The Prospects for Computing-Communications Convergence

David G. Messerschmitt
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley

This is an invited paper at the MÜNCHNER KREIS, Conference "VISION 21: Perspectives for the Information and Communication Technology", Munich, Germany, Nov. 25, 1999.

Abstract

Driven by the Internet and intelligent terminals, the convergence of computing and communications has accelerated. Due to advancing technologies, ample bandwidth, storage, and processing can be presumed wherever we want them, and at a reasonable cost. How do we make best use of these three complementary technologies? We discuss this from the perspective of applications, the supporting industry structure, and standardization. A new era of applications will integrate three longstanding traditions—the algorithmic tradition, the document tradition, and the collaboration tradition—greatly impacting end-users, organizations, and society. The organization of the vendor industry will be radically affected by this convergence, and the role of standardization will be changed irrevocably. The role of the service provider will expand, offering an opportunity for telecommunications service providers.

1 Introduction

Convergence occurs when firms and industries that were once independent become competitive, complementary (mutually dependent), or both. Often convergence is associated with industrial reorganization (such as mergers and divestitures) as firms adapt to the changing realities.

One prominent form of convergence in recent years involves the computer and communications industries [Mes96], driven initially by the networking of computers by the Internet and by increased programmability of telecommunications terminals (which include desktop computers). This leads to an explosion in networked applications [Mes99a,b], and irrevocably adds communications as a major new application arena for computing. This paper takes a forward-looking view of this convergence, predicting industry, standardization, and application implications for the future. What can be accomplished with computing and communications together that cannot be accomplished by these elements alone, and what implications does this have for end users and industry?

2 The Technology

Before dealing with its implications, it is helpful to appreciate some important features of the underlying technologies. Convergence benefits from three key technologies (communications, processing, and storage) and one basic commodity (information). Information has been defined as "recognizable patterns that inform us or affect our actions" [Mes96b], and comes by way of different *media*, such as the printed word, pictures, audio, and video. The technological roles are distinct: *processing* modifies information, *communications* conveys it over distance, and *storage* conveys it from one time to another.

A fundamental enabler of convergence is the representation of all information (including in all media) as binary data (collections of bits). The result is that different information media can be freely mixed within applications, and processed, stored, and communicated within a common infrastructure. The core measures of performance for the infrastructure are processing cycles per unit time, communication bandwidth (bits per unit time) and storage density (bits per unit area of the storage medium). These performance measures per unit cost have all advanced geometrically over two to three decades, doubling every 1.5 to 2 years. (This is true of fiber optics for communications, but wireless communication has advanced more slowly.) As fundamental physical limits lie somewhat in the future, these advances are expected to continue at least for one to two decades.

Another performance parameter is the speed of light, which determines the propagation velocity and hence the delay in communicating information across distance. This one parameter is not changing, and thus becomes the limiting factor as all other performance metrics improve. As a result, in the future the primary determinant of peak system performance is physical size (or geographic extent). The architectural and design techniques needed to overcome performance limitations thus changes considerably in the future [Mes99b]. While the speed of light is of considerable importance in some scientific computing applications, it is more generally not a serious limitation. Of greater practical concern is finding architectures that achieve scalability to maintain performance characteristics acceptable to individual users, even as the number of users sharing the infrastructure increases dramatically.

Improvements in performance per unit cost imply a wider range of applications accessible at an affordable cost. (By *application*, we mean functionality specific and specialized to the needs of end users, as distinct from *infrastructure*, which supports all applications.) This is most apparent in the expanded functionality of battery-operated portable devices. At the same time, achieving desirable cost objectives in any fixed application becomes easier over time. While much engineering effort has traditionally been devoted to answer the "how" question—how can technology be improved to achieve necessary levels of functionality at an affordable cost—increasingly the "what" question comes to the fore. That is, given an impressive set of technologies that can be molded to almost any conceivable task, what do we chose to do with them? This is addressed briefly below.

Aside from representation of information by bits, another fundamental idea separating

computers from other manufactured goods is *programmability*. The detailed steps in processing are defined by software, which is not specified at the time of manufacture but added later. This has led to a computer industry culture in which the infrastructure is designed without presupposing applications, where those applications can be conceptualized and added later. This culture is better positioned for the exploding diversity of applications that can be expected in the future, as discussed below.

