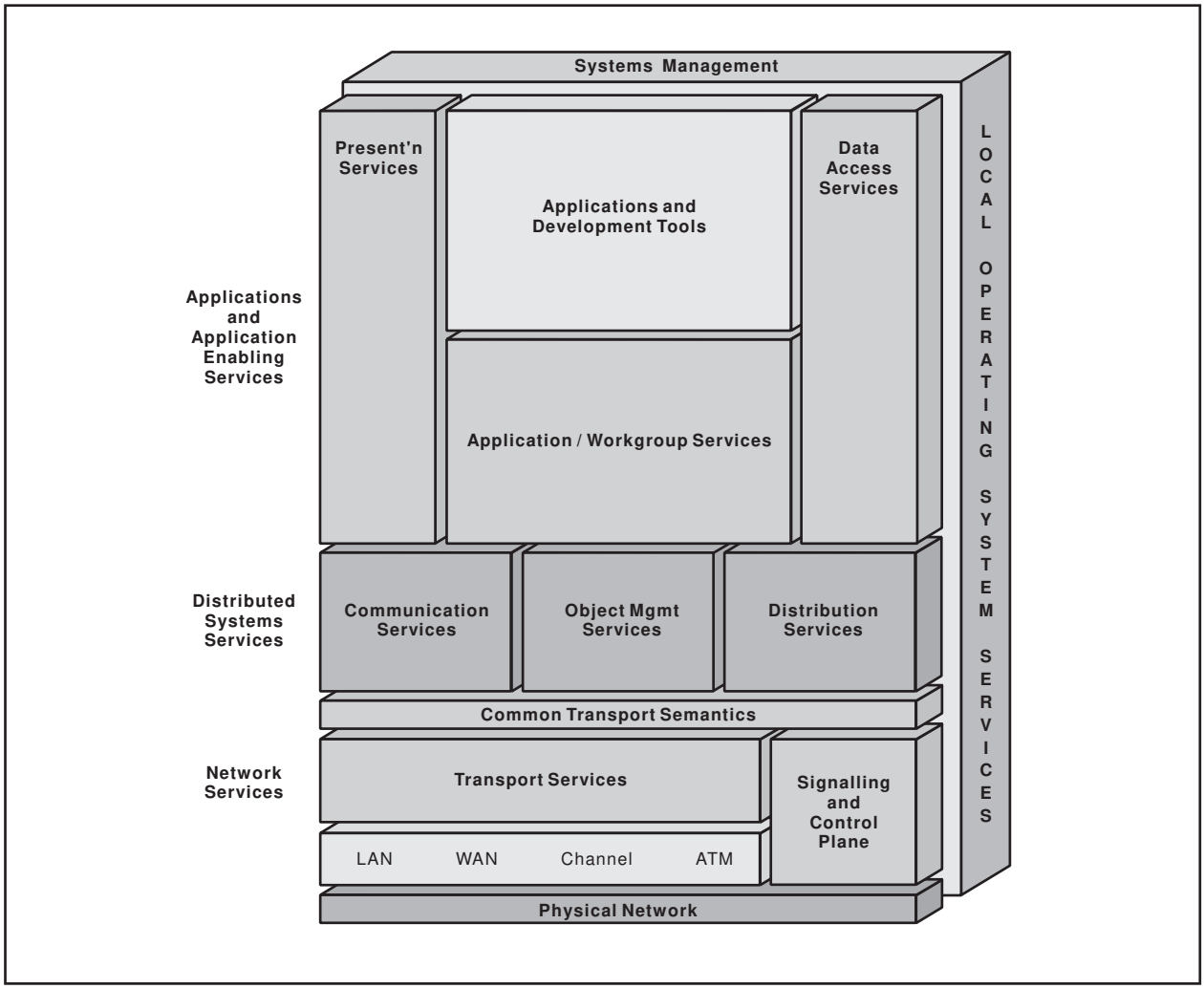
Open Blueprint

Human Computer Interaction Resource Manager



GC23-3933-01

Open Blueprint

Human Computer Interaction Resource Manager

GC23-3933-01

About This Paper

Open, distributed computing of all forms, including client/server and network computing, is the model that is driving the rapid evolution of information technology today. The Open Blueprint structure is IBM's industry-leading architectural framework for distributed computing in a multivendor, heterogeneous environment. This paper describes the Human Computer Interaction resource manager component of the Open Blueprint and its relationships with other Open Blueprint components.

