E.M. van Mulligen[1,2], T. Timmers[3]

[1]Dept of Medical Informatics, Erasmus University Rotterdam,
[2]Netherlands Institute of Health Sciences,
[3]Dept of Medical Informatics, Academic Medical Centre Amsterdam, The Netherlands

Review Paper

Trends in Integrated Clinical Workstations

Abstract: During the last decade, several projects aiming at integrated clinical workstations have been described and several prototypes have been demonstrated. In most of these projects, the clinical workstation accesses information and functionality provided by the present proprietary legacy systems of health-care institutions. We discuss trends in integrated clinical workstations from the viewpoints of software engineering and integration, considering that the clinical workstation itself basically consists of three layers: a presentation layer, a data integration layer, and a communication layer. The software engineering view on clinical workstations focuses on the development of basic building blocks from which clinical workstations, specific to a particular medical domain, can be composed. The integration view on clinical workstations addresses methods and techniques to deal with the, in general, intrinsically closed information systems in health-care institutions.

Keywords: Health-care Professional Workstations, Software Engineering, Information Architecture, Integration, Brokering.

Introduction

The introduction of a new IMIA workgroup on Health Professional Workstations in Kyoto in September 1993 shows that the clinical workstation concept has gained importance. The potential of low-priced graphical workstations connected in a worldwide information infrastructure has been clearly demonstrated by the exponential growth of the World-Wide Web and other network services. The vision of IMIA and other organizations and professionals is that an intelligent workstation will be an essential component for the realization of a worldwide health-care information infrastructure that supports health-care professionals in delivering high-quality health-care and conducting research.

Many hospitals are confronted with enormous pressure to improve the quality of health-care while reducing its costs. The needs for clinical information, the strategic role of clinical data systems, and the importance of accessible, well-managed clinical data are a logical consequence of this pressure and the growing competition between health-care institutions. Research and development programs such as Advanced Informatics in Medicine (AIM), Esprit, and High Performance Computing and Communications (HPCC) have stimulated interest in trying to use information systems more effectively for clinical applications, not just for financial, administrative, and budgetary functions. More and more, patient data are stored in different information systems; the first clinical sites are starting to routinely store biosignals and images in these systems; the monolithic information system is transformed into a diverse range of information systems running on different platforms with different operating systems. Lost in this information cyberspace, and facing the problem of an expanding biomedical knowledge base under an increasing amount of professional time, the health-care professional is challenged to both improve the quality of health-care and reduce health-care costs. The evolving clinical workstation of the 1990s has extraordinary technical potential for satisfying the needs of the health-care professional and for providing a set of tools that enable ubiquitous access to all patient data and decision-support facilities.

One of the components of a clinical workstation is data entry to support the creation, editing, and browsing of the computer-based patient record. This patient record may be scattered over different information systems provided by different vendors, and the task of the clinical workstation is to integrate multimodal data from all these information systems without bothering the health-care professional with the location of the data and the typical interfaces of those information systems. The integrated clinical workstation is, so to say, the focal point of all patient-related data. Presentation of patient data on the clinical workstation includes display of images and biosignals. Another component in the workstation supports the clinician with graphical presentation of trend data, comparing a patient record with data of reference patient groups, etc. Retrieval of on-line literature relevant for decision-making, the consultation of on-line decision-support systems, and selection of similar patient records are included in a decision-making component. An additional component in the workstation provides the clinician with protocols and medical guidelines applicable to the particular workstation domain. The clinical data analysis components support the health-care professional with conclusions and directions based on statistical evaluation of sets of patient records.

This user interface hides the differences between the user interfaces of the underlying data sources. The advent of low-priced graphical workstations has apparently created new possibilities for this type of research. Research issues in this area deal with task analysis and conceptual modeling. The conceptual model specifies what operations and functions a clinician wants to apply to what data, taking into account the specificities of the clinical domain. Task analysis provides a more intelligent approach to matching the actions the clinician performed with his intended end goals.

