Teleporting - Making Applications Mobile

Frazer Bennett, Tristan Richardson, Andy Harter

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

Abstract

The rapid emergence of mobile computers as a popular, and increasingly powerful, computing tool is presenting new challenges. This subject is already being widely addressed within the computing literature. A complementary and relatively unexplored notion of mobility is one in which application interfaces, rather than the computer on which the applications run, are able to move.

The Teleporting System developed at the Olivetti Research Laboratory (ORL) is a tool for experiencing such 'mobile applications'. It operates within the X Window System[1], and allows users to interact with their existing X applications at any X display within a building. The process of controlling the interface to the teleporting system is very simple. This simplicity comes from the use of an automatically maintained database of the location of equipment and people within the building. This paper describes the teleporting system, what it does, and how it is used. We outline some of the issues of making applications mobile that have arisen during its implementation.

1   Introduction

A great deal of research is being carried out to support the needs of the growing mobile computer industry. Effort is going into display and disk technology, pen systems and power management. Wireless networks, mobile routing and mobile services are key to making mobile computers a part of the distributed systems fabric. However, if we allow our mobile working to be limited by the facilities of the particular mobile device we choose to carry, then whole classes of application are denied us. Examples are applications requiring a large high resolution full-colour display, multimedia peripherals, or high bandwidth communication. For these, it is more appropriate to think how the mobile user may make temporary, but personalized, use of the richer facilities in the static computer and communications background.

We arrive at the notion of the *mobile application*, one whose user interface is able to be dynamically mapped onto the resources of the surrounding computer and communications facilities. Whilst such an application remains active and resident on a particular computer, it is possible to interact with its user interface wherever suitable equipment is available. The paradigm is therefore that users' computer and communications interfaces *follow* them, rather than being carried by them.

The idea of mobile applications can be extended further. With the appropriate control mechanisms, we can imagine using the public information appliance of the future to interact with our mobile applications. Such an appliance might offer a display, keyboard and pointing device. Mobile applications, resident elsewhere, would be able to make temporary use of these resources on demand.

The *Teleporting System* was developed as a means of experiencing mobile applications, without the need to re-write existing applications or design new ones. It provides a mechanism for transparently introducing the mobility element into existing, familiar, unaltered applications.

2   The Teleporting System

This section describes the teleporting system developed at ORL, and how its simple control interface has encouraged its use throughout the laboratory.

2.1   What it is

Our vehicle for experimenting with mobile applications is a system allowing users to dynamically change the display device from which their currently running applications are accessible. It does this by introducing a level of indirection between the processing of an application and its interface. The windows that make up the interface of an application can be removed from one display and re-created on another, as the user dictates. This act is described as *teleporting* between displays. A simple control interface makes it

---

[1] The X Window System is a trademark of the X Consortium [1]

possible for the user to control where and when teleporting occurs.

2.2 How it works

The teleporting system works within X, the windowing system commonly in use at ORL. Our implementation exploits the client/server model that X adopts. Applications, as clients of an X server, typically make a connection to this server for the period of their lifetime. This connection provides the means of creating windows on the screen, and receiving input from the keyboard or mouse.

The main component of the teleporting system is a special X server, known as a *proxy server*. Users of the teleporting system each have their own proxy server. Their chosen applications are made mobile by running them as clients of the proxy server, within a *teleport session*, rather than within a traditional *X session* under a real X server.

Unlike a real X server, the proxy server does not have a screen, keyboard or mouse (collectively known as a *display*) of its own. Instead, it is able to make use of the display of some real X server. It does this by representing its clients to the real X server, appearing to the server just as they would. In this way, the clients' output will be sent to the screen of the real X server, and their input will come from its input devices.

The proxy server makes its clients mobile because it is able, upon request, to break down its connections with any real X server and, if desired, re-build them with another. This occurs entirely without the clients being aware, and the result is that the proxy server's root window, and all its clients' windows disappear from one screen and (possibly) re-materialize on another.

The proxy server must also ensure that when breaking down and re-making connections between different servers, the integrity of these connections is maintained. For the clients to be unaware of this change in server connection, the proxy server must bring any new connections to the same state as corresponding old ones before allowing clients to proceed with new interaction with the server. To ensure the transparency of these changes in connection, the proxy server must also maintain a consistent view of the particular characteristics of the server to which it is connected, despite these characteristics being different between servers. Some of the subtleties involved in this task are described elsewhere [2].

3 Using the Teleporting System

In order to be able to exploit the mobility that the teleporting system introduces, we face some additional requirements. We need to be able to choose and specify appropriate equipment that we can teleport to, as well as having a means of authenticating ourselves to the proxy server,

to ensure security and to confirm our identity before teleporting.