The Open Blueprint structure continues to accommodate advances in technology and incorporate emerging standards and protocols as information technology needs and capabilities evolve. For example, the structure now incorporates digital library, object-oriented and mobile technologies, and support for internet-enabled applications. Thus, this document is a snapshot at a particular point in time. The Open Blueprint structure will continue to evolve as new technologies emerge.

This paper is one in a series of papers available in the *Open Blueprint Technical Reference Library* collection, SBOF-8702 (hardcopy) or SK2T-2478 (CD-ROM). The intent of this technical library is to provide detailed information about each Open Blueprint component. The authors of these papers are the developers and designers directly responsible for the components, so you might observe differences in style, scope, and format between this paper and others.

Readers who are less familiar with a particular component can refer to the referenced materials to gain basic background knowledge not included in the papers. For a general technical overview of the Open Blueprint, see the *Open Blueprint Technical Overview*, GC23-3808.

Who Should Read This Paper

This paper is intended for audiences requiring technical detail about the Human Computer Interaction Resource Manager in the Open Blueprint. These include:

- Customers who are planning technology or architecture investments
- Software vendors who are developing products to interoperate with other products that support the Open Blueprint
- Consultants and service providers who offer integration services to customers

© **Copyright International Business Machines Corporation 1995, 1996. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Summary of Changes 1

Open Blueprint Human Computer Interaction Resource Manager 3
Background 3
The Evolution of Human Computer Interaction 5
A New Desktop Architecture 5
Key Aspects of the New Metaphor 6

Notices 11
Trademarks 11

Communicating Your Comments to IBM 13

Figures

1. Relative Contribution to Overall Usability 4
2. A Desktop Architecture to Support Multiple, Specialized Metaphors 6

© Copyright IBM Corp. 1995, 1996

Summary of Changes

This revision:

- Describes the concept of places as an evolution of the desktop in greater detail
- Describes the relevance of speech as a human interface
- Describes the position of human computer-interaction in the network computing environment
- Describes the importance of content and the World Wide Web

© Copyright IBM Corp. 1995, 1996

Open Blueprint Human Computer Interaction Resource Manager

The Human Computer Interaction resource manager, with its associated technologies, supports the presentation of application and operating system information to end-users. The overall objective is to do this in an intuitive fashion, allowing the end-user to interact with the computer in a very natural, consistent manner.

The term *Human Computer Interaction (HCI)*[1] is commonly used to describe the front-of-screen appearance and function of an application or system, along with the mechanisms for the interaction between the person and the computer system. By itself, it does not imply any uniformity between applications. Standards must establish and enforce such conventions.

The term *desktop* is used to convey a complete metaphor for user interaction. The user's visual space is a desktop, and his or her tools and data are things that are on the desktop or that are readily accessible from the desktop. The desktop metaphor has been used in many operating systems throughout the last decade (for example, the Macintosh's Finder, Windows 95 Desktop, OS/2's Workplace Shell, and AIX's Common Desktop Environment (CDE)). The state of emerging natural computing technologies (pen, speech, agents, and virtual reality) makes it possible to evolve from the traditional desktop metaphor for user interaction into something much more powerful and intuitive. Key to the success of this new generation of the desktop metaphor is that it be a new, compelling metaphor, that all components with which a user interacts support the notion of customizable, specialized user environments, and the focus is always on end-user value.

This document describes the components of the Human Computer Interaction resource manager that support the new generation of human-computer interaction.

Background

The user environment can be described in terms of three sets of aspects: visual appearance, interaction techniques, and the user's object model. The visual aspects include both operational visuals, providing cues and feedback to assist the user in the operation of the interface, and aesthetics, such as the use of color, line widths, textures, and typography. The interaction technique aspects include how functions are assigned to mouse buttons and keyboard keys, and the sequences of button and key presses required to accomplish a particular action.