Research related to the data-model layer focuses on the "computer-based patient record". In the clinical workstation context, this layer provides a global model that links data from different sources. Differences in data representation can be taken care of in this layer.

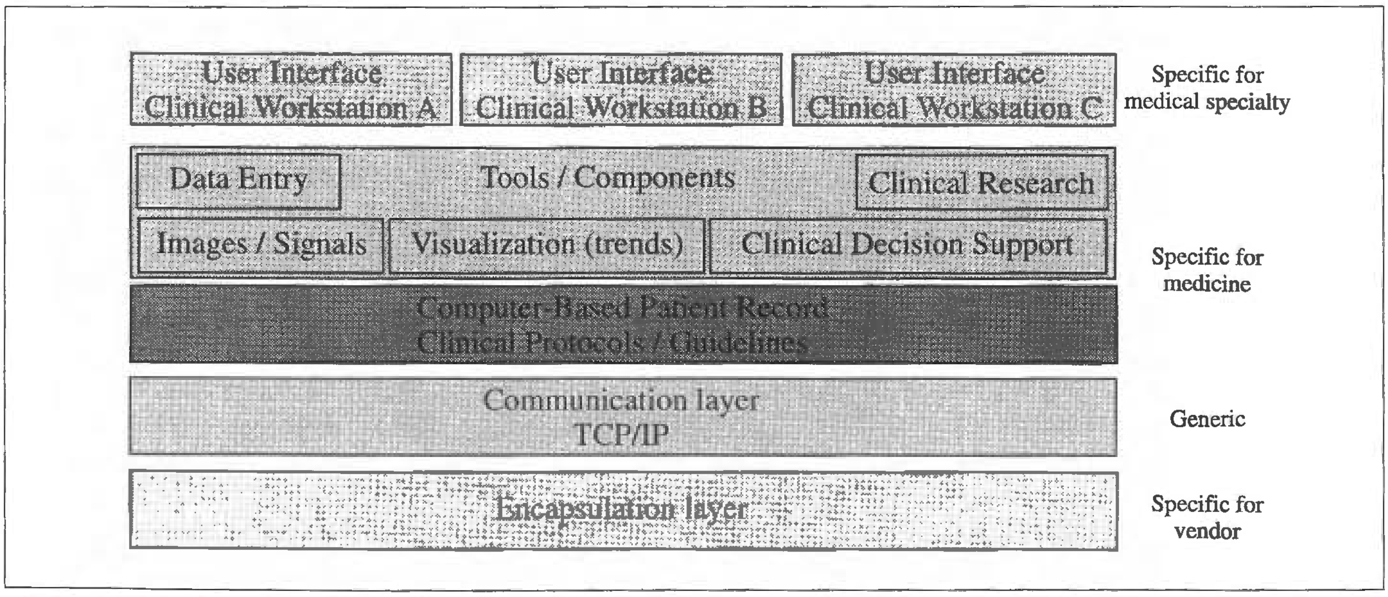
The Integrated Clinical Workstation Concept

During the last decade, several medical informatics researchers have addressed the clinical workstation concept. From a software engineering point of view, the clinical workstation can be laid out in a number of layers: a *user interface layer* that shows data in a uniform and clinically acceptable way, a *data model layer* that basically defines how data from different systems interrelate, and a *communication layer* that provides "transparent" access from the workstation to a number of network resources (see Fig. 1).

The Communication Layer

Despite the maturation of network-technology during the last decade and the resulting establishment of network communication standards, the communication layer still contains many ad-hoc solutions and requires substantial additional research. The communication layer is responsible for access to the often proprietary systems in a clinical computing environment. This access includes retrieval (and storage) of clinical data in clinical information systems, remote execution of functions and procedures (i.e., in a legacy system), and access to remote knowledge sources. The com-

Fig. 1. Layering approach of a clinical workstation. The user interfaces are specific for a particular medical specialty, whereas the tools and components can be shared between a number of clinical specialties. The communication layer of a clinical workstation can be largely based on products that are commercially available. The encapsulation layer (embedding of legacy systems) is specific for each of the proprietary systems to be embedded.



