Department of Electrical Engineering
and Computer Sciences

University of California

Berkeley, California 94720

The future of computer-telecommunications integration

by

David G. Messerschmitt
University of California at Berkeley

Invited paper in *IEEE Communications Magazine*,
special issue on “Computer-Telephony Integration”,
April 1996.

This issue of IEEE Communications Magazine describes activities in integrating the desktop computer with the telephone system (computer-telephony integration, or CTI), giving users easier access to telephone feature sets leveraging the computer’s graphical user interface, or allowing a telephone call to be incorporated into a larger computer application. In the preface to this issue, Vint Cerf asserts that CTI is but one step in a “revolution” that is taking place in the way computers are used in conjunction with telephony. Agreeing strongly with this basic thesis, in this afterword we speculate on the specific form that the computing and telecommunications infrastructure will take in the future. We hypothesize that CTI is but one step in the evolution to a seamless and interoperable integrated telecommunications and computer infrastructure, and this may arrive surprisingly soon as enabled by some recent technological developments. In [1] we give a more detailed roadmap to the convergence of telecommunications and computing, as well as describe many important research issues, and in [2] we describe some of the societal trends and problems that follow from these technological advances.

What is telecommunications and what is computing?

Telecommunications has been associated with audio and video media, and computing with data media. As all these media become integrated in both the network (in ATM, Internet, etc.) and the desktop computer (multimedia), this historical terminology is no longer as meaningful. Applications are becoming blurred as well. Accessing bank records using a DTMF telephone and voice response unit, or with a networked computer, differ as to medium but not functionality. In light of this, it is appropriate to define a more transparent classification of networked applications that is medium-blind.

Define an *application* as a collection of functionality of value to a *user* (a person). Here we are concerned with *distributed* or *networked* applications. A *service* is defined as functionality that is generic, or common to many applications. Examples of services would be audio or video transport, file-system management, printing, electronic payment mechanisms, encryption and key distribution, and reliable data delivery. A taxonomy of networked applications as shown in Table 1. We separate networked applications into two categories with respect to their temporal

Table 1. A taxonomy of applications with examples.

	Immediate	**Deferred**
Client-server	Video on demand WWW browsing	File transfer
Peer-to-peer	Telephony Video conferencing	Electronic mail Voicemail

characteristics:

- *Immediate*, meaning a user is interacting with a server or another user in real-time, with latency or delay requirements.
- *Deferred*, meaning a user is interacting with another user or a server in a manner that implies no fixed temporal relationship and for which the delay is not critical.

The second dimension divides applications according to functionality:

- *Peer-to-peer* applications, in which two or more users each interact with *peer computers* or *terminals*, which in turn communicate over a network for the purpose of providing some useful shared functionality.
- *Client-server* applications, in which a user interacts with a *client computer* or *terminal*, which in turn communicates over the network with a *server computer* that exists for the purpose of providing functionality or data management to the remote user.

Different components of a single application can fall into distinct categories. For example, in voice mail, the originating user forwards the voice message to a voicemail server, from which it is later accessed by the destination user. To the two users, the application is peer-to-peer and deferred (as listed in the table). However, the interaction of each user with the voicemail server is client-server and immediate.

Immediate peer-to-peer applications are usually associated with telecommunications, and client-server with computing. But there are many exceptions; for example, the touch-tone telephone and voice response unit have resulted in a flurry of telephony-based immediate client-server applications. Desktop computers are increasingly used for video conferencing, an immediate peer-to-peer application.

Where we are today

Two trends are striking today. First, the emergence of the desktop computer as a communications tool (in addition to its traditional data manipulation and management role), and second the integration of different media (audio, data, video, graphics, etc.) both on the desktop and in the network. The former flows naturally from the networking of computers (especially the Internet) and the latter flows from advances in electronics technology that give the computer the processing speed necessary for audio and video and also enable flexible packet switching in the network.

Programmable software definition of both services and applications is increasingly prevalent. Associated with the declining cost of processors and memory, intelligence is increasingly being added to terminals at the edges of the network, supplementing the centralized control inside the network. CTI can be viewed in this light as building sophisticated telephony feature sets, and integrating telephony into other applications, leveraging the computer already sitting on most desktops.

