3 Locating equipment

Equipment is located by attaching an infra-red transmitting computer derived from the Active Badge. The equipment badge is required to run continuously for several years, which is achieved by increasing the transmission interval to five minutes, and removing the receive capability. Much equipment is quite static and it is reasonable to wait a while for the system to realise equipment has moved. The equipment badge has a socket allowing two digital inputs to be monitored. The input values are reported in the periodic transmission, providing wireless telemetry of equipment state.

A well known wireless communication problem is the reception of the same transmission at multiple bases. The problem is apparent with the infra-red network, where a zone may have a number of bases. Transmissions for both types of badge contain a sequence number, allowing duplicates to be filtered at any level in the system.

1.4 Desk scale location

Finer grain location is supported with a hybrid radio/infra-red scheme. Low-powered radio transmitters generate a low-frequency carrier which is pulse-width modulated with one of a discrete set of widths. A passive tuned circuit in the badge detects the field, which is sampled and coded into each infra-red message. Overlapping fields can be detected but not resolved. Typically transmitters and aerials are placed so as to form zones surrounding desks (Figure 3). The range of the field is controlled by transmitter power and aerial size, and is adjusted to correspond roughly to an arm's length. Badges are stimulated to transmit immediately on first entering a field.

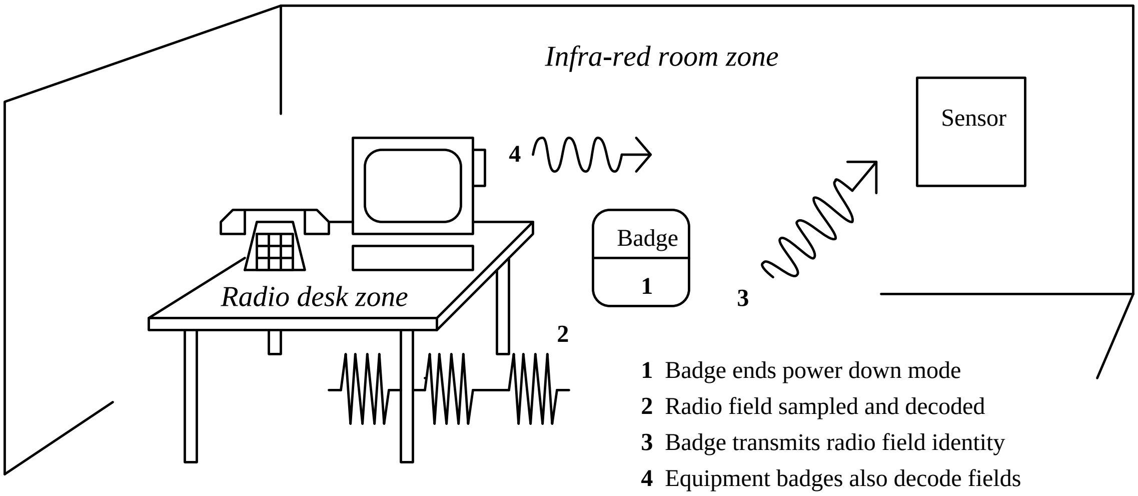

Figure 3: Desk scale location

1.5 Properties

The location technology has a number of properties which influence the design of a distributed software architecture. First, the spatial granularity correlates with natural boundaries, which define convenient units of interaction. Second, the temporal granularity is as fine as the power budget of a continuous active technology allows. An important point is that location information has a real-time element. If information is not timely, it may be of no use. Third, information is actively generated and regularly broadcast. The system can then cache information and the last known location of people or objects is available on demand.

1.6 Deployment

Over 1500 badges and 2000 sensors are deployed in the research community at ORL, a number of European universities, DEC research laboratories, Bellcore and MIT Media Lab. The largest single system is at Cambridge University Computer Laboratory, where over 200 badges and 300 sensors are in daily use. Equipment badges and desk scale location technology are deployed at ORL, where around 200 items of equipment are badged, and each of some 50 or so desks is identified. To a reasonable level of both space and time, the location of people and equipment is therefore known.

