

R.A. Greenes,
S.R.A. Deibel

Decision Systems Group,
Brigham and Women's Hospital,
Boston, MA, USA

Review Paper

Constructing Workstation Applications: Component Integration Strategies for a Changing Health-Care System

Abstract: A health-care workstation is the means by which a professional interacts with the information artifacts of health care. However, a major transformation is taking place in the software architecture of health-care systems that alters significantly the role of the workstation. Health-care systems are becoming more complex in response to the need to support "extended enterprises" across regions, and provide both horizontal and vertical integration capabilities. Component-based software methodologies are being introduced that match well the needs of these large systems and the component services they must integrate. In the component-based framework, a workstation functions less as a "portal" for information transactions carried out on distant host computers, and more as the "orchestrator" for tasks involved in assembling, organizing, presenting, and manipulating information. Applications residing on workstations access distributed software components that carry out encapsulated functions for the application. Component-integration methodologies include both formal and ad hoc approaches; the principal emerging technologies are the World Wide Web (WWW), CORBA, Java, OLE, and OpenDoc. An emerging strategy appears to be that of developing application integration environments that encompass and support all of these integration methodologies. Component-based approaches also facilitate standardization at the message level, as messages to classes of components can serve to focus such standardization.

Keywords: Workstations, Integration, WWW, Corba, Java, OLE

I. Introduction

A workstation is the means by which the health-care professional interacts with the information artifacts of health care. Some of these artifacts may be information resources such as patient data, images, and plans; others may be education and decision-support resources, research materials, and policy and guidelines. Still others may be representations of communication processes - electronic mail messages and stored video images of human participants, or live audio or video communication.

The nature of professional workstations is undergoing change because of transformations in both health care and in the information-technology world that affect directly what a workstation's purpose is and how it should be designed. An IMIA Working Conference on the Health Care Professional Workstation was held in June, 1993 [1]. We take this opportunity to review the progress that has been made since that time and to identify the most promising current approaches to construction of workstation applications.

1.1. *Seeds of change*

Significant paradigm shifts are now underway coincidentally in both the health care and the information-technology world. Surprisingly, they have a number of similar features. Both have major implications for the purpose and design of professional health-care workstation applications.

1.1.1 Re-engineering of Health-care

Change in the health care-delivery system has been stimulated by decades of inexorably rising costs that finally reached a point where the need

to control them has led to the adoption of alternative financing/payment models. Pursuit of cost effectiveness, efficiency, and quality have led to the creation of a variety of new enterprise forms for health-care delivery. The major direction of change is from a system dominated by monolithic and independent providers (physicians, small groups, hospitals, and other services), to ones in which a "federated" model predominates. New enterprises include regional networks, integrating patients, primary-care practitioners, community hospitals, tertiary centers, and other services formerly independent or only loosely associated; national chains of managed-care organizations and hospitals; and vertical services such as radiology and pathology offered on a national or even international basis [2-4].

The "extended enterprises" now being formed involve alliances that cross institutional boundaries and, in some cases, are no longer constrained by geography (e.g., for teleradiology services). Competition among practitioners and other service providers is causing them to learn to be more efficient and to function as "components" in the extended enterprise, and the enterprise management to function as the integrators of these providers and services. While the impetus for such change is less strong in some nations than currently in the USA, the root problem of increasing cost of medical care, requiring systemic change in health-care delivery, is widespread.

These new forms of health-care delivery are made possible in large part by, and dependent upon, information technology. Existing health information systems, largely evolving from hospital-based systems, are not typically well suited for the new requirements of the health-care industry. The information systems to support the health-care enterprises of the future will need to be adaptable to new forms of delivery, and to provide both inter-

faces to the various components of the enterprise, and enhanced communication, monitoring, and analysis of process and outcome.

1.1.2 Information Technology Upheavals.

The information-technology industry is curiously going through a transformation very similar to that of the health-care industry: from monolithic and independent providers of applications to a highly federated model of software [5]. During the past few years, a variety of methods has become available for constructing applications by linking of distributed information resources as well as application components. Large mainframe- and mini-computer-based applications in health-care and in business are, after years of inertia, beginning to give way to networks of client-server systems with users interacting via workstations [6].

Personal-computer applications, largely in the form of independent programs or single-vendor suites of programs, have in some cases become bloated integrated, feature-laden behemoth programs. These applications are beginning to see challenges in the form of new applications that are constructed in an entirely different fashion, as an integration of components, plug-ins, "applets", or other extensions, often provided by multiple vendors, and with network access to these elements and to other tools and services.

"Application as integration" is a paradigm in which application-specific behavior is divided into the task of providing a user-interface layout, that of "orchestrating" how individual software components are brought together, and the tasks performed by the components. Components function as a "middleware" layer, adapting information resources to application-specific needs [7]. Object-oriented technology is a key underpinning of component middleware [8]. Traditional software providers are being chal-

lenged by this new model to find new ways to incorporate network awareness, and to make some or all of their services into separable components.

2. Requirements

2.1. Workstation Applications as a Focal Point

As health-care is being reshaped through new enterprise forms based on a component/integration strategy, the information technology to support it is also using a component/integration strategy to reinvent itself. As a consequence, it is increasingly feasible to match the record keeping, decision making, education, and communication needs of a health-care user with functional components that meet those needs, which possess the information tools and services suitable for these tasks. This is the essence of object technology, in which functionality is encapsulated in the object. In other words, one can create a mapping of functional needs to information componentry. This should facilitate the construction of workstation applications that are highly tailored for specific uses.