3 Applications and Convergence

Given the increasing importance of the "what" question, it is useful to reflect briefly on the broad class of applications that are enabled by convergence.

3.1 Merging three traditions

Historically, three distinct application categories have arisen around the three component technologies: the algorithmic tradition for processing, the document tradition for storage, and the collaboration tradition for communications. Most major applications today fall predominantly into one of these categories. The *algorithmic* tradition is the basis for applications like signal processing, financial engineering, and scientific and engineering modeling. A broadly encompassing definition of the document would include pictures, video, and audio in addition to the printed document. The *document* tradition thus forms the basis for commerce, scholarship, and entertainment. Finally, the *collaboration* tradition supports groups of people working together, playing together, or socializing. This tradition also supports *coordination*, which is an element of many human and technical activities, such as design collaboration and business or military logistics.

Convergence enables and encourages applications that freely integrate the algorithmic, document, and collaboration traditions. In fact, many human and organizational activities integrate elements of these traditions. For example, an automotive design project may be conducted across several continents. It would mix modeling and simulation (the algorithmic tradition), coordination of concurrent design activities across time zones, group meetings (the collaboration tradition), and a stored and updated design database that captures the current status of the design (the document tradition). While these activities could be supported by logically separate applications—where only the people formed the linkages—in fact there is great benefit to integrating all these elements together, as enabled by convergence. For example, the design database can capture the complete trajectory of the design and how it arose from the collaboration, decisions made during collaboration can be automatically documented, and the implications of those decisions can be immediately characterized by modeling and simulation.

3.2 Sociotechnical applications

Generalizing this automotive design example, a major opportunity for convergence is *sociotechnical applications*. A sociotechnical system combines groups of people working on collective tasks (that is, an organization), information technology (processing, storage,

and communications), voluminous information and knowledge (the latter mostly the province of the people), and many physical elements (like material, goods, and shipping). The sociotechnical application is the information technology portion of this mix. All three traditions (algorithmic, document, and collaboration) are highly relevant to such applications, and usually integrated in intricate ways.

In sociotechnical applications, information technology is only a *part* of a sociotechnical system, and thus cannot be meaningfully addressed in the absence of the other elements. It is embedded in the system, much as a controller is embedded as an integral part of a mechanical system. As mentioned before, the information technology is subject to few physical limitations, and can thus be molded in ways that best accommodate application needs. But this shouldn't be done in isolation, because interrelated issues include the partitioning of functions between technology and people, the organization of the people as well as the technology, and the appropriate interface between technology, people, and organization. The design of sociotechnical applications is—considerably more than the applications of the past—an interdisciplinary activity involving technologists with organizational specialists and social scientists. Appropriately, professions intermediary between end-user individuals and organizations and technologists (not dissimilar to architecture in the domain of structures and landscape) are arising.

Another distinctive characteristic of sociotechnical applications is their diversity and specialty. Whereas the applications of the past (such as telephony, video distribution, word processing, and the like) had wide generic appeal, sociotechnical applications tend to be more specialized. This further points to the important involvement of end-users (or others who understand their needs) in the conceptualization and refinement of these applications.

4 Industrial Organization

Rapid and substantial changes are occurring in the computer and telecommunications industries, as driven by convergence, by the changing character and increasing diversity of applications, and by the increasing importance of the "what" question. It is useful to reflect on the likely and appropriate changes in industry direction.

4.1 Five industry players

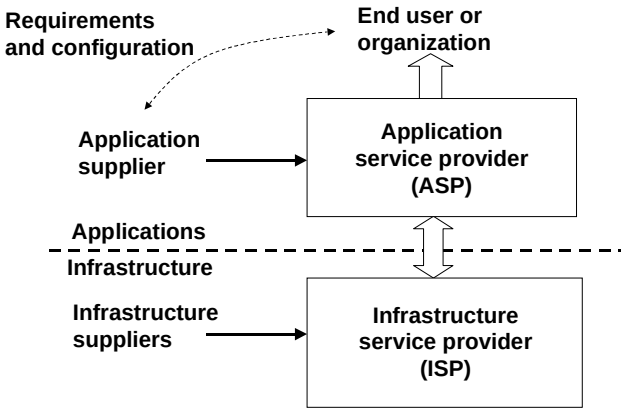

Figure 1. Five types of industry players.