The technology has proved to be an extremely effective tool in the construction of large-scale experiments to discover some of the properties and uses of location information. It has been both a focus for designing and building software systems supporting location, and an inspiration for creating location-aware applications. These activities at ORL are described below.

2 Location System Architecture

The main thrust of the design is towards a scalable architecture. Location, naming, distribution models and transaction management, are all influenced by this goal.

2.1 Location

Location can be viewed as an attribute of an object. In abstract form, the value of 'location' is a key of no particular importance other than to relate co-located objects. In a more direct form, the value of 'location' has meaning by itself, being chosen from some topological or geographic naming scheme. Efficient searching of the object space by location key is required, since the co-location function will be evaluated often by contextually aware applications. The location attribute is dynamic, and the mechanism to store it must support a high frequency of updates. It is therefore infeasible to construct global databases of location information. Deciding where to cache fine-grain information is difficult, since the relative frequency of updates and queries must be balanced against communication cost. Experience suggests that location caches are often best constructed close to the source of the information, which is usually the source of most queries.

2.2 Naming

Another attribute of an object is the name or names by which it is known. Name servers manage naming information, and hierarchies of name or directory services provide global naming. At any time, there is a mapping between object, name and location. The location attribute is actually a pointer to a location service for the named object, and an object may be located by different services during its lifetime. For the Active Badge, the internal state is manipulated as an address hint for a service currently responsible for locating it. Name spaces will need to expand when location technology matures and provides finer grain location of more objects. For example, it is currently uninteresting to register all pieces of paper in the office with a name server, since in general they are not automatically locatable. This may not always be the case.

2.3 Distribution models

A major architectural choice is the degree of centralisation of responsibility for location information. One view is that responsibility is devolved to the level of the individual, with personal agents encapsulating knowledge of and access to location information [Spreitzer93]. A less fashionable view is that of services with corporate responsibility.

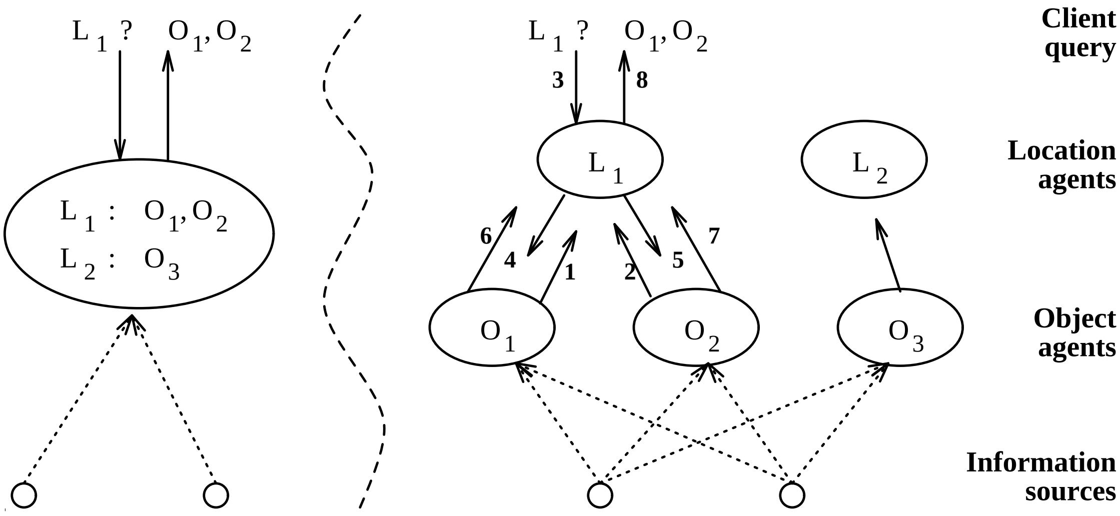

Figure 4: Contrasting styles of location information federation

Figure 4 highlights some potential problems of local scale with the functionally decomposed agent approach. First, communication channels must be managed between information sources and object agents. Second, keying by location requires the introduction of an agent for each location (L_1, L_2). Location directed queries can trigger many transactions (1...8) since individual agents maintain responsibility for access control. Third, the evaluation of predicates involving several objects is forced into the client. In fact, some types of query are hard to evaluate outside centralised services, such as the list of objects near to a location. An even harder example is the list of locations with more than a given number of objects of the same type, since typing information is itself the subject of access control.