The introduction of multimodal interaction techniques (allowing the use of multiple interaction models such as speech and mouse in conjunction) will continue to simplify the interactions required to obtain a desired action. Collectively, the visual and interaction technique aspects comprise the syntactic elements of the interface. They constitute what has long been termed the *look-and-feel* of the interface. They have also been the focus of most of the user-interface design work during the last decade. The separation of look-and-feel from aspects of application function has been one of the fundamental tenets of user-interface design philosophies. Figure 1 on page 4 illustrates these aspects of user-interface technology.

© Copyright IBM Corp. 1995, 1996

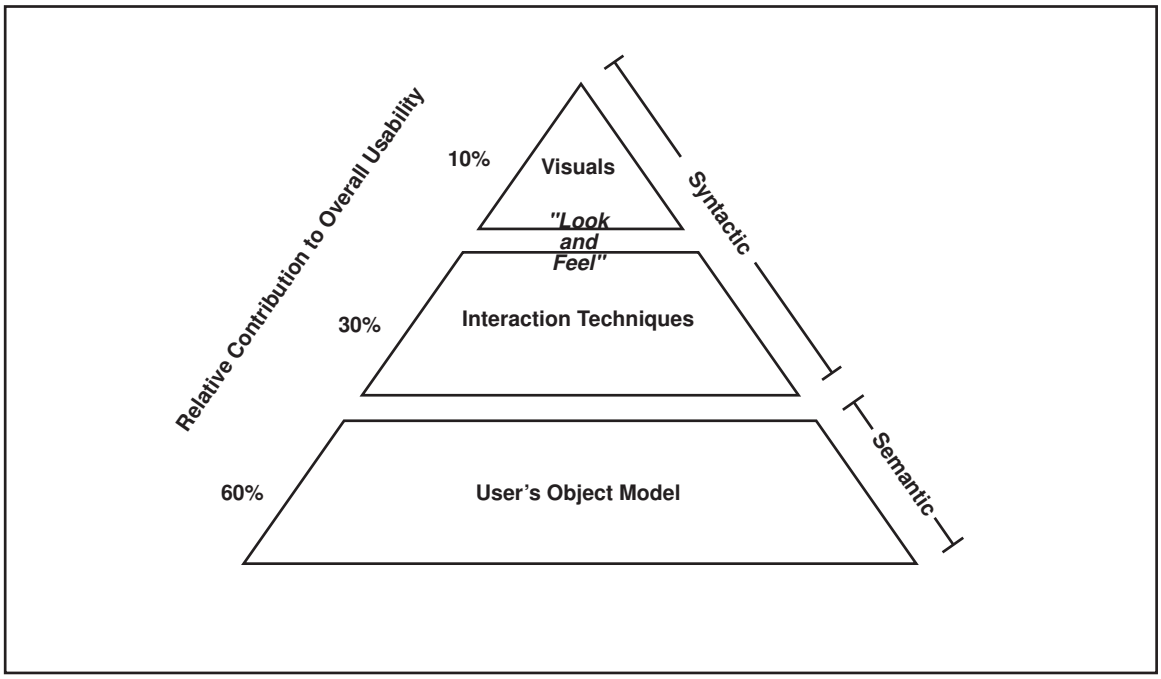


Figure 1. Relative Contribution to Overall Usability

In recent years however, the focus has begun shifting to the role of the objects manipulated by users in accomplishing their tasks. The aspects associated with utilized objects, the behaviors of those objects, their properties, and interrelationships between them are collectively addressed in the user's object model. For the Workplace Shell in OS/2, the user's object model includes objects such as folders, drives, printers, and data, as well as the concepts of templates, for creating new objects, and shadows that allow multiple appearances of the same object in different places simultaneously. The user's object model represents the semantic-level information a user must understand in order to accomplish work.

