

COMMET 01340

Section II. Systems and programs

## A prototype integrated medical workstation environment

E.M. van Mulligen<sup>a,b</sup>, T. Timmers<sup>a</sup>, F. van den Heuvel<sup>a,c</sup> and J.H. van Bommel<sup>a</sup>

<sup>a</sup> Department of Medical Informatics, Erasmus University Rotterdam, <sup>b</sup> University Hospital Dijkzigt and  
<sup>c</sup> Sophia Children's Hospital, Rotterdam, The Netherlands

In this paper the requirements, design, and implementation of a prototype integrated medical workstation environment are outlined. The aim of the workstation is to provide user-friendly, task-oriented support for clinicians, based on existing software and data. The prototype project has been started to investigate the technical possibilities of graphical user-interfaces, network technology, client-server approaches, and software encapsulation. Experience with the prototype encouraged discussion on both the limitations and the essential features for an integrated medical workstation.

Workstation; Graphical user-interface; Encapsulation

### 1. Introduction

Last decade's information technology is now rapidly ingressing health care. Professionals are challenged to employ personal workstations, graphical user-interfaces, network technology, relational database management systems, and a wide spectrum of data processing applications. However, many clinicians cannot take full advantage of this new information technology, since its use demands considerable operational knowledge. Stated differently, the current approach of information technology is still primarily tool-based, while the professional's focus on this technology tends to be task-oriented. Consequently, most clinicians favor the paper record because it is less time-consuming, certainly easier to use, although systems now provide medical data, sig-

nals, and images. The data stored in most hospital information systems are still foremost intended for administration, billing, scheduling, and resource allocation instead of health care support.

The Medical Workstation project (called HERMES-MW2000) ultimately aims at providing the clinician a personal workstation from which all data, knowledge, and functions can be accessed in a user-friendly way for research and patient care. This workstation should provide the operational knowledge to access the data for various tasks: patient care, clinical research, and education. Starting with existing applications and databases, the workstation acts like a shell around all already existing and future software systems through which clinicians can uniformly address all these data and functions in a network.

In the following sections, the various levels at which integration can be obtained will be outlined. Related research will be discussed according to the levels of integration involved and the ability to deal with existing applications. This is followed by an outline of the integration architec-

---

Correspondence: E.M. van Mulligen, Department of Medical Informatics, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands. Tel.: +31-(0)10-408-7050. email: vanmulligen@mi.fgg.eur.nl.

ture implemented in the workstation and the applications initially selected. The Conclusion section will elaborate on the usability and the necessary modifications of this architecture and ends with an evaluation of both the integration methodology and the prototyping approach.

**2. Levels of integration**

The software process outlined above is often referred to as integration: bringing together different, heterogeneous resources at one location, thereby eliminating the inter-resource differences. The differences can be clustered in different categories of integration. For each category, an abstraction layer can be defined that hides the dissimilarities from the user. For HERMES-MW2000, five categories can be distinguished. For some (parts) of the abstractions, current standards are available and the workstation has adopted these standards as much as possible. The five integration categories constitute a subdivision of the application communication layer of the

OSI/ISO seven-layer network-communication model [1] (Fig. 1). The following integration categories have been discriminated.

*2.1. Hardware integration*

This overall integration category conceals the differences between the computer systems in a network. These differences are most profound in the distinction between local and remote processing, and local and remote file access. Basically, one has to utilize a network protocol for starting remote jobs and accessing remote files. Ideally, a commercially available network protocol should eliminate these distinctions, offering one virtual processor (or processor pool) and one virtual file system. Although some commercial network protocols nowadays do offer file system integration, these protocols are limited to only a small range of hardware. Our integration project adopted a standardized network protocol (Arpa/Berkeley) that in itself also does not offer this abstraction [2]. Therefore, an abstraction layer has been defined which, based on the standard network pro-