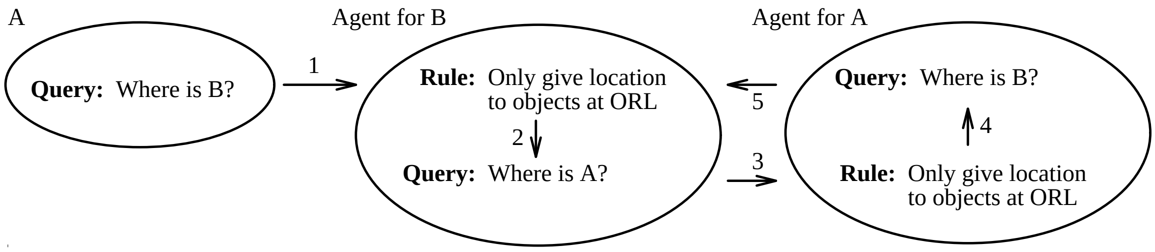

Figure 5: Cyclic recursion in location dependent access controls

The choice of access controls requires considerable care, particularly if access controls depend on location information. Even apparently straightforward cases create difficulties if rules are strictly encapsulated in an agent (Figure 5). There are several ways round this particular problem, for example the protection boundary can be moved to enclose both agents. Alternatively, reciprocity can be enforced to break the cycle, rather like caller-id for incoming telephone calls. A more complex, but still entirely realistic rule such as 'only give out location to objects within 50 feet' combines the access control and proximity problems.

2.4 Transaction management

Within a single administrative domain, large numbers of clients require distributed access to location information. This can either be by client server query or, to avoid polling, by waiting on a callback. Pushing event filtering and predicate evaluation from client to server reduces the total amount of transaction traffic in the system, at the cost of increased server load. For callback streams, a protocol delivering only changes in location can significantly reduce traffic to the client. The protocol must indicate when the object is no longer being located. This requires the concept of a minimum rate of location, which is a quality-of-service parameter settable by the client at callback establishment. More subtly, if the location of a group of objects is only required to the nearest minute say, intermediate changes in location do not need to be communicated.

3 Location System Design

Figure 6 depicts software elements in the design of a location system for the local area. A principle of the design is the model of the system as a single autonomous entity within a domain or enterprise. Some issues are explicitly excluded from the design, including sophisticated access control within the organisation, and support for replicated services.

