Mobile Computing Personae

Arindam Banerji, David L. Cohn and Dinesh C. Kulkarni

Distributed Computing Research Laboratory
University of Notre Dame
Notre Dame, IN 46556
dlc@cse.nd.edu

Abstract

Highly portable, powerful computers with wireless connections will radically change the way people think about and use computing. No longer will users limit their computations to a single machine; rather, they will use the machines that best suit their current needs. A user's environment, which we call the *computing persona*, cannot continue to be redefined on each machine. Rather, as people move between systems, their environment should follow. The notion of computing persona provides new insight into resource management in a distributed and dynamic world. Resource constraints will vary widely and for-fee services will change optimization strategies. Many of the technologies needed to realize personae are current research topics, and the persona concept will focus that work and make it more valuable.

1. Introduction

The evolution of computer technology has changed the common computing platform from machines with multiple users to single-user systems; soon multiple computers per person will be the norm. The use of highly mobile systems dramatically changes the way users view computing, and new approaches are needed to support that view. This paper identifies a user's view of the computing world as his or her *computing persona* and discusses the implications on system software design and implementation of imposing this persona on multiple machines.

We can view the world of stationary computers as a *computing infrastructure* to which resource-poor mobile computers have intermittent connections. Indeed, the mobile systems will always be at a resource disadvantage compared to fixed systems of the same generation. They may have powerful processors, but problems of power consumption, input/output device size and communication capability will continue to exist. Therefore, mobile systems software must support the sharing of infrastructure resources, including such resources as screens and keyboards which have not been traditionally shared.

Two new considerations influence system software design for mobile computers. First, user movement creates a dynamic computing environment. Since shared resources are considered part of the computer, we essentially have run-time assembly of computer systems. Secondly, the resources and services offered to mobile users may frequently have user fees. Thus, system software should treat the user's money as an important and manageable resource, just like memory and CPU time.

The ideas in this paper are based on the needs of mobile computing but have broader applications. A user of multiple machines wants to have a consistent computing persona regardless of how the machines are connected. The machines may vary, for example providing different graphical user interfaces, but the view of the computing infrastructure should remain essentially the same. This is true for the user of networked machines, for the user with an intermittently connected mobile system, or for the user who just carries a diskette home at night.

2. A Scenario

Suppose a computer scientist is in her office, preparing a paper for the fifth WWOS (Workshop on Wandering Operating Systems). She opens a Motif window on her UNIX system, starts a copy of FrameMaker and types several pages. She walks down the hall to the intellectual property office and the lawyer asks for a printed copy of the unfinished manuscript. Using her PDA, she and asks that the document be printed, and a copy appears on the lawyer's laser printer.

Eventually, the computer scientist decides to leave for home and catch the bus. On the way, she displays the unfinished paper on her PDA and uses a stylus to make changes on the small handheld display. At home, she wants to see her work on paper so she again instructs the PDA to print the document. After reading it, she decides that it needs major revisions and powers up her Macintosh. It comes up with an active FrameMaker session containing the document.

Before submitting it, she wants to make sure that the lawyer is happy so she checks her mail. Finding only a notice of departmental meeting on safety (which she has already missed), she decides to submit the paper. Unfortunately, her teenage son has just gotten on the phone, so her modem connection is unavailable. She does not want to interrupt (she likes the girl he is talking to) but knows that the paper must be in Europe first thing in the morning. Therefore, she elects to let the PDA use its more costly cellular link and sends the paper off.

3. Scenario Implications

This somewhat fanciful example (no one ever likes the kids their teenage children date) illustrates three key technologies that a computing persona must provide.

- Name-space remapping - The two instances of printing caused the document to be sent to different printers. In these cases, it was the printer closest to the PDA. However, since the PDA was in a different place, it interpreted the name *printer* differently. The persona includes a *view* of the resources in the infrastructure. The mapping between the names in this view and the actual resources is dynamic and allows location of resources in the large and growing infrastructure.
- Application migration - The FrameMaker job that started on the workstation essentially moved to the PDA and then to the Macintosh. Since these systems use different hardware architectures, classic notions of process migration don't apply. Instead, the application must periodically save enough state information so that it can be restarted on a different platform. Modern word processors do this as a matter of course by saving the current state of the document. However, for other applications, it might not be easy.
- Resource-driven operation - When the computer scientist first checked her mail, she connected to her mail site was modem over a telephone line. However, like many resources, the phone line

changed state and was unavailable when she wanted to send the paper. Since the alternative resource, the cellular link, cost more money, she was given the option of delaying. In a world of dynamic resources, it is important that their availability and cost influence operations.

The persona is the user's interface to a vast array of resources which are controlled by others and constantly changing. Typically, the user refers to these resources by names that are different from those used by the resource owners. Thus, the persona includes a resource view; a complete, well-ordered name mapping to the real files, devices, resources and so forth. It is used both by the user and, perhaps more importantly, by his or her applications. This technique is used by Prospero to manage access to data files in the Internet; in a related way, Plan-9 does name-space mapping to deal with files, devices and services. However, since the persona is mobile and may be resource constrained, its remapping is more complex than either of these systems.

4. Computing Persona

The concept of computing persona is not new. Indeed, whenever anyone uses a computer, they establish a computing persona. The persona includes the a presentation of the system resources and an environment which supports that user's work. The presentation might appear as a Macintosh desktop, a set of Motif windows or a DOS session. The persona also includes the resources available to the user and the mechanisms employed to access them. To date, personae have been separately defined for each operating system, and every instance of a persona has been fixed on a single machine.