But this is surely only the beginning. We can identify a few key trends that will profoundly influence both telecommunications and computing in the future. These trends are driven by powerful technological and economic forces.

Horizontal integration

Contrast two architectures for provisioning distributed or networked applications shown in Figure 1: In *vertical integration*, a provider provisions a turn-key application using a dedicated

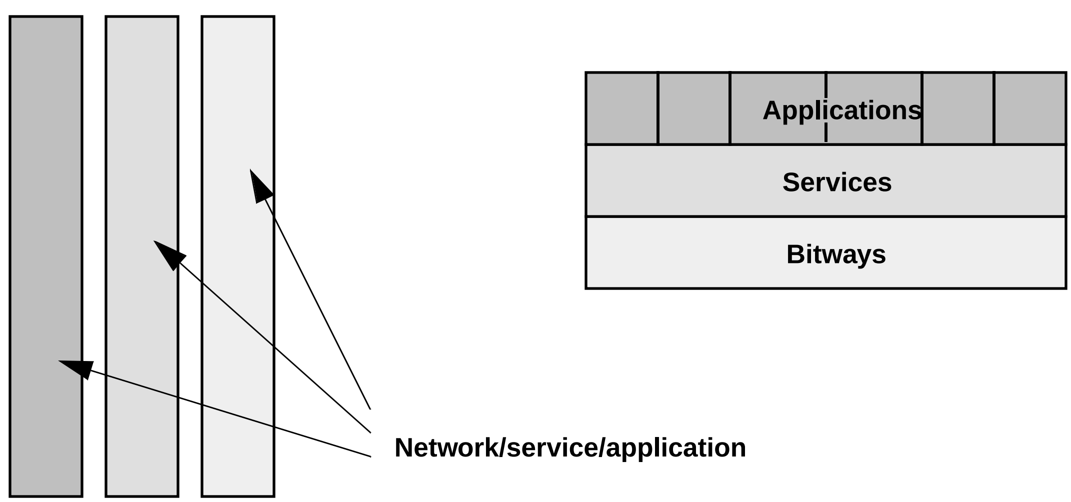

Figure 1. **Two architectural models for provisioning networked applications: vertical and horizontal integration.**

infrastructure; example applications include “telephone”, “cable-television”, or “video-on-demand”.

In contrast, *horizontal integration* is characterized by the following [3]:

- One or more integrated bitways that transport integrated data and stream media like audio and video with configurable quality-of-service (QoS) parameters (bitrate, delay, and reliability).
- A set of services, such as middleware services (directory, electronic funds transfer, privacy key management, etc.) and media services (audio, video, etc.) that are made universally available to applications.
- A diverse set of applications made available to the user.

A key feature is the integration of different media within each application, as well as within the bitways. It should be emphasized that this is a *logical* model; we deal with some implementation issues shortly.

We hypothesize that powerful economic and technological forces are driving us toward horizontal integration. Advances in technology have already resulted in the integration of different media in both the network (such as ATM or the Internet) and in the terminals (such as desktop computers). This level of horizontal integration offers the service provider substantial administrative benefits, relative to the alternatives of separate or overlay networks, and adds value to the user, since different media can easily be incorporated into *multimedia* applications.

The separation of the applications from bitways and services best serves the user by encouraging a diversity of applications, including many defined for specialized as well as widely popular purposes. Vertical integration discourages this diversity because a dedicated infrastructure demands a large market, and because users don’t want to deal with multiple providers. Horizontal integration lowers the barriers to entry for application developers since most of the infrastructure (bitways and services and even programmable terminals) are already available. Applications can be defined in software and coexist in the same programmable terminals with other applications, reducing the cost of distribution and the incremental cost of a new application. Finally, it is unlikely that a single company can accumulate the range of expertise required to provide the best solutions across such a wide range of media and technologies.

The same inherent value of application diversity does not apply to bitways and services. They are generic and widely applicable to different applications, difficult to differentiate except in terms of cost and performance, and are capital intensive and benefit from economies of scale.

The computer industry is far along in the evolution to horizontal integration. The desktop computer freed the user of the constraints from the computer center bureaucracy and lowered the barriers to entry of application developers, which in turn offered greater value to the user. Our speculation is that the telecommunications industry will be pushed by market forces in the same direction, even though many companies would doubtless prefer vertical integration and proprietary solutions.