Note that in the above and the remainder of this paper, we will consider health-care "workstations" as the user devices through which a health-care professional (provider, manager, or other worker) interacts with the health-care system and other participants. The workstation can be a device of personal-computer size or of more powerful class, traditionally referred to by the "workstation" rubric; we will consider these interchangeably for our purposes.

In the extended enterprise, a health-care workstation application may need to support a variety of users, with configurability suited to those users' information needs and the demands on them arising from their enterprise roles.

A primary change that has occurred is the need for practitioners to be now more sensitive to the demands for systemic cost-effectiveness and quality which may, at times, conflict with their perception of optimal care for individual patients. Information tools to support informed decision making, including guidelines and education resources, are becoming more essential as part of routine practice, and capturing of events for monitoring and analysis are now becoming necessary. As a result, the number and variety of interactions required of a health-care worker are increasing. A primary-care physician may need to view not only patient data and images, but guidelines and other decision aids, directories of referring physicians, and various reference materials. Findings, interpretations, and plans need to be recorded. Problem-specific information resources may need to be provided to the patient.

To enable this set of capabilities, specialized services need to be made available by a variety of information sources, including the electronic medical record system and PACS, as well as referral directory servers, guideline repositories, literature reference services, expert system and analytic tools, and others. Because of their diversity, they are likely to be found on disparate systems rather than on a single institutional computer. Even if such functions could be offered within a single system, an institution might find this limiting in terms of the ability to find "best of breed" for each function and make it available. Ideally, desirable services should be able to be incorporated into the system, independent of where they are located, or whether they are internally provided or external.

2.2. Roles for a Professional Workstation

Many kinds of health-care workstation applications are possible. We fo-

cus here only on the framework for constructing them. The following are roles performed by a workstation application.

Identification and Context-setting

The workstation must identify and authenticate its user and establish authorization and access privileges. The nature of the application depends on who its users are and the tasks it is called upon to perform. One can envision patient-oriented, student, nursing, primary care, specialist, manager, policy analyst, or researcher workstation applications. Each of these individuals has a different world view, is engaged in a specific set of activities, and has particular information and communication needs. The interface characteristics and behavior of the workstation also should be based on the most effective mode of interaction for that user. This could depend not only on the nature of the tasks, but also on individual preferences, or on features of the environment. For example, a patient interacting with the health-care system at home via an interactive two-way television system might use primarily point-and-click and voice for communication. A traveling nurse doing home visits may depend on a portable digital assistant with pen-based input and wireless communication. A radiologist would be likely to require a multi-screen display station with capacity for displaying, manipulating, and annotating high-resolution images.

Portal for Access to Information

A workstation is usually not itself the location for information but a portal for locating and accessing the information. For example, a primary care physician examining a patient with recurrent dizziness might need to access the medical record for prior clinical history, medications, and other data. She may wish to check the side effects of the medications being taken, or

consult a workup guideline. Considering the problem to be neurological in nature, she may wish to find a neurologist to whom to refer the patient from an on-line directory.

A radiologist interpreting a case might need to access the prior clinical history of a patient, images from previous examinations, tools for manipulating the images, such as 3-D reconstruction, image enhancement, or image fusion, computer-aided detection tools, anatomic atlases, differential-diagnosis aids, teaching cases relevant to the problem at hand, forms for structured reporting, clinical guidelines for appropriate workup of the problem, and multimedia mail facilities for sending an interpretation together with annotated images, highlighting the pertinent findings.

Locus for Information Integration

By supporting access to multiple kinds of information resources, as identified above, a workstation application is increasingly being called upon to bring together these disparate resources for the purposes of a particular task. The complexity of health-care is pushing clinical applications, for example, to utilize educational and knowledge-based resources such as clinical guidelines to be more cost-effective, forcing tighter integration among separate resources.

Means for Initiation and Recording of Events

The workstation is a primary location at which data are entered, decisions are made, and orders initiated. Medical record keeping through form completion, dictation with voice recognition or transcription, or pen or keyboard entry are carried out. Since the workstation identifies the user, with security and authentication procedures imposed as required, it is responsible for tagging all data and events initiated by the user with appropriate source identification and date and time stamps. Captured events can then be recorded

in archival database storage, and used for alerts and reminders, for research, for audit, for education, or for process and outcome studies.

Locus for Communication

The workstation is a place for initiating and receiving communications. This includes email, video conferences, and alerts and reminders.

Functional Linkage

A workstation application can serve to coordinate interactions among disparate information resources that provide specific functionality, facilitating the flow of information among databases, specialized processing tools, and other services, located either on the same computer or on different host computers and servers.

2.3. Design Desiderata

In considering the features required for workstation applications, the following appear to be most important:

Ubiquity

The workstation application needs to be adaptable to the work habits of the individual. If a practitioner moves from place to place or covers patient care while at home as well as at an office, then workstation capabilities either need to be portable or accessible from these different locations. It also should support a variety of different hardware and software platforms.

Consistency

The workstation integrates a variety of functions. Thus, it is important that it does this via a consistent user interface so that each function does not require unique forms of interaction.