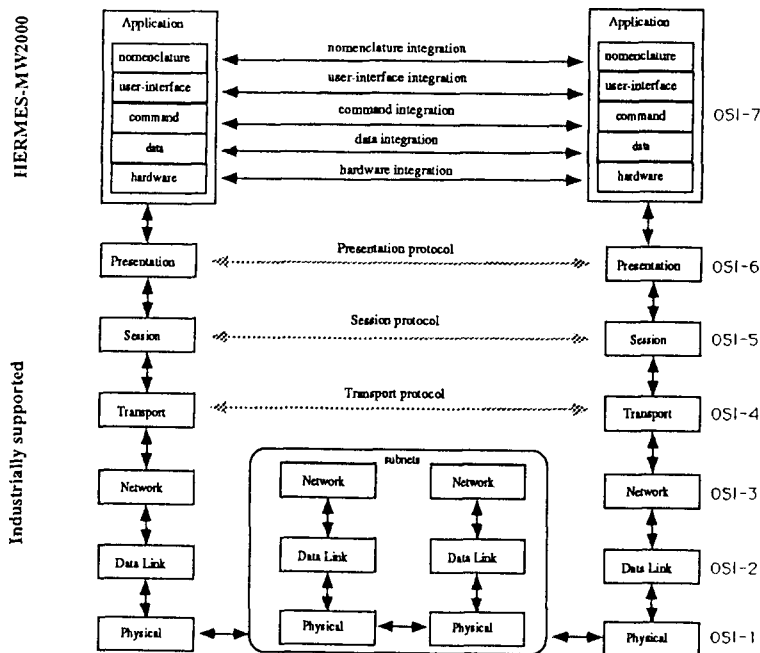


Fig. 1. The Integration Model as extension to the OSI Communication Reference Model. The HERMES-MW2000 defines a communication protocol at application level for the various components of an application.

TOCOL, eliminates these local-remote discrepancies.

## 2.2. Data integration

The second abstraction category deals with the various coexisting data formats. When using a range of applications to solve a particular task, it is necessary to have facilities for exchanging data between the applications. This exchange of data is hampered by the different data formats of the applications and the lack of common data interchange format that can be used by all applications. Therefore, many clinicians need to re-enter the data in the proper format.

In addition, some data are not available as data files but can only be accessed through a database management system. Exchange of data from a database management system to an application requires the database query to be stored as part of the data format. For HERMES-MW2000, a common data storage format has been defined that contains entries for storing database queries. This data format is used as an intermediate storage format (ISF); exchange of data between applications then always goes in two steps: from the source format to ISF and from the latter to the target format. A software data manager takes care of the translation of data between the ISF and the application formats.

## 2.3. Software integration

Each application has its typical command syntax and commands. For a clinician interacting with several applications at the same time, these differences are annoying and confusing. Moreover, for a clinical task one would prefer to issue command sequences rather than individual commands. The software integration abstraction defines one command structure with a unique collection of command sequences, which can be expanded to batch programs or series of keystrokes.

## 2.4. User-interface integration

Each application produces output, either in a file (batch mode) or on screen (interactive mode).

The first achievement of this integration is to abstract from the typical terminals required by the various applications. Each application runs on the same display, independently from the typical terminal or graphics adapter. To support this, the workstation provides terminal emulators that simulate these devices. In addition, the user-interface has to handle camouflage of the different output formats and yield a consistent output format. Difficulties arise when integrating interactive applications. Interpretation of generated device control codes and translation to the general output format is a cumbersome and complex task. In our integration project, the latter abstraction has been limited to batch-oriented applications, while each interactive application runs in a window on one display.

## 2.5. Nomenclature integration

The reduction of inter-application distinctions immediately introduces the danger of semantic misinterpretation. Many terms and headings are interpreted in the (application) context in which they appear and elimination of this visible context may cause errors. The nomenclature abstraction maps the different vocabularies to a general uniform vocabulary, which is preferably standardized. This general vocabulary includes a data dictionary and a medical data model. Such data dictionaries can be user defined.

## 3. Related research

Although much research has been directed towards integration [3], primarily only a subset of all integration categories was covered. Data integration has gained much attention lately; multi-database management systems and distributed database management systems combine hardware integration with data integration [4]. The topics covered in this area are more related to schema integration and query optimization than to abstraction mechanisms. Wiederhold [5] and Greenes [6] focus on access of data from within applications and provide common data abstraction mechanisms such as mediators or daxels (data

access elements), accomplished by adapting existing applications or developing new software. The Helios project [7] aims at network, data, and software integration and employs a software bus as the communication interchange channel among applications; individual applications have to be adapted to connect to this software bus. The main research focus of IAIMS (Integrated Academic Information Management Systems) is the inter-connection of existing systems [8]. The first phase accomplishes a network, connecting various computers and offering clinicians access to all these systems from each workstation (i.e., the first level of user-interface integration). The next phases aim at data integration, software integration, and nomenclature integration and will be mainly achieved in new software [9,10].