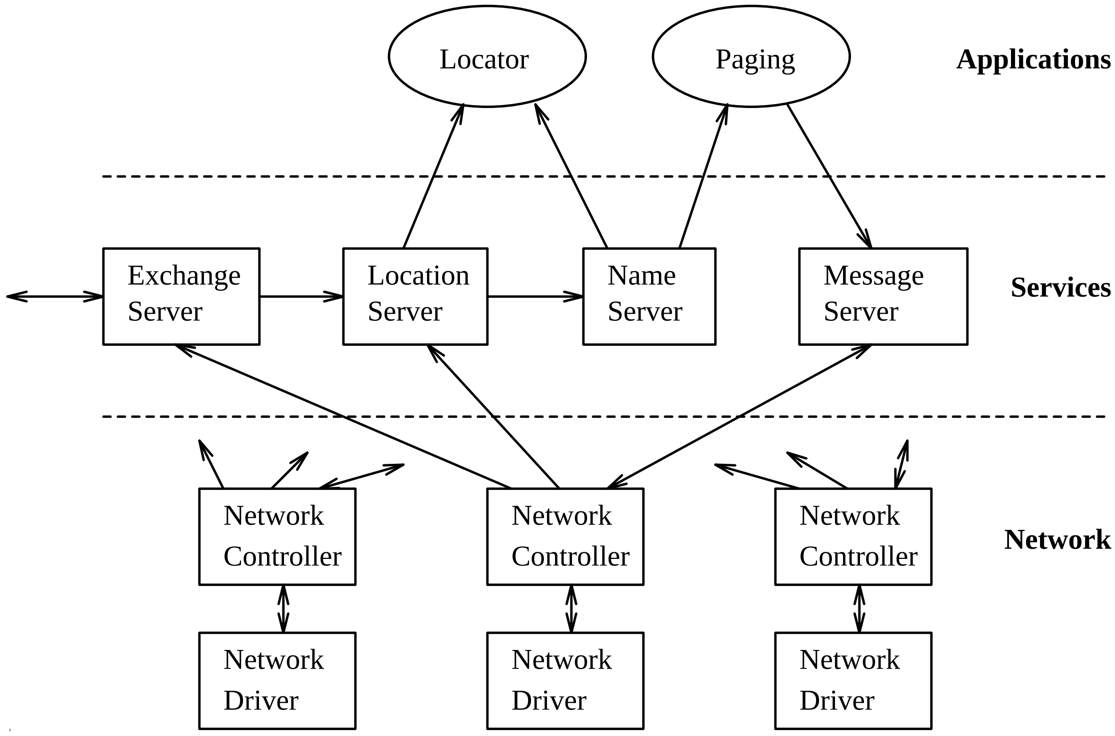

Figure 6: System design for a single administrative domain

3.1 Network driver

The network driver is a lightweight entity which encapsulates the protocols used on the wire network to communicate with sensors. It is designed to be simple enough to run on a range of equipment, from workstations to small embedded microprocessor systems. The topology of the organisation will determine the number and distribution of sensors, wire networks and network drivers. Each network driver communicates with one network controller.

3.2 Network controller

Large installations require more sensors than can be uniquely addressed on a single wire network. The network controller is responsible for adding an addressing component to distinguish sensor addresses on different networks. Another function is to time-stamp incoming location information. It is assumed that controllers, services and applications are run on parts of the computer system infrastructure capable of maintaining synchronised standard time. This is achieved locally by running daemons implementing the Internet Network Time Protocol [Mills89].

Each network controller supports concurrent communication with different classes of server. At communication establishment each server specifies a filter for the type of information it wishes to send or receive. The collection of servers communicating with the network controllers is dynamic, and application specific servers can be constructed which filter information according to precise application requirements. This avoids overloading the functionality of any one server. Filtering within the network controller rather than the server avoids generating redundant network traffic.

3.3 Location server

The location server connects to the network controllers with a filter requesting Active Badge and equipment badge information. The server maintains a cache of the last piece of location information for every badge detected. The unit of location consists of badge address, location address and a time-stamp. A location address consists of domain name, network address and sensor address. By careful choice of domain name, for example an internet address, the unit becomes representative of a globally unique event.

Simple clients start, interact with the system and terminate. An interface allows these applications to invoke queries about badge locations, and about location contents. More complex clients run indefinitely, and require streams of information to be delivered efficiently. A second interface allows these applications to specify a filter and callback.

3.4 Name server

Applications require names for badge and location addresses for textual presentation, and to frame location server queries from user input. Sophisticated applications managing computer or communications interaction also require attribute or resource lists for each type of name. A query interface provides applications with lookups by name or by address for badges, equipment, locations, and domains. A filter and callback interface communicates name changes, allowing long running applications to efficiently maintain a name cache. Sequence numbers are used to ensure cache consistency.

3.5 Exchange server

The exchange server is responsible for the distribution of information between organisations. Issues of security, access control and information flow between organisations are encapsulated in this service. Communication to other exchange servers is by message passing over TCP. Security is enforced by checking the address and port number of incoming and outgoing calls. Each badge wearer has access controls used by the exchange server to determine which external organisation may receive their information. Exported information is freely available within the remote organisation.

3.6 Message gateway

The gateway is responsible for coordinating message transport to mobile devices. The service encapsulates the complexity of routing and re-routing to mobiles, and the different transport characteristics of a range of devices. Active Badges have periodic receive capability and require store-and-forward techniques. Infra-red network interfaces to notebook computers have continuous receive capability and offer direct communication. The gateway schedules competing requests for Active Badge messages to be delivered in the same infra-red zone.