Completeness

We also require that all of the capabilities needed be available within the framework of the workstation application so that they can be integrated with one another and avoid duplicate entry of

information to accomplish related tasks.

Ease of Use

In addition to consistency, ease-of-use depends on matching user-interaction mechanisms to the setting. Patient-oriented interactions will need to take into account literacy levels, ability to type, availability of certain hardware, etc. Professionals on the move require highly portable devices with flexible data-entry tools. Some workers need to keep their hands free, which will favor use of voice input, whereas accuracy requirements or ambient noise levels may require pen or keyboard input in other situations.

Adaptability to User Needs, Configurability

Some users prefer viewing information in different formats, e.g., graphical vs. narrative vs. tabular. Some prefer pointing devices, keyboards, pen, or voice as the primary input mode. Some require adaptability to both portable and desktop use, depending on where they are located. Depending on the kind of user, different views into the same information resources and presentation formats are most appropriate.

Extensibility

As new requirements are identified, the workstation should be able to incorporate information services to meet these needs without extensive modification. Changes can be expected to be relatively frequent, so this is essential for the longevity and continued utility of the workstation.

Point of View

It is desirable for the workstation application to be able to support a specific point of view that matches well the tasks of the user. Depending on the user, a suitable point of view might be, for example, chronological, source-oriented (e.g., specialty-based), body- system oriented, problem ori-

ented, or management oriented.

We posit that the problem oriented point of view will become especially important as a means to cope with "information overload". Physicians, for example, faced with the need to integrate clinical data, medical knowledge, institutional constraints, and patient needs and preferences, will require assistance in focusing on the specific information pertinent to a current problem. Problem-specific information access is a goal of the GEODE-CM project at our institution; patient-management states are used to anticipate the information-resource needs of the physician, to facilitate data access, structured record keeping, decision making, and workflow management [9].

3. Construction of Workstation Applications

Although we will focus the subsequent discussion on component integration, we give a brief historical perspective, because it helps to understand some of the forces that affect the speed or lack thereof in adoption of new paradigms.

3.1. Historical perspective

The earliest health-care systems involved user interaction via video-display terminals. In that situation, the application executed entirely on the mainframe or minicomputer to which the terminal was connected, and was responsible for controlling the nature of the user interface at the terminal. These are called *central applications*.

In many circumstances, personal computers and workstations are also used for terminal emulation, to execute applications on existing health-care information systems, in which the application remains primarily central. Nonetheless, the introduction of workstations generally shifts the locus of an application centrifugally. The extreme

is the use of workstations for executing stand-alone *workstation-based applications* or those in which use of a network is limited to sharing of file and print services.

Workstations tend to function now most often in a hybrid fashion, in which the application is split between peripheral and central functions, for example, providing a graphical user interface to "legacy" mainframe or mini-computer-based central systems. Besides improving the "look and feel", user friendliness, and ease of use, these have the added advantage of allowing new features to be added to the workstation. For example, some workstation-based health-care applications have been developed for specialized reporting (e.g., voice input stations [10], or pen- or mouse-based structured form input [11,12]), for image interpretation [13], or for teleconferencing [14]. All of these have the feature that they share common databases but add new functionality. These are *client-server applications*. The workstation functions as a client, providing graphical user interface, data entry, and some local processing capabilities; and one or more servers support storage and retrieval of data, dispatching of communications, and other shared functions.

Component integration is the newest approach to constructing applications; yet, it is an extension of the client-server approach. The invocation of a component by an initiating process is essentially a client-server relationship. Applications of this new breed, however, have a many-to-many, rather than a primarily many-to-one relationship between clients and servers. They have the ability to create linkages among multiple kinds of functionality which may be on different systems. For example, a radiology interpretation station may link a hospital information system, a radiology information system, a PACS (Picture Archiving and Communication Sys-

tem), and various knowledge sources. A clinical workstation may link patient data to external expert systems or educational resources, or to a MEDLINE search.

3.2. Component Integration Methodologies

Component integration shows promise as a most robust model for workstation construction. The principal emerging technologies for this are the World Wide Web (WWW), CORBA, Java, OLE, and OpenDoc.

3.2.1 The World Wide Web

The basic paradigm for WWW is one in which documents residing on distant computers are accessed by a workstation-based "browser", and through which hypertext linkage among documents can occur, all of which can be potentially located anywhere in the world, provided that they are on Internet-connected computers [15-17]. In this setting, the application is not the WWW browser, but rather the set of interlinked documents collected for a particular purpose. The browser may be considered to be essentially a facilitator of the process, a kind of extension of the workstation's operating system to enable WWW access.

The WWW hypertext-transfer protocol (http) is used as the invocation mechanism for obtaining remote documents. The documents may be stored statically on disk, or they may be generated dynamically in response to an http server invocation. The latter is achieved by treating a server-side script or program as the target of an http request, and invoking the program through the Common Gateway Interface, or CGI. Scripts may involve access to a database, arbitrary processing of data, and construction of a document to be returned to the requesting workstation.

Data can be entered into WWW

forms and sent to distant computers. In fact, the term "browser" is somewhat of a misnomer for the WWW access tool, as the inclusion of forms for data entry, applets, and other mechanisms for tailoring behavior enable the WWW environment to support transaction processing and other interactive functions. Thus, it can be used for health information system applications.