The industry can be divided into five functions as shown in Figure 1. The *infrastructure service provider (ISP)* provisions, operates, and maintains the infrastructure supporting all applications. The ISP uses equipment and software purchased from *infrastructure suppliers*. The application may also be provided to end-users over the network, in which case there is an *application service provider (ASP).* (In many cases an application may be installed and operated by the end user or end-user organization, in which case there is no ASP.) The ASP acquires the application from an *application supplier*, which is usually a software company. (Applications may also be developed internally by end-user organizations.) Finally, the end user or end-user organization is the direct beneficiary of the application. It is important that the application supplier establish application requirements in cooperation with end-users, a relationship that would not normally involve the ASP.

4.2 Separation of application from infrastructure

Historically, the number of distinct applications of communications has been limited, including telephony, video broadcast, and data communication. A cause and result is that the infrastructure has typically been designed with one application in mind, such as the public telephone network for telephony, cable networks for video broadcast, and the Internet for computer communication. In contrast, due to its programmability, computers have developed a much greater diversity of applications. The computer makers have not presupposed applications, but allowed them to be defined, developed, and marketed by others long after the computer is manufactured. The number of computer applications is thus vastly greater than the number of telecommunications applications.

If the outcome is to be a diversity of applications, providing greater value to users, it is critical to separate infrastructure from applications. A test of this separation is if an application can be deployed without the involvement or knowledge of the ISP.

Successfully achieving this separation reduces the barriers to entry of new applications, bringing in many new players and many new ideas, including those directly from end users. Ultimately this is in the best interest of ISP's, as the value of the infrastructure to users is enhanced, and usage is increased.

We therefore recommend the dogmatic separation of ASP and ISP functions in Figure 1. This does not mean that the ASP and ISP are necessarily separate companies, but rather maintain a strict logical separation.

4.3 How to answer the "what" question

A proven and effective way to identify compelling new applications is to find ways to try out many ideas with end users, and use that experience to refine and further develop ideas that show value, and abandon those that don't. To be practical, this experimental process requires the separation of infrastructure from application, and also requires the close cooperation of the end-users and the application developer. Two remarkably successful new technologies of our generation, the personal computer and the Internet, both followed this approach, albeit within limited domains of applicability. Both were deliberately conceived to encourage experimentation and minimize barriers to entry by exploiting programmability. The Internet chooses to keep the network simple and generic, relegating to hosts (computers connected to the network) the definition of software-defined applications. The PC and the Internet have each marginalized incumbents (in computing and telecommunications respectively) by reducing barriers to entry for new applications and by granting users more control over applications.

It is worth examining briefly why the Internet in particular has been so successful, so that this model can be replicated. The separation of applications from infrastructure is one key, but also the fact that both the Internet and its applications were developed in a research rather than commercial environment allowed continuous user feedback and refinement. The layered architecture allowed additional capabilities to be added incrementally. The standardization process, while necessary, was kept open and inclusive, so that the best ideas from any and all were incorporated. The most important success factor was the first-mover advantage: before alternative standardization processes could complete, the internet protocols were already firmly established with large numbers of users and a collection of compelling applications refined through user experience was already implemented and available.

The Internet and PC also illustrate another reality of our age: universities and startup companies are effective at initiating and developing new applications, because they can tackle many ideas cheaply, and refine them through user feedback. Large companies are generally only interested in applications with obvious mass appeal, and usually follow a less-effective methodology of full refinement and development prior to user experience. Large companies are most effective at the deployment, operation, and customer service of established applications (like telephony).

4.4 What is service infrastructure?

There are several related questions of interest: First, what portion of the infrastructure the user owns, and what does a service provider offer as a service? Second, what exactly constitutes the infrastructure? Third, what portion of the application does the end user provision (like software on a personal computer)? These questions come to the fore with the advent of customer-provisioned infrastructure in the form of PC's and portable terminals.