The underlying technology does not support acknowledgement for message delivery, and mobiles must therefore generate explicit replies. The inherent unreliability of the wireless network means that a reply may not be received, and the gateway has no choice but to repeat the message. A sequence number is included in outgoing messages, enabling mobiles to interpret messages idempotently. The client interface to the gateway includes a per-message timeout, and a callback for success or failure.

4 Location System Implementation

Facilities are required to manage communication between servers and applications. This should include support for remote procedure calls and late, run-time binding to services.

4.1 Distributed system support

The Advanced Network Systems Architecture Testbench is a platform for creating distributed applications [ANSA90]. Remote procedure call is the fundamental communication paradigm. Procedure signatures are specified in interface definitions. Software declared the server of an interface must implement the procedures it defines. Software declared the client of an interface can invoke the procedures in a server which offers the interface. Interface definitions allow the platform to perform strong type-checking of procedure invocation and implementation at compile time. They also allow argument marshalling and communication code to be automatically inserted at both ends of the call. Exception code can be supplied in a client for each invoked procedure.

Interfaces are created at run-time and a reference is exported to a public repository service, the Trader. A list of attributes are associated with each interface reference. Interface references are obtained at run-time by importing them from the Trader, based on some predicate of attribute values. An interface reference is necessary and sufficient for invoking remote procedure calls. Software which is the server of an interface can create a reference and pass it privately to another entity which will be a client of the interface. This is the basic mechanism by which callback streams are implemented.

The platform provides two remote procedure call semantics. The exactly-once semantic requires a server reply, while the at-most-once semantic does not. For procedures which do not have results, there can be significant protocol savings if the at-most-once semantics are used. Many applications treat location information communicated by callbacks as a hint and do not require reliable transport. The at-most-once remote procedure call semantic is used in the interface for these applications. The callback interface must then include an exactly-once ping operation, since the server has no other way of periodically purging stale interfaces. There is support for concurrent programming in the form of non-preemptive threads, and support for synchronisation in the form of event counts and sequencers.

4.2 System management

Creating a robust implementation requires the dependencies between servers and applications to be understood. The system must tolerate any order of server starting or restarting. Consider the management of the callback stream between a network controller and location server. If the network controller starts first, the location server can immediately register the callback. If the location server starts first, the network controller announces itself to the location server which is prompted to invoke callback registration. By making callback registration idempotent, there are no adverse race conditions. All relationships of this type are made similarly commutative.

Another aspect of reliability and system management is tolerance to computing infrastructure failure. Although the design does not include service replication, the implementation allows manual relocation of servers within the infrastructure. When a remote procedure call fails, the client discards the interface reference and obtains a new one by reconsulting the Trader. Stale interface references are purged automatically from the Trader, and so a replacement interface reference eventually percolates through the system.

An important requirement is the management of software upgrades. All software components have a version identifier compiled in. Servers export the version identifier as an attribute, enabling clients to make a run-time compatibility check on the imported or re-imported interface. The action on detecting incompatibility is client dependent, but typically applications terminate. This prevents long running applications from attempting to rebind to incompatible new interfaces.

4.3 Commentary

A distributed location system implementing the above design has been in use at ORL and the Computer Laboratory since early 1991. The system has proved to be highly successful, and has facilitated the building of a wide range of applications. Information is continuously exchanged, which streamlines communications between the organisations. ANSA has proved to be an extremely effective tool. Remote procedure call, support for concurrent programming, and sophisticated interface management have been major benefits. An added advantage is that the platform inter-operates between Sun, DEC, and HP architectures.

5 Applications Using Location

So many location-aware applications have been tried that an exhaustive list is beyond the scope of this article. Applications which prove useful and endure are now described.

5.1 Presentation and security

The simplest application presents textual information about people, equipment and locations in response to user queries at the command line. It endures because it is straightforward and can be run on any sort of terminal. Badge location can be used for security, and processes to guard workstations and doors provide hands-free access control.

More sophisticated applications make use of the X Window System[2]. A graphical user interface presents a hierarchy of two-dimensional floor plans annotated with badge locations. The map provides visual cues for proximity, which is less easy to present textually. The most generally useful application presents a hypertext style interface. The top level describes the location of people and equipment, arranged by organisation. Selecting a person displays as much information as possible about their current context, such as a list of other people and equipment present, and a window of communication methods which are currently available.