The open interface

An important feature of horizontal integration is the *open interface*, which enforces modularity and thus allows a diversity of implementations and approaches to coexist and evolve on both sides of the interface [3][4]. For example, CTI interfaces such as TAPI separate the telephony infrastructure from higher level control features incorporated into a diversity of desktop computer applications. In the computer industry, the two most important open interfaces are the internet protocol (IP) between bitway and services, and the operating system application-programming interface (API) between services and applications. The success of the Internet follows in part from the low barriers to entry to developers, who require no modification to the OS services or IP bitway to develop and deploy new distributed applications. On the bitway side of the IP interface, Internet service providers are able to deploy new technologies like ATM without affecting OS services or applications (except of course through the one common denominator of QoS).

Open interfaces offer vendors a large and immediate market for new applications. The resulting diversity of applications increases the utility of the open interface to the user. This positive reinforcement leads eventually to a dominant open interface, to be displaced only by a new interface that offers significant functional or performance advantages. The key question for the future is where these horizontal interfaces should fit in the hierarchy of functionality. (For example, we mention later a new “virtual machine” layer that is just now emerging.) Another question is how we avoid a proliferation of multiple interfaces that have not only different syntactical structure (a minor problem), but also present different semantic models of the underlying functionality. (For example, can we define parameterized QoS models that fit universally across radically different transport media like congestion-dominated backbone networks and interference-dominated wireless access links?)

The distribution problem: network deployment

One puzzling observation is that peer-to-peer applications are relatively small in number: telephony and video conferencing, and functionally similar deferred counterparts, voice mail and electronic mail. There has been considerable research in collaborative applications, such as shared editing of a document, a whiteboard, collaborative design tools, etc., but there has been little commercial activity in these applications or in adding collaborative features to standard desktop applications like word processors and spreadsheets. Why is this? One possibility is that *compelling* peer-to-peer applications are few in number. Another possibility is that this class of applications has been overlooked by the application software industry. Yet another is that the human factor aspects are not sufficiently developed.

In our view, none of these reasons is as important as a fundamental obstacle to the commercial exploitation of peer-to-peer applications that economists call “network externality”. A peer-to-peer application offers the user a value that grows with the number of other users that have an interoperable application available. Early adopters derive very little value. (Who is the first user to buy a video conferencing application if there are no other users with whom to conference?)

Client-server applications do not have this obstacle: once a server is made available, the first user derives the same value as later users.

Network externality is essentially a distribution problem. If a peer-to-peer application can be distributed to a large number of users virtually simultaneously, interoperability and a community of available users are guaranteed, even for early adopters. For software-defined applications, this is technically feasible, since an application can be distributed over the network itself. In the Internet, developers of client-server applications like World-Wide Web browsers, document viewers, and audio and video players, are distributing new versions of those applications over the network. By bypassing traditional slow distribution channels, the velocity of innovation has been increased dramatically.

The network distribution of applications has the potential to make a much bigger impact on peer-to-peer applications than client-server ones, since getting those applications to many users simultaneously is the key to commercial viability. Nevertheless, the current approach in the Internet, in which the user has to anticipate the need for an application and execute the relatively sophisticated and manual "network file transfer", remains a barrier. Other obstacles are multiple microprocessor instruction sets and operating systems, and security problems associated with downloading binary executables from untrusted sources. Recently, a technical advance with great promise has appeared that addresses these problems, associated with a new horizontal open interface called the *virtual machine*.

The virtual machine: dynamic deployment

The virtual machine is illustrated in Figure 2. A layer of software is inserted between the operating system and the application that separates the application from the specifics of the operating system and hardware platform. The virtual machine open interface defines a general instruction set, as well as API's to resources like network services, all in an OS-independent way. An application can be written in a high-level language that is compiled into the virtual machine instructions, and then distributed over the network to be executed in terminals with compliant virtual machine implementations. Thus far, this approach is embodied in three application-description languages: Safe-Tcl [5], Telescript [6], and Java [7].