Recently, applications have been implemented in software running on a PC or PDA, or embedded in user-owned equipment such as a telephone handset or facsimile machine. Encouraged by the success of applications depending on the World-Wide Web, there is increasing industry interest in the ASP model. (Of course, this is not a new idea, as illustrated by the telephone network where the telephony application is supplied as a service.) The ASP model is advantageous for both the provider and user. An ongoing revenue stream is attractive to the ASP, as opposed to the fixed price of the application product model. There are opportunities for price discrimination and advertising, enhancing supplier revenue. The user avoids the hassles of installation and integration, as well as maintenance, and may find attractive a contractual obligation on the part of the ASP for availability and quality of service. Applications can become location and terminal independent, which is especially attractive to travelers. Quality and availability is enhanced, because the ASP environment is more controlled, and because the ASP takes responsibility. For these reasons, the ASP model is likely to become common. It represents a natural stage of maturity in the application software industry.

4.5 Processing and storage in infrastructure

The ASP model requires that processing and storage join connectivity as a basic infrastructure element. Of course, processing and storage have been embedded in the basic operation of the infrastructure for some time (particularly in switching control), but they become resources directly available to applications as well.

There are other reasons for processing and storage in the infrastructure. A major opportunity is *valued-added infrastructure (VAI)*. One example is information brokerage, which intermediates between the users and numerous sources of information. Another is VAI that makes it easier to design and deploy new applications, much like the operating system aids development of applications on the PC.

The ASP model presumes reasonable network performance to enable interactivity. This is aided if applications are run topologically and physically close to the user, reducing connectivity resources and improving performance (including overcoming the speed of light limitations mentioned earlier). Processing and storage (such as compression and caching) can also compensate for limited connectivity (like wireless access bottlenecks) [Mes99b].

Most important, the addition of processing and storage make the infrastructure itself programmable from an application perspective, opening up many possibilities for innovative ideas. For example, an attractive opportunity is reducing the role of standardization in applications, as described later.

4.6 Stovepipe to layering

The architecture of an application implementation and infrastructure defines the modules of which it is composed, the functionality of those modules, and their interaction [Mes99ab]. Since each such module is readily designed and maintained by different companies, architecture also defines the boundaries of competition and complementarity in the industry—the industrial organization. As a result of convergence and other factors, the industrial organization is rapidly changing. Most evident is a shift from a stovepipe to layered architecture.

In the most extreme case of a *stovepipe* architecture, an entire infrastructure and application are designed as a unit. Examples of such vertically integrated application/infrastructures are the public telephone network, digital cellular, and video distribution network (cable TV or direct-broadcast satellite).

The layered architecture organizes functionality as horizontal layers. Each layer depends on the layer below, either elaborating or specializing its functionality. Within the infrastructure, in contrast to the stovepipe, each layer is designed to be general, supporting a variety of applications. (Thus, layering can also be termed horizontal integration.) It is easy to add new applications, without requiring the investment in the design and provisioning of a new infrastructure, by layering them on an existing infrastructure. (This is the separation of application from infrastructure advocated earlier.) Also it is possible to add new infrastructure layers that elaborate or specialize further what was there previously, or replace existing layers with new technologies without impacting other layers. (In practice this is difficult, as it requires great forethought and discipline to design layers that can be extended arbitrarily.)

The Internet is not a stovepipe, as it methodically separates applications from infrastructure. In its present form, the Internet is also not horizontally integrated in the sense defined here—as originally designed it cannot support some applications with strict quality of service requirements. However, it is likely the Internet can be modified and generalized in ways that will render it a layer acceptable to (if not ideal for) a wide range of applications.

4.6.1 Proposal for a layered architecture

It is important to define a coherent layered architecture that provides a framework for industrial organization. It allows firms to focus on their special competency while the architecture provides the needed coordination. While a layered architecture is well entrenched for the network (embodied by the internet protocols together with capabilities layered on top of them), what is really needed is a layered architecture that unifies

processing, storage, and connectivity.