Documents displayed by the browser have an internal format, in which various elements of the document are encoded in tagged fields. The markup language used for tagging, HTML, can identify text, headers, images, tables, and other elements, some of which have become agreed upon as standards and some of which are proprietary implementations by specific browser implementors. Since the HTML tags can be arbitrarily extended, elements can be defined that add new kinds of functionality. One of these is a Java applet (see below), which is a set of code to be executed as a small program on the client workstation.

WWW document elements are interpreted either by the browser or by other auxiliary or "helper" programs, which might include a Java runtime engine, an image decompression and display tool, a 3-D animation tool, or other locally resident "plug-ins" or extensions.

Cross-platform execution is possible to the extent that the browser and all auxiliary programs exist for each target platform. Extensions allow the user interface to be somewhat more flexible than the basic hyperlinking interface mode supported by most browsers.

A number of groups have used the WWW model to build professional workstation applications for accessing and updating the electronic medical record [18-20], for supporting research investigation [21], and for education [22,23], among other applications.

The various functions performed

by a WWW-based application may be viewed in some sense as components, and the WWW as the integration mechanism. The main deficiency of the WWW model is that it does not intrinsically foster standardization of components. Standards can be created, to be sure, but any ad hoc document or applet can be linked to any other document. Thus, it requires discipline extrinsic to the WWW model itself to attempt to identify functional components and to standardize interfaces to them. Certain types of images (GIF and JPEG) are supported, as are sound and motion video sequences. The WWW can be used to access data from relational databases by using a common interface standard for SQL query called ODBC (Open DataBase Connectivity) developed by Microsoft Corp. [24]. Beyond that, however, very little convergence on standards has yet occurred. Components for accessing EMR data could be developed though, supporting HL7 [25], and others for accessing clinical images supporting DICOM [26,27]. Still others could be adopted for accessing MEDLINE (e.g., based on Z39.50 [28]), drug-formulary references, cancer-trial protocols, clinical practice guidelines, and other kinds of capability.

3.2.2 CORBA

The Common Object Request Broker Architecture (CORBA) is an approach to formally standardizing network-distributed software components [29,30]. CORBA is a set of standards put forth by the Object Management Group (OMG), a world-wide industry consortium including over 600 software vendors and virtually all major computer and operating systems manufacturers (with the exception of Microsoft Corp.; see below).

With CORBA, a health-care application can be constructed for a workstation in which a table, a graph of laboratory data, or a radiograph is produced by a component which retrieves

the data from an appropriate clinical database or image server, an interpretation is produced by a component connected to an expert system, and a guideline for further workup is displayed by a component connected to a guideline server [31].

Functional elements of the system are decomposed into components, such that these are invoked through formally defined interfaces. The messages, data types, and message content for these interfaces are specified using a programming language-independent Interface Definition Language (IDL). These specifications are mapped onto invocation (client side) and implementation (server side) languages through the use of code-generation tools. Communication between the client and server code is encapsulated within the automatically generated invocation and implementation stubs, which hide the details of network-message transfer from the component implementor and client.

Thus, a component can be implemented in C, C++, SmallTalk, LISP, or any of a variety of other languages, and can interoperate with components written in other languages. Stub mappings that define the production of invocation, and implementation stubs from IDL have been defined and standardized for many of these languages. Additional standards for other aspects of the component run-time environment, including the Internet Inter-ORB Protocol (IIOP) for TCP/IP networked invocation, have also been developed, allowing components written with different CORBA implementations to interoperate. Still other languages such as Visual Basic and MUMPS can communicate with CORBA components if they are able to invoke extensions that implement IDL stubs.

CORBA is a truly platform-, programming language-, and network-independent standard, allowing the construction of mixed application environments that integrate diverse dis-

tributed systems. CORBA is both a powerful tool in architecting large distributed systems, and a good tool for linking between new workstation-based applications and older legacy information systems.

Applications can be constructed from CORBA-compliant objects, or components, by various "interface builders" or application-construction tools. These enable a user to determine the visual layout of screens, and to attach components to various visual and non-visual elements. For example, buttons may be attached to components that generate a list, retrieve an image, or analyze a set of data. The attachment is done by a scripting language that determines how events, such as user input or elapsed time, should be used to invoke a component, and which parameters should be passed to it. Several CORBA-compliant application-construction tools are now available commercially, and have been applied to the development of clinical workstations [32]. Another, known as Arachne, has been developed by our laboratory, a publicly available version of which is distributed by us via the Internet [33].

Since CORBA's IDL is object-oriented, it is well suited to the creation of component interface standards through the definition of IDL component classes. These standards are important in the development of a multi-vendor component market, since the integration of components relies as much on the semantic compatibility of component interfaces as it does on the standardization of the component's runtime environment. Arachne includes a variety of basic interface standards, including those for visual object display and screen-space arbitration, text and large image transfer and handling, and basic types for application GUI construction.

A component-based software engineering environment known as SEE, although not CORBA-compliant, was

used to implement the HELIOS clinical workstation [34]. The HERMES project is aimed at integrating distributed components to encapsulate functions of a variety of legacy systems to support clinical investigation [35].

CORBA-based applications in health-care are somewhat of a rarity yet, due to the relative newness of application-construction tools. However, components are being utilized also in a variety of other environments as well, to extend functionality.