When a user uses a computer to perform a task, such as preparing a monthly activity report, the user must first determine what facilities are provided by the computer and which are appropriate to performing the task at hand. In other words, users must understand their object model, at least to a level compatible with their own task goals. Users must map their task goals into the objects provided, and finally execute their tasks using the look-and-feel aspects of the interface.

As a result of the more recent focus on the user's object model, a closer relationship between desktop architecture and the design of the interface between the human and the computer has been recognized. Traditionally, the look-and-feel aspects of an interface have been implemented in user interface toolkits and style guides, for use by developers. The aspects embodied in the user's object model are implemented in the desktop and the objects that populate that environment. One estimate of the relative value of these aspects attributes 10% of overall usability to the visual aspects (look), 30% to the interaction techniques (feel), and the remaining 60% to the user's object model. In other words, the user environment established by the desktop and the objects that populate it are the major contributor to overall usability. This is why the Human Computer Interaction resource manager is based on an object model built on top of the local windowing system to provide underlying support for the visual aspects and the interaction techniques. To enforce consistency among applications using a similar object model, user-interface standards like Common User Access (CUA) for OS/2, *Windows 95 User Interface (UI) Guidelines*, and CDE on AIX, have been adopted to ensure common look-and-feel (visual appearance and interaction techniques).

The Evolution of Human Computer Interaction

The need exists for specialized user experiences focused on environments such as the office, factories, retail establishments, home, and while travelling. The *messy desk* metaphor is being complemented by a more task-based solution. A user environment is required that is highly modular and configurable, built from components that can be combined in different ways quickly and without requiring expensive highly-skilled programmers. An environment in which domain experts can augment or even implement the business logic associated with their tasks is needed.

The user environment has evolved considerably since the introduction of the personal computer (PC) in 1981; from command line interfaces, to point-and-click program and file managers, to a fully object-oriented, graphical user interface. At the same time there is a growing trend toward major consumer acceptance of personal computers. Computer and software firms are scrambling to make PCs less intimidating, resulting in a variety of new, simplified user environments.[2]

The desktop is evolving into a compound document, where there is no boundary between the desktop and other environments (for example, the World Wide Web (WWW)). In some environments, Web browsers will cease to exist as a separate entity and will converge with component[3] software to provide the new generation of human-computer interaction.

Similarly, the user will not perceive any difference in the location of the content (data): a mixture of desktop, file system, WWW, collaboration documents, relational database, and so on. The user will interact with content in the same manner, regardless of origin.

In addition, users can experience a variety of views on each object. These can appear as icons, various content lists, and more specialized content-specific views. Using multiple views, users can display contents in a variety of formats according to their taste and needs.

User interfaces are being substantially improved through the adoption of more natural user-interaction techniques, coupled with compound document technology. Compound-document technology enables a new user model that allows a display to be constructed from numerous small interacting but independently constructed components. This will provide for far greater flexibility and control over the construction of a desired composition of visual elements than today's monolithic applications.

A New Desktop Architecture

Ideally, a desktop serves as a base environment into which user objects can be plugged (see Figure 2 on page 6). Similar to the way in which a hardware system board provides infrastructure for plug-in adapter cards, the desktop subcomponents provide software infrastructure for drag-and-drop, embedding and linking, dynamic data exchange, installation and configuration, help, and many other functions shared by applications. The Human Computer Interaction resource manager sits on top of an enabling technology base including the Compound Document resource manager and the Object Request Broker resource manager, which supports the System Object Model (SOM).