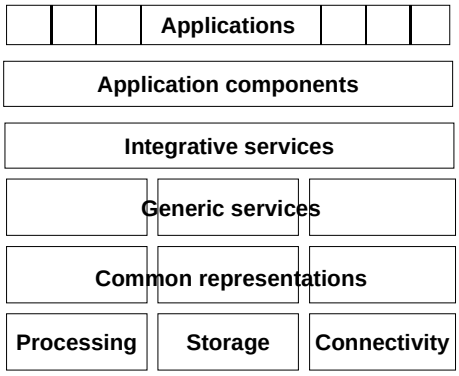

Figure 2. A layered unified architecture for processing, storage, and connectivity.

Figure 2 is an attempt at such an architecture. The bottom three layers are specialized to processing, storage, and connectivity, while the top layers integrate these three technologies for the benefit of applications. Starting at the top, brief descriptions of the layers are as follows:

- The *applications layer* includes a diversity of applications that provide direct and specific functionality to users. An example would be the global auto design application mentioned earlier.

- The *applications component layer* captures functionality that is useful to many different applications, so that they do not need to be re-implemented for each application. A simple example is a component that allows textual material to be edited by users—a requirement of many applications.

- The *integrative services layer* provides capabilities that integrate the functions of processing, storage, and connectivity for the benefit of applications. For example, this includes directory services, which uses stored data to represent and identify the location of various entities—an essential element of connectivity.

- The *generic services layer* provides services specialized to processing, storage, and connectivity—services that are widely useful to different applications, and also sufficiently diverse and general to meet the needs of all applications. Examples include the reliable and ordered delivery of data (connectivity), the structured storage and retrieval of data (storage), and the execution of a program in an environment that includes a user interface (processing).

- The *common representations layer* provides abstract representations for information in different media (like processing instructions, numerical data, text, pictures, audio,

and video) for purposes of processing, storage, and communication. These representations are deliberately separated from specific technologies, and can be implemented on a variety of underlying technologies, thus separating the irrelevant details of technology from the remainder of the infrastructure and applications. Examples include a virtual machine representing an abstract processing engine, a relational table representing the structure of stored data, and a bytestream with reliable and ordered delivery semantics for the communication of data.

- The specialized processing, storage, and connectivity layers provide the core underlying technology-dependent services.

Each of the layers shown could be further subdivided into smaller-granularity layers. In addition, layers may be added in the future. For example, there could be a layer above integrative services that provide a coherent infrastructure for information brokering—an intermediary between applications or users and a variety of information sources.

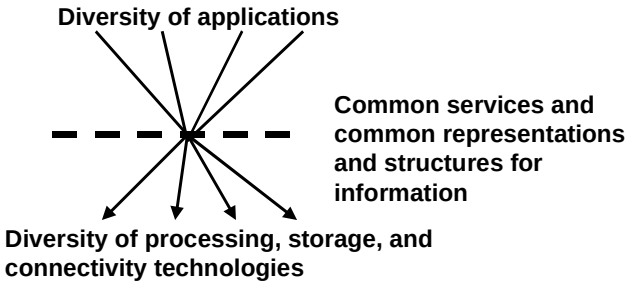

Figure 3 Illustration of how layering separates a diversity of applications from a diversity of technologies.

Backing away from the details, the essential idea of the layered model just presented is illustrated in Figure 3. The intermediate layers provide a common set of services, and also a common set of representations and structures for the information entities used by applications. The goal is to allow a diversity of technologies to coexist with a diversity of applications without imposing the resulting complexity on applications. Providing a re-implementation of the common representation layer for each distinct technology accommodates this. This separation of applications from technologies is an important part of separating applications from infrastructure, and is particularly important on a network where applications often have to coexist with multiple technologies. The middle layers are examples of VAI that requires the provisioning of processing in the infrastructure.

4.6.2 Solidifying the layers

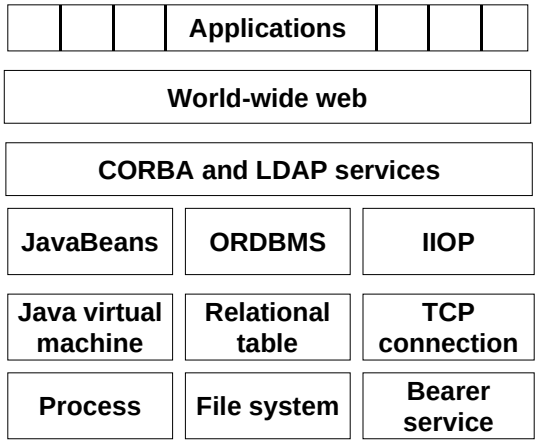

Figure 4. Examples of industry standards that fit somewhat loosely to the layered model shown in Figure 2.