A competing component-integration architecture known as COM (Component Object Model) [29] is supported by Microsoft Corp. (Bellevue, WA), the only major computer vendor that is not currently a participant in the Object Management Group consortium. This may have considerable impetus on the marketplace, because of Microsoft's influence. However, interoperability with CORBA is highly likely [29] because of the magnitude of its acceptance by other vendors.

3.2.3 Java

Java, as mentioned in the context of the WWW above, is a C++-like language [36] developed by Sun Microsystems Inc. (Sunnyvale, CA). Java is currently primarily an interpreted language (although compilers are under development) that is executed in the context of a runtime engine, available for many hardware platforms. Java capabilities are being licensed by Sun to many other vendors.

Although Java is a complete software development environment, current interest centers on its use in developing dynamically extensible applications. The Java technology allows small executable "applets" to be sent over a network from a WWW server (or other server) and executed on a client workstation containing the Java runtime environment. The result is that WWW browsers and other applications can be extended dynamically to include new processing and interface

functionality, as defined in the Java applets. Security of the client workstation is maintained by restricting the capabilities of the Java applet in terms of disk and memory access (e.g., to prevent the spread of viruses), and by allowing higher levels of access to applets on the basis of their authenticated and verified origin.

The basic emerging model of integration in Java is thus the incorporation of add-on functionality into a central client application that acts as the host to applets. The application is the point of integration, and communication is centered around the retrieval, loading, and execution of visually manifesting Java components. Java currently does not support communication among applets, nor is the integration of non-visual applets stressed in its use. Standards fostering these additions are under discussion and are likely to be developed in the future.

3.2.4 OLE

Microsoft's Object Linking and Embedding (OLE) [29] is another standard that provides support for component development and integration. OLE is based on a model that currently relies on dynamic loading of component code where components publish an interface for invocation after they are loaded into the invoker's process space. The invoker makes calls to the component upon inspection of the component interface. With OLE Network, component invocation will be extended to include transparent network communication.

OLE interfaces are not object-oriented, and are not specified formally through the use of an invocation interface-specification language as is done in CORBA. As with Java, the bulk of interest in OLE has been on its use in developing dynamically extensible applications, and most OLE components are visually manifesting applets that are intended to extend the functionality of larger host applications.

OLE is controlled solely by Microsoft, and is licensed selectively to vendors. It is currently available on all Microsoft operating systems (Win3.1, Win95, WinNT) and Macintosh; it was not intended or designed to be cross-platform, and has not emerged as a competitor on UNIX and other environments.

3.2.5 OpenDoc

Visual embedding of parts of disparate applications on a screen is the early focus of most of the technologies described above, even where the technologies are intrinsically capable of supporting other types of integration. Another standard, called OpenDoc [29], is concerned specifically and exclusively with the visual integration of components in the context of a document-centric application architecture, where the document may consist of many dynamically assembled components. OpenDoc has been developed by the Component Integration Laboratory (an industry consortium of Apple, IBM, Adobe, Lotus, Novell, the Object Management Group, Oracle, and many others).

OpenDoc is based on IBM's System Object Model (SOM) and adheres to the standards defined by the CORBA specification, including the use of IDL in component interface standardization. OpenDoc represents a suite of document-centric component interface standards that are built on the generic distributed object standards of CORBA. Applications using OpenDoc adhere to these component interface standards and incorporate the OpenDoc runtime, which includes implementation of the basic document integration interface classes.

OpenDoc is freely licensed and can interoperate seamlessly with OLE-2 (although the technology is primarily based on CORBA). OpenDoc was designed to be cross-platform and is currently available on the Macintosh, MS Windows, UNIX, and OS/2 oper-

ating environments.

4. Discussion

4.1. Mix and Match Strategy

The plethora of component tools and construction methodologies that are emerging provide very flexible methods for creating workstation applications. One can build a workstation using the WWW as the primary information-delivery model, and include extensions which access remote databases, or execute Java or CORBA components. Java can be extended to access CORBA components. CORBA components can be constructed that access information using WWW pro-

ocols and interpret HTML, or incorporate the Java run-time engine.

Any given application might be constructed of components based on any and all of these technologies, since gateways can be constructed on so many axes.