[2] The X Window System is a trademark of Massachusetts Institute of Technology

5.2 Location sensitive communications

The communication window (Figure 7) gives the caller a snapshot view of the callee's communication profile. The profile uses rules to determine which communication methods are appropriate, based on the identity, location and communication resources of both caller and callee. Access methods are provided when communication is applicable, allowing call-by-name. This obscures details such as telephone number, which overload properties of naming, addressing and routing. Video-phone service is provided by the Pandora system, which integrates multiple real-time audio and video streams into the workstation [Hopper92]. Some twenty or so Pandora systems are distributed between ORL and the Computer Laboratory, and applications make full use of the interworking location systems. The future absence messages replace the location text at the appropriate time. The messages are automatically cancelled when the badge wearer is located thereafter, providing a convenient method of managing an active diary.

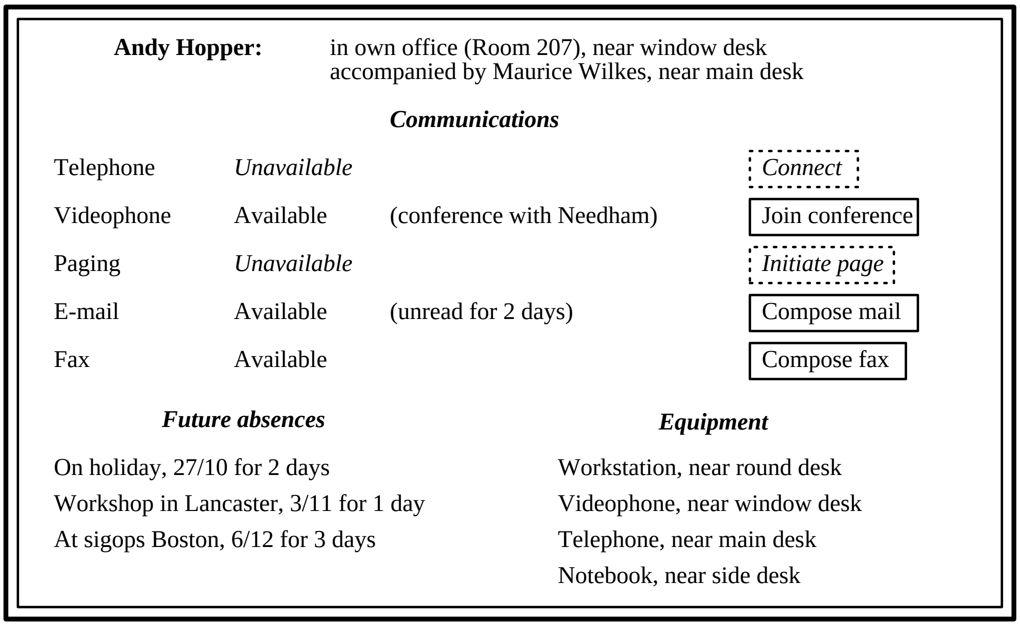

Figure 7: Snapshot of a communication window

5.3 Global communication service

External callers are given communications context by the Unix[3] program 'finger'. The finger service has been extended at ORL to include the location of those users for whom it is authorised. It is possible to check that the caller is fingerable before replying, which is an example of a reciprocal privacy arrangement. The callee is notified of all finger invocations using the paging facilities described below. Any machine with suitable internet access can invoke the service with the command

```
finger @badges.cam-orl.co.uk
```

[3] Unix is a trademark of Unix System Laboratories.

The service is popular, having been invoked thousands of times from hundreds of different internet machines around the world. There is anecdotal evidence that it is used for communication purposes in addition to curiosity.

5.4 Location oriented paging

A variety of note sequences can be sounded on the Active Badge speaker as an audio paging cue. It is interesting to integrate paging with system services such as finger, which triggers the 'finger' cue for each incoming call. Another example is e-mail delivery, which triggers the 'mail' cue. A nuance is that the user is only paged if they are away from their workstation, which otherwise provides an audible cue itself. An alarm function is raised when a user specified location predicate becomes true. The 'alarm' cue is paged, and a message posted on the user's display. Those parties causing the predicate to be true are paged with the 'watched' cue, which is another example of reciprocity in the system.