Widely adopted infrastructure elements provide more value to applications (this is called a *network externality* [Mes99b]). Standardization of infrastructure thus adds value to end-users. An important practical question is how the layers are defined, and by whom? Two diametrically opposite options are by a top-down standardization process, and de facto standardization by market forces (similar to how the internet protocols were established). Each option has its advantages and disadvantages. For example, the top-down process is less chaotic, and allows greater reflection on the overall structure and goals. It attempts to provide a lasting solution to the whole problem, all at once. The refinement process involves the market in a sort of "survival of the fittest" strategy, and benefits from actual experience and success as winning technologies are chosen. It also admits that technologies are dynamic, and therefore it is not appropriate to establish the architecture once and for all. Even in this latter approach, network externalities typically result in a dominant market solution, at least at one point in time. These winner-take-all effects cause the market to eventually settle on a single (or at most a few) solutions, just as in the top-down process.

For the time being, the de facto standardization approach is most appropriate, because of the immaturity of the infrastructure. This approach will much more freely admit new innovations. When a stage of maturity is reached where the infrastructure functionality is well defined, traditional standardization processes can take over.

Many current industry standardization and commercialization efforts provide capabilities similar to those categorized in Figure 2. Examples are illustrated in Figure 4 (although the fit is not perfect—for example, the World-Wide Web is currently layered directly on TCP:

- At the common representation layer, Java supplies a common representation for program instructions that can be emulated on any architecture, the Java virtual machine (VM). Similarly, the relational table supplies a standard structured representation for data, the relational table, together with a query language for manipulating that structure, SQL. The Internet supplies TCP, a representation of data convenient for communication in many applications—the bi-directional bytestream with ordered and reliable delivery.

- At the generic service layer, shown are three standards that support the object-oriented programming model—Java in the processing domain, ORDBMS in the storage domain, and IIOP (a CORBA standard) in the connectivity domain.

- At the integrative service layer, the Common Object Request Broker Architecture (CORBA) attempts to identify and standardize a set of common services that integrate processing and connectivity (by incorporating Java mobile code capabilities) and processing and storage (by providing persistence for objects).

- There does not yet exist a viable market in application components—although some firms for their internal use have developed components—in part because the underlying infrastructure has not matured sufficiently. However, the World-Wide Web does provide a set of standardized components that are now widely incorporated into applications, particularly in the client presentation. (The Web is a nascent version of what may evolve into an extensive layer for information brokering, making it much more than application components, perhaps a layer of its own. The Web can also be considered an application in its own right.)

4.6.3 Layering and industry structure

A shift in industry structure is ongoing, from the stovepipe to layered architecture. In practical terms, this means that the most successful firms wishing to develop a core competency focus on one (or at most a few) layer. Their product solutions thus tend to be general, supporting a wide range of applications, but they constitute a total solution only when combined with other layers. Other layers (in the form of products from other companies) need to be composed and integrated to yield a complete solution useful to end-users. Focusing on one layer allows a firm to specialize and emphasize competence and differentiation in that domain, and at the same time provide the end-users with what they want—an infrastructure that supports a diversity of applications, each incorporating different information media. The layered architecture shifts the emphasis from competition to complementarity. Although awareness of competitors is always advisable, in a layered industry structure each firm is also critically aware that it does not provide a total solution and cooperation with complementors is essential.

A pragmatic view of layering is that as the infrastructure generalizes to support a variety of applications, no firm possesses or can develop the expertise to pursue a “grand strategy” that provides a total infrastructure solution for all applications. As it is forced to

specialize, a natural inclination is to maintain generality to all applications—thereby maximizing its market potential—while narrowing its focus vertically and depending on other suppliers to provide missing pieces.