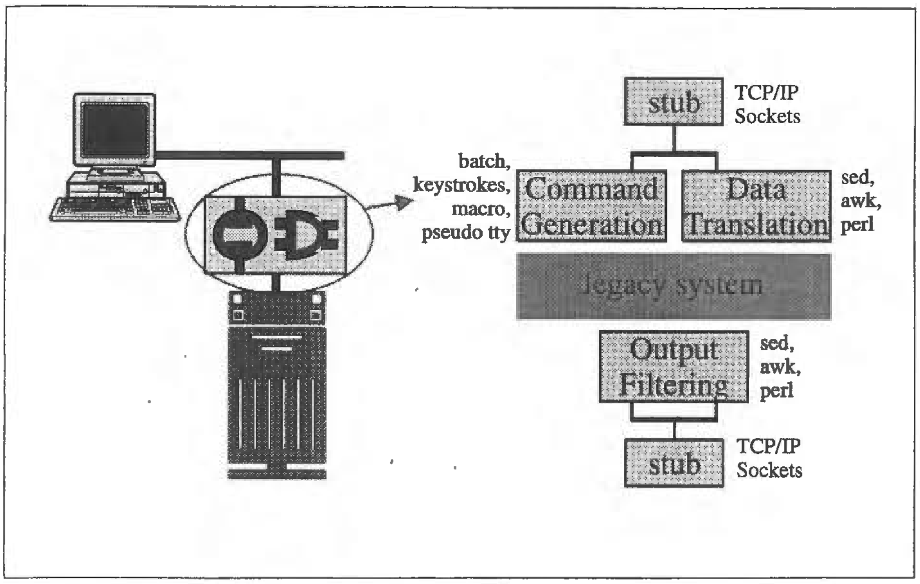


Fig. 2. Schematic overview of an encapsulator as developed in the HERMES project. The encapsulator hides the proprietary features from the clinical workstation. The stub reads the incoming (network) requests, analyzes them, and activates the command-generation and data-translation modules. The command-generation modules will generate for each request an equivalence of instructions specific for the underlying legacy system. The data translation module will translate any data present in the request into the data format used by the legacy system. Any data generated by the legacy system is fed to the output filtering module, which will generate data in the format used by the clinical workstation. The stub will finally include the data in a reply to the clinical workstation. Several commercial or public-domain tools exist for the development and implementation of the three encapsulator modules.

mon interest in client-server solutions for information systems of office automation companies has stimulated the development of standards for data communication. Many database management system vendors conform nowadays to the Open Database Connectivity (ODBC) standard. However, the typical proprietary clinical information systems still use a dedicated database management system that, in general, does not comply with these industrial standards. The ODBC standard specifies how client and server can exchange queries and data, and it assumes that the client "knows" the database schema of the server. For data exchange between information systems of different health-care institutions, this assumption is often not true. As a consequence, standardization efforts in health-care tend to focus more on the description of data exchange messages [1-3]. In Health Level 7 (HL7), for instance, several of these messages have been defined, explicitly stating what data are contained in a message and what these data mean. The chunks of data contained in a message have been defined according to the specific needs of health-care personnel. New efforts in the direction of data-exchange standards explore the possibilities of object-orientation as a part of the message protocol [4-6].

Most proprietary information systems in health-care do not provide client-server access. Due to the fact that instantaneous replacement of these systems with open counterparts would have a large impact on the resources and organization of health-care institutions, alternative approaches have been investigated. Several projects developed software layers that wrap legacy systems [7-9]. These encapsulators combine client-server access for clinical workstations with the traditional input-output mechanisms to proprietary systems. The encapsulators translate both data and requests between the standards evolving at the workstation side and the specific data and instruction formats of the legacy systems (see Fig. 2).