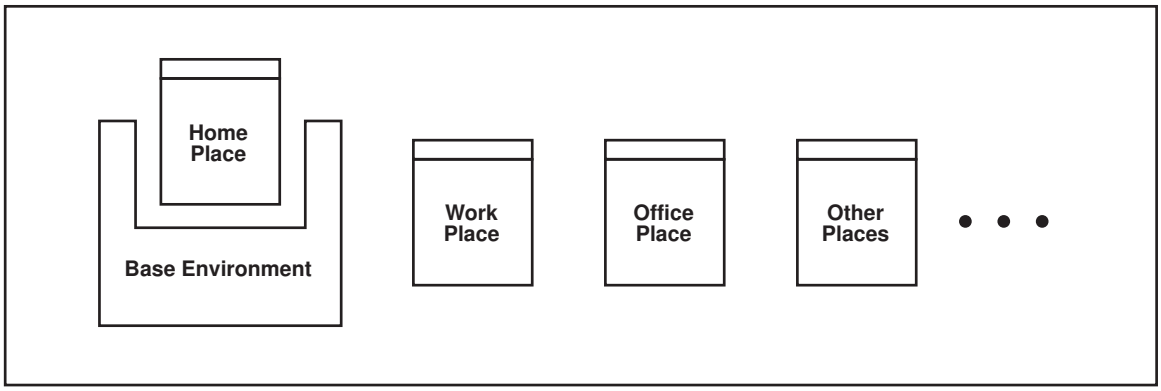


Figure 2. A Desktop Architecture to Support Multiple, Specialized Metaphors

This provides end-user value "out-of-the-box" in the form of content-rich, task-oriented environments in which users can be instantly productive, as well as the ability to create specialized environments for specific market and end-user requirements.

The prospect of multiple specializations of a common desktop metaphor (for example, specialized metaphors for the home, executive, office, manufacturing, and other environments; as shown in Figure 2) offers an opportunity for developers. Development environments that offer high-productivity for component development, modular construction, and user-level customization using scripting languages will provide a significant advantage. The Human Computer Interaction resource manager, using object and compound-document technologies, provides these advantages.

Key Aspects of the New Metaphor

In the Human Computer Interaction resource manager, specialized desktop environments are based on *places*, which supersede the current, single-desktop metaphor. Places are task oriented. A place contains the resources (including other places) necessary to accomplish a specific task or set of tasks, providing a context in which the user works. A place is distinguished by its content, visual appearance, and user-interaction paradigms. Existing applications will be supported in all places. Places will be compound-document containers and will be able to contain things that are components. Places and the things in them incorporate the latest in natural computing technologies and provide for a rich variety of content.

A component can be shared between different places. And since places are distinguished by their visual appearance and user interface interaction techniques, it is imperative that the proper separation be made between a component's view and its model (or state). System-provided components have such separation.

Places support sharing and collaboration between users. Multiple users can occupy (be "in") a place at the same time. Things in a place can be shared, and users can determine who is in the place and communicate with them.

There are various ways in which places can represent themselves to a user. These can range from abstract representations similar to today's desktops (using familiar constructs such as windows) to photo-realistic places that minimize the use of computer-oriented artifacts. Places can be provided for specific activities in the home, such as a library, a place for playing games, and a variety of industry-segment-specific places. Places will be provided by IBM, by independent software vendors, and by users themselves.

The people and things in places, as well as the places themselves, are developed using facilities of the Compound Document resource manager, which provides linking, embedding, and scripting functions to places and their contents. Thus, places are themselves components that can be combined in a variety of ways. A place can be shared by multiple users or accessed from multiple other places. Places can be manipulated using a simple scripting language like LotusScript. This allows application developers and users to customize behavior and incorporate business logic into places and documents. For maximum usefulness and productivity, visual builders are provided in support of the scripting capability. Thus, users can customize their environment and even create new applications by dragging and dropping prebuilt components and connecting them with logic written in a simple scripting language.

Functional Description

The Human Computer Interaction resource manager encompasses many technologies, each of which contributes to its value. The power of each of these technologies is harnessed through its application programming interfaces (APIs) using development tools. While each technology has its own associated frameworks and APIs, there are several unifying conventions that are key to the Human Computer Interaction resource manager:

- IBM OpenClass embodies an abstract component model that allows a component developer to build components that may be targeted to specific component technology, including:
  - OpenDoc's compound-document technology, including linking and embedding, which enables parts to be interchangeable, sharable, and portable across multiple environments.
  - Object Linking and Embedding (OLE) and Component Object Model (COM), which are of particular relevance to Windows platforms.
- SOM is the underlying, low-level object model; providing the capabilities of binary release-to-release compatibility, multiple language support (in the form of language bindings for APIs), and OMG CORBA-compliance.
- Platform-specific windowing and graphics systems provides underlying support for the user-interface elements: frames, controls, and graphics (for example, Window's Win API and GDI, OS/2's Presentation Manager and GPI, and X-Windows and Xlib on AIX.)