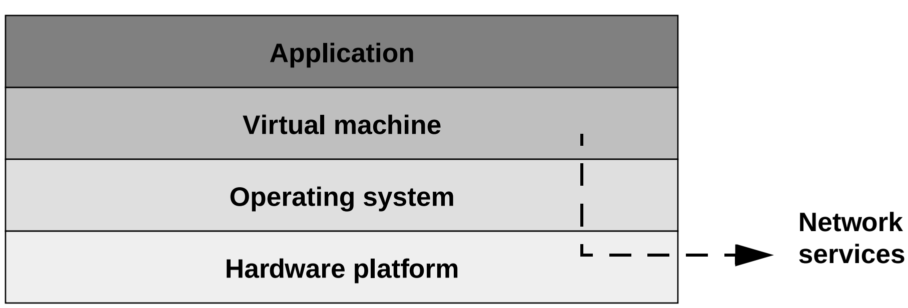

Figure 2. **The virtual machine open horizontal interface, which separates the application from the details of the OS.**

The virtual machine interface facilitates the *dynamic network deployment* of applications. That is, a distributed application can be copied over the network as a part of its establishment phase, transparently and invisibly to the user, with guaranteed interoperability. Thus far, dynamic deployment has been applied primarily to client-server applications, such as adding functionality to a WWW browser. It should have a much greater impact on peer-to-peer applications, since it bypasses the obstacle of network externality. Peer-to-peer applications interoperable over the network can be established, without prior standardization or even the need for users to obtain the requisite software in advance, to a community of interest consisting of all networked implementations of the virtual machine. We previously demonstrated this using Tcl as the application-description language [8].

Dynamic deployment benefits from (and may even require) broadband networking, since application executables will often be large. This will be an important driver for broadband access to the network, just as low-latency downloading of executables is a primary driver for broadband local-area networks.

A dynamic and flexible environment for applications

The widespread deployment of virtual machine interface software will offer both client-server and peer-to-peer application developers a lower barrier to entry and a larger market for their applications. A more important consequence will be a dramatically increased activity in peer-to-peer applications, which may become as dynamic and innovative as the client-server market.

Our final speculation, therefore, is that an infrastructure consisting of networked terminals incorporating virtual machine interfaces, plus a horizontally integrated bitway and services infrastructure, will offer a rich and dynamic environment for both peer-to-peer and client-server networked applications in the future. Because various media -- data, audio, video, etc. -- will be horizontally integrated throughout this infrastructure, the old intellectual vestiges of telecommunications and computing will have completely disappeared. The industry will likely be organized into a relatively small number of bitway and services providers, and a large number of applications vendors offering their wares for instantaneous dynamic deployment to the terminals. Standardization will continue to be important in bitways and services, but not the applications.

Acknowledgments

The author is indebted to the following colleagues who provided valuable comments on early drafts of this paper: G. David Forney, Levent Gun, David Leeper, and John Major of Motorola, Stewart Personick of Bell Communications Research, Bob Rosin or ESPI, Edward Lazowska of the University of Washington, John Godfrey of the National Research Council Computer Science and Telecommunications Board, Carl Strathmeyer of Dialogic, and Hal Varian and Wan-teh Chang of the University of California at Berkeley.

References

1. D.G. Messerschmitt, "The convergence of telecommunications and computing: What are the implications today?", submitted to *Proceedings of the IEEE*, Feb. 1996.
2. D.G. Messerschmitt, "Convergence of telecommunications with computing", invited paper in the special issue on "Impact of Information Technology", *Technology in Society*, to appear in 1996.
3. National Research Council, Computer Science and Telecommunications Board, *Realizing the Information Future; The Internet and Beyond*. Washington D.C.: National Academies Press, 1994.
4. P. Haskell and D. Messerschmitt, "In favor of an enhanced network interface for multimedia services," submitted to *IEEE Multimedia Magazine*.
5. N. S. Borenstein, "E-mail with a mind of its own: The Safe-Tcl language for enabled mail," *ULPAA 94 Conference*, Barcelona, Spain, 1-3 June 1994.
6. J. Tardo and L. Valente, "Mobile agent security and Telescript," *IEEE CompCon*, 1996.
7. J. Gosling and H. McGilton, "The Java(tm) Language Environment", an unpublished white paper (http://java.sun.com/whitePaper/java-whitepaper-1.html).
8. W-T Chang, W. Li, D.G. Messerschmitt, and N. Zhang, "Rapid Deployment of CPE-Based Telecommunications Services", *Proc. Global Communications Conference*, San Francisco, Dec. 1994.