4.2. Standardization Efforts

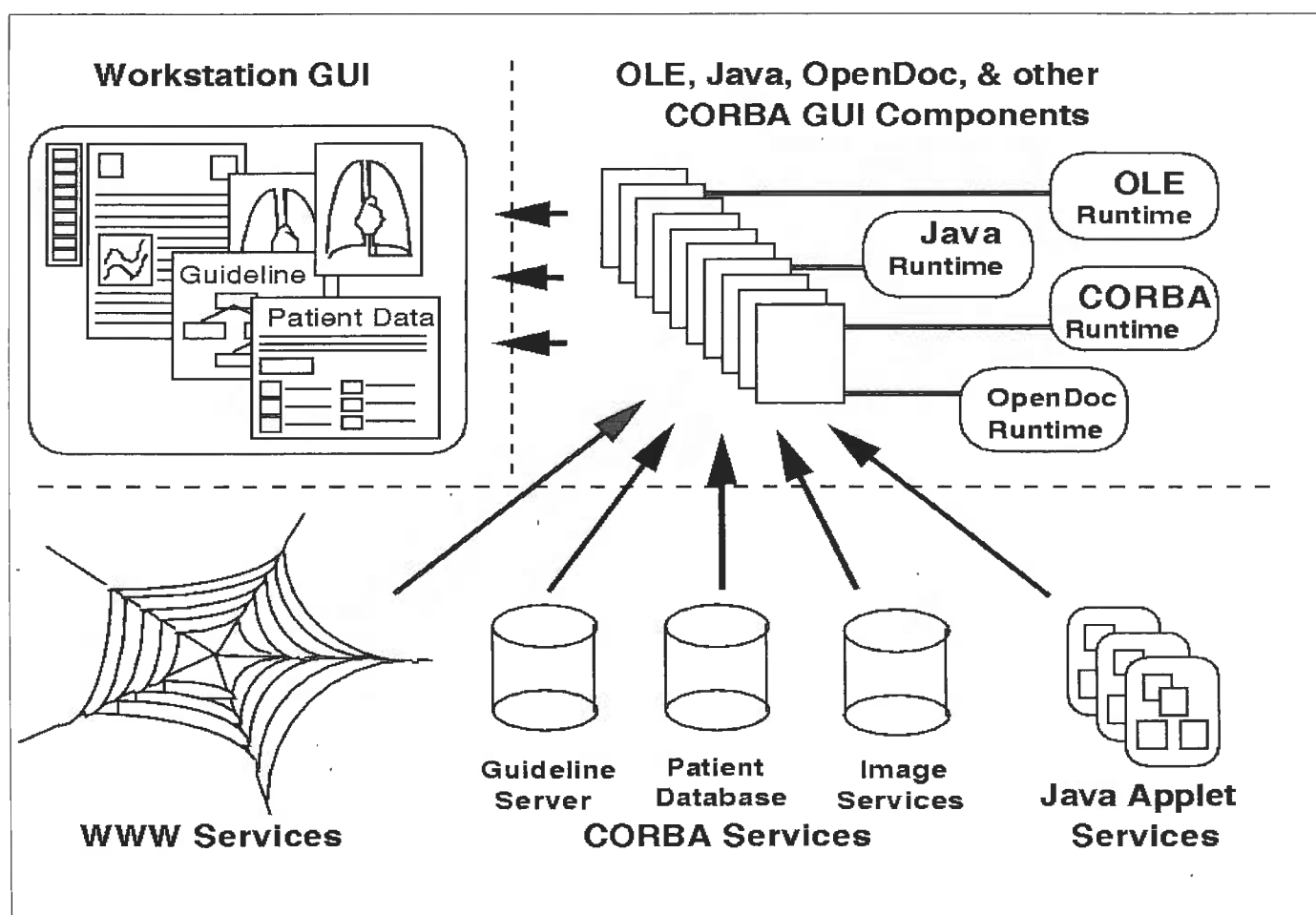
The goal of this work, component integration, can be achieved in a variety of ways, as we have discussed. The best way to do this is not settled, since there is so much flexibility provided by CORBA, Java, WWW, and visual embedding methodologies.

Intrinsic to progress, however, is reusability. If one implementor develops a useful tool or makes available a valuable service, it should be able to

be incorporated into applications wherever needed, and be independent of the integration strategy used. This will be best achieved if standards are adopted for classes of components. HL7, DICOM, Z39.50, and other standards for specific classes of resources are already being adopted. Others will evolve for guidelines, expert systems, and other functionality. In a component-based environment, interfaces to components are messages; standards work needs to focus on the syntax and semantics of the messages for specific classes of components.

Another common theme that relates to the ability to integrate diverse componentry in health-care is the use of a common semantics for health-care concepts. If a guideline tool or event processor is to operate on a clini-

Fig. 1: Workstation integration of OLE, Java, OpenDoc, and other CORBA GUI-level components with WWW, CORBA, and Java-based network services.



cal repository, or a physician referral directory is to provide a list of specialists, appropriate for a patient's clinical problem, a mapping is needed between the representations of the clinical findings in the electronic medical record and those used in the guideline logic or in the event-processor's knowledge base or in the referral-directory search process. Further evolution of the Unified Medical Language System (UMLS) [37] and extensions that link it or other derivative vocabulary resources to workstation applications [38] are required.

4.3. Putting it all Together

The appearance of the WWW, Java, CORBA, and related technologies, has shifted the paradigm for a workstation application. The application is the aggregate of the work performed in support of the user. Most of the work, retrieving data, analyzing it, displaying it, entering findings, generating alerts, receiving advice from an expert system, etc., is done by the components. The integrating framework determines the graphical layout of the components that have visual manifestations (not all of them do), and the sequences in which components are invoked, along with parameters that need to be passed in the form of messages to the components. The integrating framework itself may be provided by a component that lives on a distant computer.

Thus, a component-integration tool might allow an application developer to graphically construct a user interface that incorporates OpenDoc and other CORBA components, OLE components, WWW documents, and Java applets into a composite application framework. High-level scripting might be used in addition to integration of non-visual components, including image repository and processing services, patient data repositories, scheduling services, guideline and vocabu-

lary services, and others (see Fig. 1).

The technology is now available to build the health-care information system environment cooperatively. It is not surprising that in recent years a number of consortia and collaborative groups have formed to address this problem, both in the public sector and in industry.