Audio cues alone for alarms and e-mail turn out to be tantalising. The immediate thought is 'Which alarm?' or 'Mail from whom? And about what?' A solution is for the system to store a textual hint for each page. For example, when e-mail is received the name of the sender and the text of the subject line are stored. The hints are retrievable by program, requiring the user to login to a nearby workstation. A method of making the output available in a reasonably secure way without the overhead of login is described below.

6 Applications Using Location and Input

The Active Badge has two pushbuttons, allowing it to be used as a simple ubiquitous signaling device. This provides a control mechanism for location-aware applications.

6.1 Ubiquitous application control

A menu of applications appears on the nearest X display when the left badge button is pressed. The identity of the display is determined by the location information generated by equipment badges. Further button presses scroll through the menu, and a double press invokes the selected application. The list of applications is personalised, and is also configured by context of location and display type. The default list includes a location tool, paging hint retrieval and communications applications. Some applications invoked in this way make use of further button events, particularly those with uncomplicated user interfaces such as the video-conference. Applications launched by badge are automatically terminated when the initiator leaves the vicinity of the display.

It has been instructive to see how much can be done with the simplest possible input device. In this mode, the badge acts as a personal remote control. It is particularly attractive, since the user simply presses a button to temporarily bootstrap familiar functionality in an unfamiliar environment. Furthermore, on display-only devices such as a video tile, some applications can be completely controlled with the badge alone. Using the badge button in this way becomes quite natural, and it is interesting to review the whole computer and communications environment in this light.

6.2 Transportable desktops

A complimentary approach to managing applications is to make them able to follow the user around. For X Window System applications, this can be achieved by making the user's X session relocatable to other displays. So called 'teleporting' can be approached in a variety of ways. One of the most flexible is to pipe the session through a proxy server. An advantage of the proxy approach is that it can be made to work transparently for all X applications, with sufficient attention to subtle detail. An implementation is in daily use at ORL [Richardson93].

The location system is of paramount importance in making the system easy to use. A right button press indicates to the proxy server that the user wishes to teleport. The location system presents the list of X displays at the user's current location to the proxy, which uses context sensitive heuristics to select a display on which to materialise. The teleport session automatically de-materialises when the user leaves the vicinity of the display. Integrating location information into this powerful tool has generated more thought about ways in which the large amount of context which the system provides can be harnessed. The sophisticated interaction of Active Badge, equipment badge and resource database makes the user interface to teleporting particularly appealing.

6.3 Contextually aware mobiles

An example of a mobile device running location-aware applications is the Quaderno, a notebook PC. A prototype interface to the infra-red network gives the notebook general purpose wireless communication with the system. A PCMCIA card interface is being developed. The notebook periodically transmits, providing a basic location function. A Quaderno server implements a reliable transport protocol providing bidirectional communication between the notebooks and clients of the server.

The notebook demonstrates knowledge of the environment with an application showing the co-located people, equipment and resources. A location-aware printing facility is an example of the notebook making intelligent use of this information. A service enables a file on the notebook to be transferred to the system over the infra-red network, and spooled to the nearest suitably resourced printer.

Other purpose built devices have been connected over the infra-red network. The Xerox ParcTab is a palm sized computer with a touch sensitive display [Adams93]. The ParcTab is primarily a graphics terminal, and relies on remote hosts for running applications. One possible use of the device is as a ubiquitous interaction and control tool, providing a more sophisticated first level of interaction than the Active Badge.

7 The Active Office

Future office and home environments will contain many computer systems. Communication mediums will be much more widespread and diverse, offering fibre, wire, and wireless networks. All computer systems will be networked, and a homogeneous architecture will allow inter-operation. Embedded systems or applications will be rare, since simple ubiquitous human interfaces are only possible in a truly distributed environment.