4.6.4 Layering and education

Individuals face a similar problem to firms. As human knowledge expands geometrically, the capacity of each individual to become facile with that knowledge is relatively static. Individuals have been forced to specialize as to discipline—those disciplines partition knowledge in a manner analogous to the stovepipe architecture. However, in a world where systems and applications are more important than the past, successful individuals increasingly take a broad perspective, which is at odds with stovepipe specialization. One possible response to this is to redefine education to emphasize specialization in a layered rather than stovepipe sense. This idea is discussed further in [Lee99].

4.7 Changing role of standardization

One of the most important mechanisms for coordination among complementary vendors is standardization. As a direct result of the intermingling of ample amounts of processing, storage, and communications, the role and nature of standardization is likely to change.

Networked applications experience network externalities, making applications that are more widely adopted more valuable to users. For example, a particular facsimile product becomes more valuable to individual users as more users possess compatible fax machines. If suppliers offer incompatible fax machines, the value of each to users is reduced, and the fragmented market develops more slowly. A response is standardization, which enables fax machines from different manufacturers to exchange faxes. However, this standardization also limits innovation, and a lengthy standardization process can delay product introductions.

The elements of applications that are standardized may include all elements of the architecture: the modularity, the functions of modules, and the interoperability of modules. Another standardized element may be the representation of information elements used by the application. In the facsimile example, the standardized modules are a fax transmitter and receiver, where the functionality of the transmitter is scanning and compression. The compression is a standardized representation of the image, and the protocols used by the transmitter and receiver may also be standardized.

Avoiding standardization in applications (while still avoiding networked externalities) would allow much more experimentation in applications and reduce the barriers to entry, greatly enhancing application diversity. This is feasible if the application is software defined, and the infrastructure includes appropriate processing resources. Two basic ways of accomplishing this will be mentioned in the following subsections.

4.7.1 Mobile code

One way to achieve interoperability without standardization is if all elements of the

application—including those distributed to different users—originate from the same source. This can be achieved by dynamically provisioning all these elements in appropriate terminal and server nodes using *mobile code*. In the facsimile example, this would be analogous to making sure all fax machines originated from the same manufacturer by opportunistically supplying each user with a compatible machine as needed (as a piece of software over the network). Alternatively, one user wishing to send a fax to another could first provide a fax machine compatible with his own (as software over the network), followed by the fax.

An illustration of this is RealNetworks, which is able to advance its audio streaming application rapidly without worrying about network externalities. It depends on users to upgrade their players to the latest version. Mobile code will accomplish the same goal transparently to users.

4.7.2 Metastandards

Another approach to reducing the role of standardization is to exploit processing in the infrastructure, and standardize languages that can describe application elements, such as the interaction between modules, the functionality of modules, or the representation of information elements. We term this a *metastandard*, because it standardizes a way of describing something, rather than specifying that "something" directly.

For the facsimile example, there might be a metastandard for the compression algorithm (the binary representation of the scanned image). This metastandard would be in the form of a language specialized to describing a large set of compression algorithms. A typical description of a facsimile coding algorithm would be something like "use an 8 by 8 discrete cosine transform followed by a quantization algorithm of the following form…", where of course the result is constrained by the linguistic expressiveness of the metastandard. If each transmitting fax machine is able to forward a description of its image representation to the receiver, the latter can translate this into an implementation of the conversion of the representation back into a scanned image. Each manufacturer can innovate freely in compression algorithms, within the expressiveness of the metastandard.

Rudimentary forms of metastandards already exist. Examples include the Interface Definition Language (IDL) of CORBA, and the eXtended Markup Language (XML) for documents. Such standards can be constructed to be extensible, so that expressiveness can be augmented in the future without affecting current implementations. Metastandards depend upon the processing and storage elements of the infrastructure to translate descriptions into actual implementations. These processing elements can be dynamically endowed with metastandard translators using mobile code.

5 Recommendations

Computing-communications convergence is resulting in a profound shift in applications, infrastructure, industrial organization, and the role of standardization. Two industries—computing (especially software) and telecommunications—are being radically transformed by convergence. One question is, do both industries gain, and if so equally?

What lessons are to gleaned from the past, and what advice can we give to these industries?