The Data Integration Layer

The data integration layer supports a global data schema that relates patient data from different existing information systems. This global data schema can be seen as the standard view on all patient data that are available in the network. It is the task of the communication layer to manipulate the data from the legacy systems so that they fit into this data schema. Based on this global data schema, a new generation of computer-based patient records (CPR's) is built. These advanced CPRs contain mechanisms to relate data from different sources and incorporate medical semantics that are relevant for clinical structured data entry [10-15]. This CPR layer, including its semantics, makes the development of clinical workstation applications more straightforward.

Current research in database management systems also focuses on database schema integration and semantic modeling. Research in database schema integration is being done on the development of algebras for expressing the relationships between schema elements from different databases. Some of the semantic differences between database schemas can be resolved using this kind of approach [16,17]. One of the difficult problems to solve in database integration is unification of different patient identification systems. Research has been devoted to extensive matching algorithms that exploit demographic information to correctly link records from different databases in the absence of an overall patient identification system. Concerning semantic modeling, the object-orientation of database schemas is seen as an important step towards semantic modeling [18-20]. Although several commercial object-oriented DBMSs are available nowadays, their performance is still considered to be inadequate for routine use. With object-orientation, relationships between data can be made more explicit than in the usual relational database schema.

The User Interface Layer

The user-interface layer provides the clinician with a presentation of all data and functions that are provided through the workstation. Basically, two issues are important in the presentation of data and functions: the visualization technique and desktop metaphor, and the conceptual model. The visualization technique exploits a particular way of presenting information and functionality to the user. With the advent of cheap workstations with powerful graphical capabilities, the visualization technique is not limited to 80x24 character text screens. Data can be visualized in a window using graphics (histograms, line-charts) and functions can be presented on the screen simply by writing their name and activated by pointing at them with a mouse. In order to make intuitive how data and functions are related, metaphors are used. A metaphor is a natural concept from our common experience that can be easily recognized, understood, and remembered [21]. Two well-known metaphors are the WIMP (Window-Icon-Menu-Pointer) desktop metaphor and the hypertext metaphor.

The WIMP desktop metaphor regards the screen as a representation of the surface of a desk, and windows are viewed as paper documents. The mouse is used to point at data and make selections from a menu of functions. The organization and naming of functions corresponds with the idea of a window being a sheet of paper. Due to the fact that a computer screen is still smaller than the surface of a desktop, functions have been added that shrink a window to a small image, an icon, and that size it to the full size of the screen.

The hypertext metaphor as provided in world-wide-web browsers, visualizes data as a network of linked pieces of text and images. Phrases in the text are marked as containing a link to another document. Upon selecting that text, the associated text is retrieved. Several research projects explore the capabilities of the world-wide-web's browser hypertext interface for the development of interfaces to clinical vocabularies [22,23], to clinical information systems [24], and to medical guidelines [25]. New developments such as SUN's Java and HotJava enhance the browser of hyperlinked documents with powerful data visualization applications ("applets") that can be used to provide specialized browsers for complex data (e.g., a 3-dimensional browser for molecular structures).

A central research theme, important for clinical workstations, is the assessment of the usability of the mainly 2-dimensional desktop and hypertext metaphor for presentation of 3-dimensional medical data and the development of new metaphors that possibly are better suited for presentation of clinical data [26].

The conceptual model of the user-interface layer specifies what data should be shown in one window and what functions are applicable to what data at what time. Input for the development of the conceptual model are the tasks, work-flow, and roles of clinicians. An example is Weed's problem-oriented model for medicine [27]. This model uses a classification of medical problems and a patient problem-list at the physician's level of understanding. Rector uses a conceptual model for his computer-based patient record application that is separate from the clinical process model [15]. For services such as patient billing, accounting, and personnel records many hospitals use functional models to represent the needs and requirements of the organization.

The first initiatives for clinical workstations came from the clinical information systems arena. After decades of developing centralized systems accessible through large numbers of terminals, the trend has been to develop clinical workstations that provide transparent access to all information systems present [28]. This first generation of integrated clinical workstations aims at so-called desktop integration. In these systems, the workstation desktop consists of a number of windows to the various clinical information systems integrated.