#### 4. Development model

The HERMES-MW2000 project aims at facing all categories of integration, starting with existing applications and avoiding any modification of the applications. Critical to this process of integration is the strategy through which it is achieved. Two opposing approaches can be pursued, i.e., a top-down approach starting with a common integration mechanism and working down to applications and data, and a bottom-up approach, starting with applications and constructing hierarchies of abstraction layers around these applications, until one ends up with one top layer. The first approach starts with a sound theoretical mechanism, but the danger is that it lacks a practical reflection. The second method commences with a typical application domain, but there is a danger that the end solution might be constrained to integration of those specific applications only. Both strategies tend to have a breadth-first, or total solution orientation, making corrections awkward and a waste of time. For HERMES-MW2000, a depth-first approach has been utilized by implementing only a limited but representative set of applications according to a general integration mechanism. Deliberate selection of the applications prevents a too specialized and narrow solution, whereas the common integration

mechanism can probably be utilized and tested for other applications as well. In the next sections, both the architecture (integration mechanism) and the applications initially selected will be discussed. The Conclusion section will elaborate on the general usability and the necessary modifications of this architecture. Finally, both the integration methodology and a prototyping approach will be discussed.

#### 5. Architecture

Current integration research mainly focuses on new developments that obey the practice of open-systems technology, network dispersion, and modularization. Unfortunately for integration, the medical milieu already has a vast amount of software, data, and knowledge resources available in closed, often isolated systems that cannot fulfil new software requirements. The advent of powerful workstations enhances the possibilities to bridge this gap between what is offered and what is desired. The architecture of the HERMES-MW2000 enhances existing applications with the possibility to cooperate with new applications that satisfy criteria of openness, network dispersion, and modularization. In addition, the architecture can easily be extended with new (existing) applications.

Our integration project has been arranged around a client-server mechanism. The client, a graphical user-interface, sends high-level commands to a server. These commands are stored in a database, together with associated parameters, a full (medical) name and a specification of what groups of users are entitled to use it. When activated, the command is transmitted to the server, and the server will properly address the application.

In order to address existing applications that are not immediately able to respond to the high-level commands and cannot utilize HERMES-MW2000's ISF data files, two facilities were added: a data translation facility (DTF) and a command generation facility (CGF). The DTF translates data from the workstation's ISF to the application's format and vice versa, and the CGF

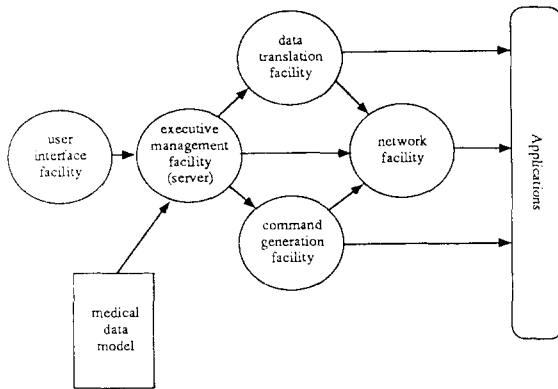


Fig. 2. Overview of the architecture of the prototype HERMES-MW2000. Via the user-interface facility, a user can order a function and activate the executive management facility to solve the request. Subsequently, the executive management facility (EMF) will activate the data-translation facility and the command-generation facility to create a data file in the proper format and a command file with the correct instructions. Via the network facility, this is sent to remote computers and the application is activated. The medical data model is used to translate descriptive names of data attributes to the different attribute names used in the various applications.

expands a macro command to a series of keystrokes or batch-commands for the application. Both facilities employ the network facility (NF) to address files on distant computers and to start applications remotely. Both DTF and CGF encapsulate an existing application, equip it with a standard interchange interface and free the outside world, i.e., the user, from its internal details (Fig. 2).

The medical data model can be used by all applications (or encapsulators) to provide the user (and the application) with a standard vocabulary. The workstation's data model has been organized according to a relational structure in which information about each individual attribute can be stored. This information includes a data dictionary, code lists (and their associated meaning), and descriptive names for the attributes. The model has no entries for specifying a computational relation between two attributes, nor has it synonyms. The pretty names can be changed independently by the users, and permit a first level of nomenclature abstraction.