As should be evident from the foregoing, technology developed by the software industry is dominating the converged infrastructure. Of course, communication links are an important contribution by the telecommunications industry, but almost everything else is an outgrowth of the software industry. While this is a manifestation of the separation of applications from infrastructure mentioned earlier, it is telling to reflect on the historical background. The telecommunications industry is much older, and developed around serving a few individual applications—telephony, video distribution, and data communication. Over the years, it has attempted to expand this stable of applications (at least in the U.S.), with major initiatives like videophone, videotex and video-on-demand—usually with a striking lack of success. The computer industry, on the other hand, has always focused on developing infrastructure independent of and without presupposing applications. This falls back to the genesis of the industry—the idea of programmability avoids defining functionality at the time of manufacture.

As we move into an area of much greater diversity of applications, the computer industry approach is more effective. To do what telecommunications has attempted to do—identify, develop, and subsequently deploy an application with wide appeal and with little user experience—is simply too difficult. In contrast, the separation of application from infrastructure has resulted in some surprising new applications, often coming from the users themselves (the World-Wide Web is an inspirational example).

Does this mean that the software industry has hijacked the agenda, and the telecommunications industry is relegated to an increasingly minor role? Certainly not! The core competency of the telecommunications industry—a competency lacking in the computer industry—is the *operation* of infrastructure and applications, high *reliability* and *availability*, and *customer service* (including service initiation and billing). This competency is still very much in need in the ISP role, and it is also needed for the ASP role.

The telecommunications industry has an opportunity to expand in both horizontally and vertically. Horizontally, processing and storage will join connectivity as basic infrastructure elements. Vertically, there are many opportunities for VAI, including innovative new VAI that cannot even be anticipated. Companies should look for opportunities to expand the infrastructure (by adding additional layers) in ways that makes it easier to develop and experiment with new applications. Finally, expanding into the ASP role is a vertical opportunity.

The historical fact that the telecommunications industry has emphasized services (and has in place both processes and infrastructure) is an advantage over the software industry, which has emphasized products. It isn’t necessary to define applications in order to benefit from them, just as the electric power industry defines no uses of its service.

Our advice to the telecommunications industry is therefore to exploit its core competency in the operations of reliable infrastructure and customer support. Rather than trying to

identify new applications, concentrate on making the infrastructure support *all* applications, including by adding processing and storage services to its traditional connectivity services. Insure that its infrastructure is open and available to everybody with good application ideas. Look for opportunities to expand the infrastructure to make it easier to experiment with and deploy new applications. In the future, there will be a diversity of more specialized applications. Many of these ideas will be formulated by end users, not by developers or service providers, but those end users will be receptive to delegating operations to providers.

In the final analysis, this can be a symbiotic relationship between those conceptualizing and developing new application ideas and those provisioning and operating the infrastructure to realize them. The software industry will focus on defining and developing the new applications and associated infrastructure technologies. The telecommunications companies can deploy, operate and maintain the required infrastructure, and support the customer. Telecommunications will benefit directly from those compelling new application ideas, because it will be in a position to license the software and supply them as a service. Both industries should see tremendous opportunity looking forward.

6 Conclusions

The convergence of computing and communications is having a major impact on two industries. While each has attempted to move into the other's territory—telecommunications into new applications and computing and software into infrastructure and application service provision—we have argued that each industry should focus on core competencies, in which case their roles are entirely complementary. We have identified a number of other issues arising from convergence, such as the appropriate new applications and the appropriate role for standardization. Over all, however, it is a time of exciting opportunity for industry, and especially for the end users and end-user organizations that will benefit from a diversity of new applications integrating the algorithmic, document, and collaboration traditions.

7 References

[Mes96] D.G. Messerschmitt, "Convergence of Computing and Communications: The Implications Today." *IEEE Proceedings*, August 1996.

[Mes99a] D.G. Messerschmitt, *Networked Applications: A Guide to the New Computing Infrastructure*. San Francisco, Morgan-Kaufmann, 1999.

[Mes99b] D.G. Messerschmitt, *Understanding Networked Applications: A First Course*. San Francisco, Morgan-Kaufmann, 1999.

[Lee99] E.A. Lee and D.G. Messerschmitt, "A Highest Education in the Year 2049". *IEEE Proceedings*, Sept. 1999.