Depending on the intelligence of the user-interface layer, the windows either function as terminal windows to the clinical information systems or as mediators that transform the terminal output into a uniform format. The first phase of the Integrated Academic Information Management System (IAIMS) project is an example of a system exploiting integration at the user interface level [29].

Several Approaches to an Integrated Clinical Workstation

The nature of an integrated clinical workstation has challenged computer scientists to test new software-engineering principles [7,8,30]. The most widespread concept is to construct a clinical workstation from a set of standardized building blocks or components [31,32]. These components represent independent blocks of functionality that can be used in several different clinical settings. In a clinical workstation, for instance, a basic building block could be the diagnosis component, the medication component, or the graphical presentation component. Each component provides an application programming interface (API) that makes a number of its functions accessible for other components (see Fig. 3). A component can in its turn call functions from other components. A medication component could, for instance, provide an API function called "Prescribe medication" which in its turn could call the "Print" API function from the print-manager component. In this notion, an application (or tool) is nothing more than a program that arranges the flow between a number of components. When the communication between components is arranged using (standard) network communication facilities, the components can be dispersed in the network. This facilitates the sharing of expensive system resources by a number of clinical workstations. The hypothesis behind this software-engineering approach is that components can be reused for different clinical applications and that this component-based software construction is less awkward than writing the application from scratch.

Communication between components is organized according to the client-server model [33]. In the client-server model, the application that initiates communication is called the client whereas the application that responds to the request is called the server. A server can in its turn initiate a request for another server and thus become a client. Apart from the network layers that are essential to establish a reliable communication channel, client and server have to share a common message protocol for data exchange and expressing a request. HL7 and EDIFACT are the best known examples of standard message protocols [1,3]. However, they do not cover the whole domain and do not provide messages for communication between components. Several standard message syntaxes exist, ASN-1 being the most prominent [4].

The component approach with its APIs is very similar to the object-oriented approach. One can consider a component as an implementation of an object class that holds one or more instances of that class. The API functions correspond with the public methods of the object class. It is, therefore, not surprising that the object model has been exploited to implement component-based architectures. OMG's definition of the Object Request Broker (ORB) [34], Microsoft's OLE2.x implementation [35], the HELIOS architecture [36], and the HERMES architecture [8] all adhere to the object model for implementation of their component-based applications.

In component-based architectures, it is important to have locator or brokerage facilities. A locator or broker keeps track of all the APIs provided by the components and is able to bind a request from a client to a particular server component. When there is a large number of components dispersed in a network without locator or brokerage facilities, replacing components or adding new components is an immense task since all components have to be updated with the new component identifiers. In a broker or locator approach, the new component can be added "on the fly" by editing its entry in the broker's directory. This flexible approach to component-based architectures is especially important if existing legacy systems are gradually replaced by open counterparts [37]. When introducing state-of-the-art information systems, it would be much preferred if the only action to switch clients to these new systems would be the editing of the broker database.

Challenges

Since industry is gradually providing support for the development of integrated clinical workstations, medical-informatics researchers should focus on new challenges in this field. One of the important issues remains a reference architecture for clinical workstations. This architecture could specify what the basic components are that are considered to be the building blocks. The mechanism to combine components into a clinical workstation environment should follow the industrial standards (OMG's ORB [34], OSF's DCE [38], and Microsoft's OLE [35]) and employ available communication syntax standards for exchange of data (ASN.1) and use already defined message standards (HL7, EDIFACT). Having this reference architecture, inter-institutional exchange of components could be promoted. Subsequently, research could be initiated to design a "standard" open computer-based patient record architecture that promotes the construction of a world-wide health-care information network. Typical research issues of this development are related to security and privacy matters. Another important role for medical informatics researchers is to quantitatively assess the benefits of integrated clinical workstation for health-care and use that assessment as input to the development of components that support the clinician with his medical decision-making tasks.