The proxy server offers a simple textual command-line interface which meets these requirements. In addition, its integration with the location facilities developed at ORL provide a far richer and far simpler interface. We outline both control mechanisms, and describe how, with the appropriate connectivity, it is possible to teleport between the office and the home.

3.1 The simple control interface

The command-line interface to the proxy server offers one single command, `teleport`. This command is used to initialize a teleport session, as well as control its materialization to, and de-materialization from, any X workstation within the building.

By quoting a user name and password, it is possible for individuals to materialize their teleport sessions alongside the X session of someone else. In this way, by sharing the resources of a single display, teleporting offers a mechanism by which people can easily share information.

3.2 The automated control interface

Integration with the Active Badge Location System[2] [3] [4] has introduced a level of automation to the teleporting procedure. Teleporting has become a trivial activity - far easier, indeed, than using X servers in a conventional manner.

The Active Badge System provides a means of locating individuals and equipment within a building. Personnel and equipment wear a small infra-red transmitting *Active Badge*, which has a unique identity. With a networked infra-red receiver in each room, the Active Badge system is able to create a dynamic database describing the location of individuals and equipment within the building. The proxy server uses this database to determine which X displays are co-located with a particular individual.

In addition, the Active Badge can be used as the control mechanism for the teleporting system. One of two small buttons on the badge is used to indicate the selection of an appropriate X server within the room, and the other is used to confirm the selection and initiate teleporting.

Furthermore, the badge acts as the required authentication mechanism, since it has a unique identity, and is generally worn by its owner at all times. When a user leaves a room in which they have teleported, it is possible for the proxy server to treat this as an indication that they wish their teleport session to disappear from view. This happens automatically, and represents an additional security feature.

---

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

3.3 Teleporting at home

Redirecting windows from office to home is a particularly obvious example of how useful teleporting can be. The homes of a number of ORL staff are equipped with a workstation connected to the office by ISDN[3]. With a connection to the Active Badge system, the home setup becomes the same as that in the office. On arriving home, the normal control mechanisms can be used to materialize a teleport session, and work can continue where it left off.

4 Issues in Teleporting

Our first experiences of mobile applications have prompted much interest, and many new questions have arisen. We outline a couple of areas which will need further consideration before a ubiquitous platform for mobile applications can be realized.

4.1 Teleporting at large

Clearly, issues arise when we consider teleporting between different domains of administration. A particular site may wish to make the facilities of an X display available for teleporting purposes without providing local accounts to all those who wish to teleport. Such a site may wish to offer this facility as a service to others, whilst ensuring local system integrity.

Two important considerations come into question. We need both an authentication and an authorization mechanism. The authentication mechanism may be a two-stage one, in which an individual provides authentication to both the local and remote systems. When satisfied with the authentication, some means of authorization must be passed between sites before teleporting can take place. To this end, we have implemented a simple 'global teleporting' mechanism by use of the X *Magic Cookie* authorization technique.

4.2 Location-aware applications

The proxy server goes to some length to hide from clients the fact that their application interface has moved. However, with mobility inherent, we can now envisage applications which exploit their knowledge of the location and surroundings. Appropriate support for this new class of 'location-aware applications' must be provided within the programming environment.

Presenting information about the location of people and equipment to an application in a suitable form is an area of some interest to us. At ORL, various network services exist which offer this information in a raw form to any application that wishes to make use of it. Some support for access to this kind of information from within higher level scripting languages is also provided.

With location information available to a mobile application, configuration becomes a dynamic process. Traditionally, an application would configure itself on startup, and may remain in this once-configured state until terminated. We see now that within the context of mobile applications, the configuration process has become a dynamic and continuous one, and support for this must be inherent within the application layer.

5 Summary

Teleporting offers us a simple and adaptable platform for experimenting with mobile applications. We see mobile applications as a complementary paradigm for the notions of mobile and nomadic computing. We need not be constrained by limitations of the computing devices that we are able to carry around, but rather can make use of those more powerful, networked computing objects that may be already to hand. In addition, ubiquitous support for mobile applications would still offer the level of personalization to our computing facilities that we expect.

References

[1] Robert W Scheifler and Jim Gettys. *X Window System.* Digital Press, Bedford, Massachussetts, 1992.

[2] Tristan Richardson, Frazer Bennett, Glenford Mapp and Andy Hopper. *Teleporting in an X Window System Environment.* IEEE Personal Communications Magasine, Third Quarter, 1994.

[3] Roy Want, Andy Hopper, Veronica Falcao and Jonathan Gibbons. *The Active Badge Location System.* ACM Transactions on Information Systems, January 1992.

[4] Andy Harter and Andy Hopper. *A Distributed Location System for the Active Office.* IEEE Network, Special Issue on Distributed Systems for Telecommunications, January 1994.

---

[3] Integrated Services Digital Network