APIs

The Human Computer Interaction resource manager supports a mix of procedural and object-oriented APIs. Rather than replace all existing procedural APIs at once with object-oriented counterparts, object-oriented alternatives are provided. The current procedural API will be maintained, although much of the new function will be provided through C++ classes and components.

The API sets supported within the Human Computer Interaction resource manager are those that provide access to the following:

- **IBM OpenClass**. The IBM Open Class libraries implement a wide range of items from threads to user-interface elements. These libraries are implemented across platforms for use by developers. OpenClass is the vehicle for delivery of object-oriented abstractions of both the traditional services delivered through procedural APIs, and the newer services to be delivered as frameworks for which there may be no corresponding procedural API.

  All the elements required to implement the user interface of an application are supplied by OpenClass in an abstract, platform-neutral set of classes:
  - An abstract component model, which eases the development of compound document components
  - Model and view separation with an associated synchronization mechanism
  - Multimodal interaction management, which maps user requests into explicit actions

  - Windowing, controls, and graphics
  - 2D and 3D graphics

- **Speech recognition**. These APIs are used to provide speech input to applications. Speech input can be used to invoke actions (commands) or to create text in a document or form (dictation). Currently, speech recognition is not ubiquitously available, as are keyboard and mouse, in all operating system platforms. Therefore, the API set used depends largely on the speech recognition vendor chosen. With the release of OS/2 Warp in mid-1996, OS/2 includes speech recognition through the Speech Manager API (SMAPI). SMAPI provides both commands and dictation and is also supported by IBM on Windows and AIX platforms. Microsoft is in the process of defining a set of vendor-independent speech APIs for Windows platforms, called SAPI. SAPI is well-defined for commands and still emerging for dictation. Novell has also initiated a vendor-independent effort called SRAPI. While SRAPI is defined to also be platform-independent, the current implementations are for Windows.
- **Windowing, controls, and graphics**. These APIs are used to draw elements on the screen. Each platform has its own framework or set of APIs for this (PM and GPI on OS/2; Motif, X-Windows and XLib on AIX; Win32 on Microsoft Windows). These APIs are used by other higher-level frameworks, and by individual resource managers (for example, compound document).
- **OpenGL**. OpenGL is a software interface for applications to generate interactive 2D and 3D computer graphics. OpenGL is designed to be independent of operating system, window system, and hardware operations, and it is supported by many vendors.
- **Platform-specific desktop APIs**. The OS/2 Workplace Shell and CDE on AIX provide APIs that implement similar traditional desktop object models.
- **Places and Desktop Components**. These object classes support the development of places, specialized to a particular task-oriented environment. As stated earlier, places and components on the desktop are based on compound document technology, the base behavior of which is implemented in the compound document frameworks. However, places and desktop components have behavior beyond that required by a compound document. This behavior is implemented by frameworks within the Human Computer Interaction resource manager.

Support for previously existing applications is a key objective of the Human Computer Interaction resource manager. Migration paths exist from existing APIs to new ones.

Network Computing Implications

Ubiquitous access to the WWW and other Internet (and intranet) services are contributing to the perception that the distinctions in resource location (on the network or a user's local machine resources) are gradually disappearing.

The Human Computer Interaction resource manager plays several roles in network computing. These roles can be explained as follows:

- In its traditional role, the desktop's primary environment is as a client machine, making use of services provided by many distributed servers. As a client machine, the user perceives no differentiation between viewing and navigating data locally or remotely.
- Places can be distributed, and the occupants of a place at any given time can actually represent users in different locations.
- As places enable multiple users to collaborate, there are also peer-to-peer relationships between collaborating users' individual desktops. For example, a user on one machine can appear in a common place that resides on another user's desktop; both working on the same document.
- Additionally, the desktop can itself reside on a server. For example, OS/2's remote-IPL (RIPL) feature allows major elements of the desktop to reside on a server and be shared by multiple client

workstations. This configuration is useful in an operating environment that needs to be closely administered.