4.4. Economic Implications

In a previous section we listed an extensive and ambitious set of requirements for workstation applications. It is probably fair to say that no existing health-care information system fully provides these capabilities. Furthermore, it is even questionable whether they should be part of such systems. A workstation may perhaps be more effective if it is viewed as independent of particular systems, functioning rather as a means for accessing health-care information systems as well as other resources, and integrating them. The following industry relationships can be considered:

4.4.1 Single-vendor Complete System

With this approach, a health-care information system is made available by a vendor along with specific workstations tied to that system. The advantage is that all components work well together, because of the unified architecture and central responsibility. Disadvantages are that it makes it more difficult to take advantage of "best of breed" components developed by third parties, and extension and modification are dependent on vendor motivation to provide them.

4.4.2 Multiple Distributed Systems

In this model, a workstation may be used to interact with multiple systems provided by different vendors, but the systems themselves are not integrated. This is somewhat typical of the situation today, in which a vendor HIS may

be accessed via the same workstation able to run local teaching programs and CD-ROM-based electronic textbooks, and on which the Internet can be used for E-mail and WWW access, and MEDLINE searches can be performed.

4.4.3 Component and Integrator Marketplace

Pursuit of the component-integration model opens up a variety of new economic opportunities, although this marketplace is not yet well established. Opportunities exist at a number of levels, as component tool providers, as content service providers, and as integrators of specific applications. The robustness of the model is based on the ease of adaptation to changing needs, extensibility as technology advances, reusability of components, and enabling of cooperative work.

4.5. Other Issues

New complexities arise in the component-integration model. Responsibility for an application is distributed, complicating contractual relationships, liability, issues of ownership, and security. The evolution from existing systems to this new approach needs to be addressed. Legacy systems will not disappear quickly, but will gradually migrate to back-end servers, as they are encapsulated as components.

Another important challenge is to learn how to adapt an application to a user's needs. This is facilitated by component-integration strategies because it can separate the visual presentation aspects from the communication and processing aspects of an application's work. To be effective, an application will need to provide a "point of view" that supports the user's work. As we stated earlier, we predict that a principal direction for this in the future will be approaches that foster the incorporation of clinical guidelines and problem-oriented views as

frameworks for supporting the work of the health-care professional.

Next generation workstation applications will need to integrate resources for clinical practice, education, and decision support, in ways such as described above. The changes in health-care require this capability. Fortunately, technology is now able to provide it.

Acknowledgment: Supported in part by grant #LM 07037 and contract #N01-LM-4-3512 from the National Library of Medicine, and by contracts #N66001-94-D-6052, #N66001-95-D-6007, and #MDA972-94-3-0047 from the Advanced Research Projects Agency, Department of Defense.

References

Note: Technical references are currently best found on the World Wide Web. WWW pointers are listed for some of the following citations, where less current print citations or no print citations exist.