Today, if one works on multiple computers, separate personae are established. Typically, this means that one logs in to each system and deals with multiple computing worlds. Even if the machines are on a single network, they tend to look and feel different. Full use of modern computing hardware requires a computing persona to be defined universally and allowed to control multiple, heterogeneous machines.

When a user is active, his or her persona can be quite complex. It includes the state of the user interface, the active applications, any hidden services, environment variables, locally defined names and a variety of other items. To understand how to capture the persona, we have to look at how it is established.

When a machine is powered off, a user's persona is essentially a name-space and its mapping. Part of that mapping includes special files which will define the initial state of the active persona. In a UNIX system, the name-space resides in the file system; in other systems, device names may be separate from the file system. The name-space continues to exist when the user becomes active, and the management of its mapping is a key element of handling personae.

For an active user, the persona includes elements set up from the name-space and other elements that the user's actions have created. Some are passive, such as environment variables, aliases and mounts; others are active, including daemons, TSR routines and special device drivers. To capture the persona, the creation of these entities and the evolution of their state must be monitored.

The concept of computing persona is further complicated by the need to deal with heterogeneous operating systems. A mobile persona must handle this heterogeneity and adapt to the current operating system.

5. Managing the Persona

In the scenario, the computer scientist's persona was imposed on various machines. To actually do this, the user's actions must monitored regularly monitor, and the persona continually evolved. Any changes in the computing infrastructure that effect the persona must also be tracked.

Consider what happens when our computer scientist returns to her office. Her persona is re-imposed on the office system and indicates any work which had been done the night before. If her home and office machines had had a network connection and clever enough software, the home system could have made the transfer. However, even without a connection, the PDA can act as a smart cache and perform the update.

In this example, the PDA is given sole responsibility for monitoring the user's computing persona. This means it is continually powered and aware of all computing activities. A more complex, but perhaps more reasonable, approach would be to migrate the responsibility for persona monitoring between the PDA and the other computers. This would reduce power consumption problems and mitigate failure concerns. This is just a minor difference; it does not change the basic architecture.

6. Basic Architectural Approach

The creation and management of computing personae can be handled by three types of entities: resource representatives, persona managers and application agents. They are described as active entities but can probably be activatable since they tend to be quiescent much of the time.

A machine which exports access to its resources does so through a *resource representative*. The resources can be as complex as those on a full workstation including its processor, memory, file system and communication connections or as simple as a display screen. The resource representative defines a protocol for authentication, accounting and security, and it provides an interface to its database on resource state and availability. When a *resource event* occurs, that is, an event which affects a resource, the resource representative is notified; it can, in turn, notify a client using that resource.

The *persona manager* is responsible for creating and maintaining computing personae. This persona is primarily a *view* of resources, a set of active applications and a guardian of the user's monetary interests. Therefore, the persona manager must maintain the name-space that supports that view and must see to the good health of the applications. The persona manager maps each name in the space to a resource with the help of the appropriate resource representative. The manager also registers up-calls in order to be notified of resource events.

When the resources and the persona manager are on the same node, mapping is reasonably straightforward. However, for remote resources, the persona manager must contact a *name service* to locate them. Access to the name service is considered a local resource and, hence, is controlled by the local resource representative.

In general, the persona manager is mobile. For safety, portability and performance, it can be replicated on a new node with the permission of that node's resource representative. The replica must recreate the

persona state, including the name-space and application states. This can require remapping, and possibly extending, the name-space. Moving applications requires help from the application agents.

We view an active application as an entity with state and resource requirements. When an application is activated, an *application agent* is created. This agent makes sure that the application's resource needs are met and its state is periodically saved. It identifies required resources and determines whether they are available through the name-space. If they are not, it notifies the persona manager which must find and access the resources. These resources include the executable code (with appropriate DLLs), a suitable processor, necessary devices and any required files.

Application migration actually involves stopping the application on one node and restarting it in the proper state on another. The application agent moves the state to the new node, verifies that the resources are available and oversees the restart.

7. Enabling Technologies

Successful implementation of mobile computing personae will draw heavily on other operating systems research. The specific technologies include:

- Naming and Location Services - These services assist the persona manager in mapping the user's name-space to actual resources. They provide location information on remote resources and access mechanisms to the appropriate resource representatives. The domains managed by these services include file systems, communication protocols, interface repositories and libraries.
- Cache Management - Resource access is provided by the name-space, but resource use will require caching. Mobile systems can be highly dynamic, so the optimal cache management policy will change. Caching requirements vary with data type, but the basic structure is the same. Thus, a flexible cache management service could be used by many parts of the system. It should export appropriate caching interfaces which allow control by either source or sink. In a mobile computing environment, the performance of caching will be determined by the savings in both *time* and *money*.
- Dynamic linking - This is necessary to allow run-time alterations as resource availability changes for applications which access resources through library code.
- Persistent objects - Future applications built with object technology that supports persistence will provide automatic checkpointing and make migration and restart easier.
- Improved object model - An object model based on CORBA and extended to include reification and reflection can serve as the basis for building mobile computing personae. The realization of such a model would include many of the necessary tools and services mentioned above.

8. Conclusion

The combination of mobility, multiple computers and monetary considerations leads to the need for a new approach to user support in system software. A user sees the computing world through a personalized view called the computing persona. Persona management should be the responsibility of system software. It includes access to remote resources with due concern for their capability, availability and cost.