### 5.1. Applications

The prototype HERMES-MW2000 has mainly been accomplished with already existing applications. A representative range of different applications was selected to test the client-server concept, the DTF and CGF encapsulation, and the ISF definition. Part of the applications may reside on remote computers, linked with the workstation by way of the NF. The idea of a medical data model has been tested in cooperation with existing applications.

### 5.2. Databases

The workstation has been developed primarily with the aim to support clinical research. The large data sets involved in clinical research and the preferred interactive response, led to the requirement of a rather static clinical research database on the workstation, for which INGRES was selected. During its analysis, a research database should not be changed and no new data should be added (static), in contrast to a clinical database that is continuously updated. Before the analysis, data are most often transferred and mapped from dynamic clinical databases to this static research database. During this data transfer process, the data dictionary part of the medical data model is automatically constructed. The user is prompted to specify the codelists and to identify the keys. All other information is obtained from the data transfer. Currently, data can be imported from a Hospital Information System, from the cardiology departmental information system, from a general practitioner's information system (such as ELIAS [11]) and from widely known systems as dbaseIII, Lotus 123 or SPSS. SQL-based databases can also be imported directly into the research DBMS. For the prototype, eight clinical databases were constructed on the HERMES-MW2000 for different research projects.

### 5.3. Analysis

The statistical package BMDP has for a major part been encapsulated in the HERMES-

MW2000 [12], while it is operational on a different computer in the network. BMDP is a batch-oriented application that has an extensive and very specific command language. Its output is written into a file and transformed into graphical output (using OSF/Motif) by HERMES-MW2000. BMDP's descriptive statistical modules and survival-analysis module are currently operational in this way. Additional statistical modules cover the graphical exploitation of cross-tables and confidence intervals with *P*-values. The former module has been especially developed for the HERMES-MW2000 users and directly accepts high-level commands and ISF data files, rendering encapsulation superfluous. The latter was developed by our Department of Epidemiology and has been encapsulated.

#### 5.4. Presentation

The interactive use of the HERMES-MW2000 demands graphical presentation of results. Two approaches have been followed: utilization of an existing application and specific innovative developments. The existing application selected for graphical presentation is Harvard Graphics. In an MS-DOS emulator, this interactive application has been made accessible in the HERMES-MW2000. The CGF expands macros such as for histograms and pie-charts to the keystrokes necessary for Harvard Graphics, even when the application runs under MS-DOS, and the CGF under UNIX. The second approach has been employed for the data inspection module, which includes histograms and tables.

#### 5.5. Images and signals

Besides ASCII data, the prototype also contains facilities for displaying 2-D or 3-D images (CTs, MRIs, angiograms, nuclear images) and signals (ECGs). The integration of these modules has been limited to the user-interface level: on each workstation these facilities can be called and the output is shown in a window. All facilities have adopted the OSF/Motif user-interface.

## 6. Implementation

Each of the integration layers is based upon a regular interchange definition for data and commands. These definitions are the common interface for the applications and provide a means for uniformly accessing data, knowledge, and functions.

### 6.1. Intermediate storage format

The data format has been defined under the name internal storage format (ISF). This format consists of three files, one containing an SQL command, another containing the format itself, the number of variables and the variable names in the data model, and the third file containing the actual data. The command language contains a set of macro commands together with parameters. This set of macro commands can be extended with new commands; its parameter-passing mechanism complies with the UNIX convention.

Each existing application is encapsulated by two drivers: a data translation facility and a command generator. These drivers are activated by the HERMES-MW2000 to transform the data from the ISF format to the application format and vice versa, and to expand a macro command into a series of keystrokes or commands for the application. Activation of a macro command automatically activates the data translation driver and the command generator, which in their turn activate the application. The network layer supports networkwide activation of these drivers and the application. Translated data files are transferred through the network while the command generator is started remotely by the HERMES-MW2000 itself.

Macro commands can be activated graphically on the workstation display screen. In an X11-based OSF/Motif user-interface, the descriptive names of the macro commands are depicted by button widgets that can be activated by pressing a mouse button. The command buttons are organized in a hierarchy of menus. The definitions of macro commands and their hierarchical organization in menus is stored in a database. New macros

can be added by editing the database. Activation of a button causes the associated command to be transmitted to the HERMES-MW2000 server. The server then will search its database to obtain the executable shell script.