References

1. Health Level Seven: An Application Protocol for Electronic Data Interchange in Healthcare Environments. Version 2.1.
2. Draft DICOM Standard, PS-3 through PS-11, ACR/NEMA, 1992.
3. ISO 9735 (1988): Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) - Application Level Syntax.
4. ISO/IEC 8824: Information Technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).
5. Draft P1157.1 Standard for Healthcare Data Interchange - Information Model Methods, IEEE
6. Seliger R. Mapping HL7 Version 2.x to

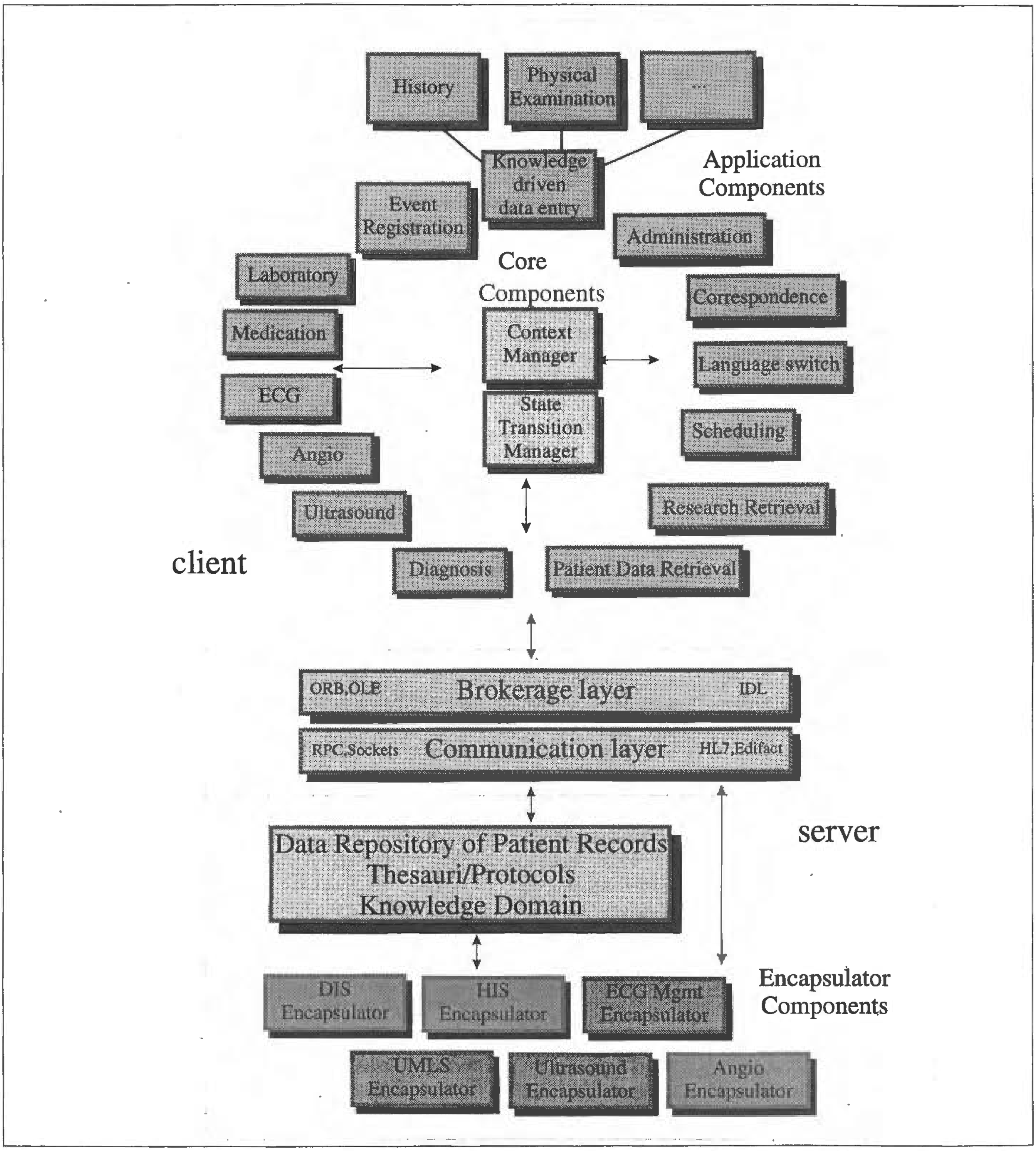


Fig. 3. An example of a component-based architecture. This Figure shows the architecture for a multi-media workstation for cardiology (I4C) as being developed in the 4th framework programme of AIM.

OLE and CORBA. Work-In-Progress. July, 1995. Hewlett-Packard Medical Products Group, Andover, United States of America.

7. Fournier F. The MINI software factory: development of kernel mechanisms around process modeling and software bus techniques. J Systems Integration 1992;2:145-80.

8. Van Mulligen EM, Timmers T, Van Bemmel JH. A new architecture for integration of heterogeneous software components. Methods of Information in Medicine 1993;32:292-301

9. Lemaitre D, Sauquet D, Fofol L, Tanguy L, Jean F-C, Degoulet P. Legacy systems: managing evolution through integration in a distributed and object-oriented comput-