7.1 Role of location

It is easy to see how the distributed location system and applications relate to the architecture necessary for this level of integration. Our experience is that providing context is important. Knowing where people are, and the computer and communications context in which they find themselves is the key principle for building location-aware applications. Mobile devices themselves can also determine their context, and so run location-aware applications. Contexts must be maintained automatically using location technology, since it is difficult to conceive of a world where they are manually administered.

7.2 A distributed mobile systems architecture

Work is in progress to define some of the architecture for the active office. One of the goals of the architecture is to provide a system which maximally supports location-aware services and computing. Further research will be in several key areas, particularly the federation of location systems and global system inter-operability. This will involve a fresh look at issues of privacy and security. Some of these issues are now felt to be better understood as a result of our experiences. Another area is the description of location events in a more general space-time framework, which will support complex queries involving proximity. A feature of the architecture will be the definition and support of quality-of-service at all levels, from basic queries through to negotiation of inter-operability requirements.

8 Summary

Location of people and equipment has become a totally indispensable part of the ORL environment, due to the large number of location-aware systems and applications in daily use. The large-scale deployment of location technology has given many insights into the kind of complex distributed systems which are at the core of the active office. Work is in progress on new architectures for supporting the requirements of next generation office systems.

9 Acknowledgement

Our thanks go to colleagues at Olivetti Research and the University of Cambridge Computer Laboratory, whose efforts have made much of the work described in this article possible. At various times, Roy Want, Tom Blackie, Mark Chopping, Damian Gilmurray, Frazer Bennett, Joe Dixon, Mark Hayter and Roger Needham have all made a contribution. Cosmos Nicolaou and John Porter have suggested many improvements to the text.

References

[ANSA90] Advanced Networked Systems Architecture. *Testbench Implementation Manual.* Architecture Projects Management Limited, Poseidon House, Castle Park, Cambridge, England. August 1990.

[Adams93] Adams, N., Gold, R., Schilit, B., Tso, M., Want, R. *An Infrared Network for Mobile Computers.* Proceedings of the USENIX Mobile & Location-Independent Computing Symposium, Cambridge Massachusetts, August 2-3 1993. 41-51.

[Harter93] Harter, A., Bennett, F. *Low Bandwidth IR Networks and Protocols for Mobile Communicating Devices.* ORL Technical Report 93/5.

[Hopper92] Hopper, A. *Improving Communications at the Desktop.* Report of the Royal Society. Discussion Meeting on Communications After 2000 AD. Chapman and Hall, 1992.

[Mills89] Mills, D., *Network Time Protocol (Version 2) Specification and Implementation.* Network Working Group Request for Comments: 1119. September 1989.

[Richardson93] Richardson, T., Mapp, G., Bennett, F., Hopper, A. *Teleporting in an X Window System Environment.* Submitted to IEEE Personal Communications Magazine.

[Spreitzer93] Spreitzer, M., Theimer, M. *Scalable, Secure, Mobile Computing with Location Information.* Communications of the ACM, Vol. 6, No. 7, July 1993. 27-27.

[Want92a] Want, R., Hopper, A., Falcão, V., Gibbons, J., *The Active Badge Location System.* ACM Transactions on Information Systems, Vol. 10, No. 1, January 1992. 91-102.

[Want92b] Want, R., Hopper, A. *Active Badges and Personal Interactive Computing Objects.* IEEE Transactions on Consumer Electronics, Vol. 38, No. 1, February 1992. 10-20.

[Weiser93] Weiser, M. *Some Computer Science Issues in Ubiquitous Computing.* Communications of the ACM, Vol. 6, No. 7, July 1993. 75-84.

Biographies

Andy Harter received the B.A. and Ph.D degrees in computer science from the University of Cambridge. His doctoral dissertation "Three-Dimensional Integrated Circuit Layout" was judged by the British Computer Society to be the most distinguished of 1990, and was published by Cambridge University Press. Since joining Olivetti Research he has been responsible for distributed location system projects, and maintains an interest in VLSI.

Andy Hopper is Director of Corporate Research at Ing. C. Olivetti & S.p.A., Reader in Computer Technology at the University of Cambridge, and Fellow of Corpus Christi College, Cambridge. His research interests include computer architecture, ATM networking, VLSI, and multimedia.