1. Ball MJ, Silva JS, Douglas JV, Degoulet P, Kaihara S (eds.) The Health-care Professional Workstation. Proc of IMIA Working Conf, Washington, DC, June, 1993; Int J Biomed Comput 1994;34:414.
2. Bodenheimer T, Grumbach K. The reconfiguration of US medicine. JAMA 1995;274:85-90.
3. Frenkel M. Health financing policy after the 1994 Clinton plan experience. J Health-care Finance 1995; 21(3):68-71.
4. Gold MR, Hurley R, Lake T, Ensor T, Berenson R. A national survey of the arrangements managed-care plans make with physicians. N Engl J Med 1995;333:1678-83.
5. Grimes J, Potel M. Software is headed toward object-oriented components. Computer 1995;28:24-5.
6. Lewis TG. Where is client/server software headed? Computer 1995;28:49-55.
7. Bernstein PA. Middleware: A model for distributed system services. Commun ACM 1996;39:86-98.
8. Pancake CM. The promise and the cost of object technology: A five-year forecast. Commun ACM 1995; 38:33-49.
9. Stoufflet PE, Deibel SRA, Traum JH, Greenes RA. A state-transition method of modeling clinical encounters. In: *Proc 1995 AMIA Spring meeting*. 1995;81.
10. Starren J, Friedman C, Johnson SB. The Columbia Integrated Speech Interpretation System (CISIS). In: Gardner RM, ed. *Towards Cost-Effective Clinical Computing. Proceedings of the 19th Symposium on Computer Applications in Medical Care*. Philadelphia: Hanley & Belfus Inc, 1995:985.
11. Poon AD, Fagan LM. PEN-Ivory: The design and evaluation of a pen-based computer-system for structured data entry. In: Ozbolt JG, ed. *Transforming information, changing health-care. Proceedings of the 18th Symposium on Computer Applications in Medical Care*. Philadelphia: Hanley & Belfus Inc, 1994:447-51.
12. Bell DS, Greenes RA. Evaluation of UltraSTAR: Performance of a collaborative structured data entry system. In: Ozbolt JG, ed. *Transforming information, changing health-care. Proceedings of the 18th Symposium on Computer Applications in Medical Care*. Philadelphia: Hanley & Belfus Inc, 1994:216-21.
13. Giger M, MacMahon H. Image processing and computer-aided detection and diagnosis. Imaging and Information Management: Computer Systems for a Changing Health-care Environment. In: Greenes RA, Bauman RA, eds. *Radiol Clinics of North America* 1996;34:in press.
14. Goldberg MA. Teleradiology and telemedicine. Computer Systems for a Changing Health-care Environment. In: Greenes RA, Bauman RA, eds. *Radiol Clinics of North America* 1996;34:in press.
15. Berners-Lee T, Cailliau R, Luotonen A, Nielsen H, Secret A. The World-Wide Web. Commun ACM 1994;37:76-82.
16. Berghel H. The client's side of the World Wide Web. Commun ACM 1996;39:31-40.
17. Lowe HJ, Lomax EC, Polonkey SE. The World Wide Web: A review of an emerging Internet-based technology for the distribution of biomedical information. J Am Med Informatics Assoc 1996;3:1-14.
18. Kohane IS, Greenspun P, Fackler J, Cimino C, Szolovits P. Building national electronic medical record systems via the World Wide Web. J Am Med Informatics Assoc 1996;3:in press. See also: <http://www.emrs.org/medweb>
19. Cimino JJ, Socratous SA, Clayton PD. Internet as clinical information system: Application development using the World Wide Web. J Am Med Informatics Assoc 1995;2:273-84. See also: <http://www.cpmc.columbia.edu/cisdemo>
20. Jagannathan V, Reddy YV, Srinivas K, Karinthe R, Shank R, Reddy S, Almasi G, Davis T, Raman R, Qiu S, Friedman S, Merkin B, Kilkenny M. An overview of the CERC ARTEMIS Project. In: In: Gardner RM, ed. *Towards Cost-Effective Clinical Computing. Proceedings of the 19th Symposium on Computer Applications in Medical Care*. Philadelphia: Hanley & Belfus Inc, 1995:12-6. See also: <http://www.cerc.wvu.edu/nlm/nlm.html>
21. Goldwein J, Wong M, Buhle L. A multimedia radiation oncology information resource on the internet. Proceedings of the American Society for Therapeutic Radiology and Oncology. Int J Radiat Oncol 1994;30:298.
22. Galvin JR, D'Alessandro MP, Erkonen WE, Knutson TA, Lacey DL. The Virtual Hospital: A New Paradigm for Lifelong Learning in Radiology. Radiographics 1994;14:875-79.
23. Mammone GL, Holman BL, Greenes RA, Parker JA, Khorasani R. Inside BrighamRAD: Providing radiology teaching cases on the Internet. Radiographics 1995;15:189-98.
24. Schwartz R. Writing Interoperable Applications with ODBC. See: http://discorp.com/futures/8_22/interop.htm
25. Hammond WE. The role of standards in creating a health information infrastructure. In: Ball MJ, Silva JS, Douglas JV, Degoulet P, Kaihara S, eds. *Proc of IMIA Working Conf. 1993*. Int J Biomed Comput 1994;34:29-44.
26. Horii SC, Prior FW, Bidgood Jr DW, Parisot C, Claeys G. DICOM: An Introduction to the Standard. See: http://www.xray.hmc.psu.edu/dicom/dicom_intro/DICOMIntro.html
27. Ratib O. Multimedia image and data navigation workstation. RSNA'95 Scientific Program. Supplement to Radiology 1995;197(P):546.
28. Turner F. An overview of the Z39.50 information retrieval standard. See: <http://www.nlc-bnc.ca/ifla/VI/5/op/udtop3.htm>
29. Adler RM. Emerging standards for component software. Computer 1995;28:68-77.
30. Bernstein PA. Middleware: A model for distributed system services. Commun ACM 1996;39:86-98.
31. Greenes RA, S Deibel. Clinical image interpretation, review, and education support tool kit. (InfoRad exhibit). RSNA'93 Scientific Program. Supplement to Radiology 1993;449.
32. Chueh HC, Barnett GO. Building a clinical workstation using components and a distributed object services architecture. See <http://www.lcs.mgh.harvard.edu/>
33. Deibel SRA, Greenes RA. An infrastructure for the development of health-care information systems from distributed components. J Amer Soc Inform Sci 1995; 46:765-71. See also: <http://dsg.harvard.edu/>

34. Arkad K, Allfeldt H, Gao X, Shahsavari N, Wigertz O, Francois-Choristophe J, Degoulet P. Integration of data driven decision support into the HELIOS environment. The Health-care Professional Workstation. In: Ball MJ, Silva JS, Douglas JV, Degoulet P, Kaihara S, eds. *Proc of IMIA Working Conf 1993*. Int J Biomed Comput 1994;34:195-205.
35. van Mulligen EM, Timmers T, Brand J, Cornet R, Van den Heuvel F, Kalshoven M, van Bommel JH. HERMES: A health-care workstation integration architecture. The Health-care Professional Workstation. In: Ball MJ, Silva JS, Douglas JV, Degoulet P, Kaihara S, eds. *Proc of IMIA Working Conf 1993*. Int J Biomed Comput 1994;34:267-75.
36. Gosling J, McGilton H. The Java(tm) language environment: A white paper. See: <http://java.sun.com/whitePaper/java-whitepaper-1.html>
37. Lindberg DAB, Humphreys BL, McCray AT. The Unified Medical Language System. Meth Inform Med 1993;32:281-91.
38. Cimino JJ. Vocabulary and health-care information technology: State of the art. J Am Soc Inform Sci 1995;46:779-82.

Address of the authors:
 Dr. R.A. Greenes,
 Decision Systems Group,
 Brigham and Women's Hospital,
 75 Francis Street, Boston, MA 02115,
 USA.
 E-mail: greenes@harvard.edu