ing environment. In: Gardner RM, ed. *Towards Cost-Effective Clinical Computing. Proceedings of the 19th Symposium on Computer Applications in Medical Care.* Philadelphia: Hanley & Belfus Inc, 1995:132-6.
10. Moorman PW, van Ginneken AM, van der Lei J, van Bemmel JH. A model for structured data entry based on explicit descriptional knowledge. Meth Inform Med 1994;33:454-63.
11. Van Ginneken AM, Stam H, Moorman PW. A multi-strategy approach for medical records of specialists. In: Greenes RA, Peterson HE, Protti DJ, eds. *Proceedings of the 8th World Congress on Medical Informatics. MEDINFO 95.* Amsterdam: North-Holland, 1995:290-4.
12. Dick RS, Steen EB, eds. The Computer-Based Patient Record. An Essential Technology for Health Care. Washington DC: Institute of Medicine, 1991:15-9.
13. Rind DM, Safran C. Real and imagined barriers to an electronic medical record. In: Safran C, ed. *Proceedings of the Seventeenth SCAMC.* New York: McGraw-Hill, 1993:74-8.
14. Griffith SM, Kalra D, Lloyd DSL, Ingram D. A Portable Communicative Architecture for Electronic Healthcare records: The Good European Healthcare Record Project. In: Greenes RA, Peterson HE, Protti DJ, eds. *Proceedings of the 8th World Congress on Medical Informatics. MEDINFO 95.* Amsterdam: North-Holland, 1995:223-6.
15. Rector AL, Nowlan WA, Kay S. Foundations for an electronic record. Meth Inform Med 1991;30:179-86.
16. Raijnikanth M, Jakobson G, Lafond C, Papp W, Piatetsky-Shapiro G. Multiple Database Integration in Calida: design and implementation. In: Ng P, Ramamoorthy CV, Seifert LC, Yeh RT, eds. *Proceedings of the First International Conference on Systems Integration.* Los Alamitos: IEEE Computer Society Press 1990, 378-84.
17. Landers T, Rosenberg R. *An Overview of Multibase. Distributed databases.* Amsterdam: North-Holland Publishing, 1982:153-83.
18. Codd EF. Extending the database relational model to capture more meaning. ACM T Database Syst 1979;4:397-434.
19. Nguyen GT, Rieu D. Schema evolution in object-oriented database systems. Data and Knowledge Engineering 1989;4:43-67.
20. Kilov H. From Semantic to Object-Oriented Data Modeling. In: Ng P, Ramamoorthy CV, Seifert LC, Yeh RT, eds. *Proceedings of the First International Conference on Systems Integration.* Los Alamitos: IEEE Computer Society Press. 1990:384-93.
21. Esterhay RJ. User metaphors for health care professional workstations. Int J Biomed Comput 1994;34:95-113.
22. McCray AT, Razi A. The UMLS Knowledge Source Server. In: Greenes RA, Peterson HE, Protti DJ, eds. In: Greenes RA, Peterson HE, Protti DJ, eds. *Proceedings of the 8th World Congress on Medical Informatics. MEDINFO 95.* Amsterdam: North-Holland, 1995:141-5.
23. Cimino JJ, Socratous SA, Grewal R. The informatics Superhighway: prototyping on the World-Wide Web. In: Gardner RM, ed. *Towards Cost-Effective Clinical Computing. Proceedings of the 19th Symposium on Computer Applications in Medical Care.* Philadelphia: Hanley & Belfus Inc, 1995:111-5.