Regardless of the particular relationship between desktops, and between a place and its contained components, the services provided by the Human Computer Interaction resource manager are consistent with Open Blueprint Distributed Systems Services' goal of presenting the user with a single, coherent computing facility instead of a complex, distributed system. The user's impression is that information, data, applications, components, and places on remote systems are merely extensions of their own desktop.

Relationships to Other Open Blueprint Resource Managers

Of the elements provided by the Distributed Systems Services resource managers, the Human Computer Interaction resource manager has the most dependencies on Object Management Services.

The Human Computer Interaction resource manager is related to other services provided by the Application Enabling Services resource managers (Application Services and Data Access Services) to the extent that the resource manager services (in the form of a telephony API, mail API, and so on) are exploited by components written as the content for places (for example, telephone and fax machine components).

The Compound Document resource manager provides linking, embedding, scriptability, and composition, that enable the Human Computer Interaction resource manager's notion of highly customizable, collaborative task-oriented environments (places), and sharable, reusable, configurable components.

Summary

Desktop environments are evolving to accommodate specific industry segments and user needs. This is leading to a proliferation of new user-interface metaphors. The days of the messy desk metaphor as a single, generally acceptable, user environment are coming to a close.

Desktop environments have become much more modular and configurable. The Human Computer Interaction resource manager takes advantage of key component technologies to establish a base for developing specialized places intended to support the user in the execution of a task.

The emergence of a new generation of component-based software has produced programming environments for building components, basic building-block components for building component assemblies, and component-based products to address specific enterprise-level requirements. These components can be combined in a variety of ways with other existing components and with those yet to be developed. Thus, a high degree of productivity for developers, application assemblers, and users is enabled through the use of scripting languages and visual builders.

The Human Computer Interaction resource manager combines the use of multiple technologies and the use of other Open Blueprint facilities to support modern human-computer interaction.

---

[1] The term Computer Human Interaction (CHI) is common, but the bias implied by Computer to Human, rather than the converse is undesirable. The term User-Interface (UI) is also common.

[2] Including Packard Bell's Navigator, PDA interfaces such as General Magic's MagicCap, and World Wide Web products such as Netscape Navigator.

[3] This use of the term *component* relates to objects that the user recognizes and manipulates, such as printers, folders, or mailboxes, and parts found in a compound document such as a graph, photograph, equation, or text.

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
CUA
IBM
IBMLink
Open Blueprint
Open Class
OS/2
Presentation Manager
SOM
System Object Model
Workplace Shell

The following terms are trademarks of other companies:

Bell	American Telephone and Telegraph Company, Incorporated
C++	American Telephone and Telegraph Company, Incorporated
CORBA	Object Management Group, Incorporated
Finder	Apple Computer, Incorporated
Lotus	Lotus Development Corporation
Macintosh	Apple Computer, Incorporated
Motif	Open Software Foundation, Incorporated
Novell	Novell, Incorporated
OpenDoc	Apple Computer, Incorporated
OpenGL	Silicon Graphics, Incorporated
RIPL	CTA Incorporated
Win32	Microsoft Corporation
X-Windows	Massachusetts Institute of Technology

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

© Copyright IBM Corp. 1995, 1996

Communicating Your Comments to IBM

If you especially like or dislike anything about this paper, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply. Feel free to comment on specific error or omissions, accuracy, organization, subject matter, or completeness of this paper.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

- If you prefer to send comments by FAX, use this number:

  United States and Canada: 1-800-227-5088.

- If you prefer to send comments electronically, use one of these ID's:
  - Internet: **USIB2HPD@VNET.IBM.COM**
  - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
  - IBMLink: **CIBMORCF at RALVM13**

Make sure to include the following in your note:

- Title of this paper
- Page number or topic to which your comment applies

© Copyright IBM Corp. 1995, 1996

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC23-3933-01