24. Hinds A, Greenspan P, Kohane IS. WHAM! A forms constructor for medical record access via the World Wide Web. In: Gardner RM, ed. *Towards Cost-Effective Clinical Computing. Proceedings of the 19th Symposium on Computer Applications in Medical Care.* Philadelphia: Hanley & Belfus Inc, 1995:116-20.
25. Barnes M, Barnett GO. An architecture for a distributed guideline server. In: Gardner RM, ed. *Towards Cost-Effective Clinical Computing. Proceedings of the 19th Symposium on Computer Applications in Medical Care.* Philadelphia: Hanley & Belfus Inc, 1995:233-7.
26. Robertson GG, Card SK, Mackinlay JD. Information visualization using 3D interactive animation. Commun ACM 1993;36:57-71.
27. Weed LL. *Knowledge Coupling: New Premises and New Tools for Medical Care and Education.* New York: Springer-Verlag, 1991.
28. Young CY, Tang PC, Annevelink J. An Open Systems Architecture for Development of a Physician's Workstation. In: Clayton PD, ed. *Proceedings of the 15th Annual Symposium on Computer Applications in Medical Care.* New York: McGraw-Hill Inc. 1991:491-5.
29. Matheson NW, Cooper JAD. Academic information in the academic health sciences center: roles for the library in information management. J Med Educ 1982;57:1-93.
30. Jean F-C, Jaulent MC, Coignard J, Degoulet P. Distribution and communication in Software Engineering Environments. Application to the HELIOS Software Bus. In: Clayton PD, ed. *Proceedings of the 15th Annual Symposium on Computer Applications in Medical Care.* New York: McGraw-Hill Inc. 1991:506-10.
31. Van Mulligen EM, Timmers T, Van den Heuvel F, Van Bemmel JH. A prototype integrated medical workstation environment. Comput Methods Programs Biomed 1993;39:331-41.
32. Greenes RA. A "building block" approach to application development for education and decision support in radiology: Implications for integrated Clinical Information System Environments. J Dig Imag 1991;4:213-25.
33. Sinha A. Client-server computing. Commun ACM 1992;35:77-97.
34. OMG. The Common Object Request Broker: Architecture and Specification. OMG Document 91.12.1, X/Open publisher.
35. Toohey J. Using OLE 2.x in Application Development. 1994 Que Corporation.
36. Degoulet P, Safran C, Bowers GH. Design and processing issues for the health care professional workstation: summary and recommendations. Int J Biomed Comp 1994:34;241-7.
37. Power LR. Post-facto integration technology: new discipline for an old practice. In: Hg PA, Ramamoorthy CV, Seifert LC, Yeh RT, eds. *Proceedings of the First Conference on Systems Integration.* Los Alamitos: IEEE Computer Society Press, 1990:4-13
38. OSF. Introduction to OSF DCE. Englewood Cliffs, New Jersey: Prentice Hall, 1992.

Address of the first author:
E.M. van Mulligen, Ph.D.,
Department of Medical Informatics,
Erasmus University Rotterdam,
PO Box 1738,
3000 DR Rotterdam,
The Netherlands.