This server software, called the executive management facility (EMF), accepts the macro commands through a shared memory segment. The user-interface client inserts the command in the shared memory segment and sends an activation signal to the EMF. This EMF then reads macro command and starts an associated UNIX script, which contains commands for obtaining the data file, possibly through the network, and for (remotely) activating the command generator and the application. These UNIX scripts contain the necessary commands for testing the presence of already transformed files.

## 6.2. Standards

For the development of the workstation, it was decided to adopt current standards as much as possible. The following standards have been adopted in the HERMES-MW2000 environment:

- (1) UNIX system V as the workstation's operating system,
- (2) Arpa-Berkeley network standards,
- (3) X11 OSF/Motif as the graphical user-interface management system,
- (4) ANSI-C programming language for new developments,
- (5) SQL for expressing data selections to a database management system,
- (6) ASCII storage format to facilitate easy transmission to other computers.

Starting with these standards, migration of the HERMES-MW2000 approach to different hardware containing a Unix environment can be accomplished with a minimal effort [13].

## 6.3. Tools

The HERMES-MW2000 environment has been developed with a number of standard software tools. These tools are mainly directed at the

construction of data-translation drivers and command generators. With the UNIX system V 'awk' tool [14], data translation programs can be developed at a high conceptual level. The 'awk' programs specify a series of rules that can be activated conditionally. Command generators use a template command script and an associated UNIX system V 'sed' program, which replaces the variable parts of a template script with the actual parameters.

In addition, a graphical editing tool has been developed to compose a data model. Two separate tools are available: a simple tool for modifying the pretty names associated with the attributes and an advanced tool for specifying a complete object-oriented data model [15].

## 7. Conclusion

The strategy of encapsulating existing applications in a network environment can be followed successfully: current workstations are powerful enough to overcome the overhead caused by the encapsulation layers. Despite the power of present workstations, the success of this approach is limited unless tools are developed that support the developer for creating and modifying the necessary data translation and command generation facilities; since the core of existing applications changes over the years, as necessarily do their application interfaces, the workstation developer should be able to rely on graphical tools to adjust the encapsulation layer adequately.

Preferably, these tools should maintain a database in which all access details of the applications are stored. Updating the encapsulation layers would require a tool to update the contents of that database. Current research in the HERMES-MW2000 project aims at the provision of these tools, making the encapsulation layers more widely and easier available.

In addition to the support for the development of tools, the client-server architecture could be extended to a multi-client/multi-server architecture. Currently, standards are evolving for network-wide client-server communication, such as the TCP/IP-based Arpa-Berkeley sockets. Uti-

lization of this type of client-server architecture could diminish the current problems with hardware integration and offer a virtual one-machine concept. The application-specific information currently stored in scripts and accessible by the central EMF server, could in this architecture be deferred to the application's encapsulation server and dramatically reduce the management complexity.

Current experience with the ISF has taught us that this format should be extended to contain a variety of other data formats as well: binary data files and free-formatted data files should also be covered by the ISF. Preferably, the three format files should be combined into one file, reducing the chance of incorrect ISF data files.

Although the HERMES-MW2000 offers users a higher level of support for their tasks than originally provided by the individual applications, future research should be directed at the definition of a user model for the various tasks of a clinician. This research is strongly related to the desk-top metaphor applied, and various different approaches should be investigated [16].

The effort necessary to establish a user-friendly interface around existing applications is considerably less than the development of these applications from scratch; evaluated software is precious and its maintenance is even more precious. The performance of current workstations is reasonable to surmount the overhead involved with the extra layers for software integration.

Although the technical integration layers have been established and provide an open, integrated workstation environment, research should be directed towards how to accomplish true clinical task-orientation and to establish which user-interface best fulfils the user's needs.

Extra attention should be paid to solutions that manage changes and provide dynamic extensions. New applications should be incorporated in the HERMES-MW2000 without modifying the already existing applications or encapsulation layers.

Finally, the integration architecture should be extended to have entries for other integrated architectures: bridges that transform messages between the various systems that currently evolve.

## References

- [1] H. Zimmermann, OSI reference model—the ISO model of architecture for open systems interconnection, *IEEE Trans. Commun.* 28 (1980) 425–432.
- [2] Hewlett Packard Company, *Using Arpa services* (February 1991).
- [3] L.R. Power, Post-facto integration technology: new discipline for an old practice, in *Proceedings of the First International Conference on Systems Integration*, Morristown, New Jersey, 1990, pp. 4–13 (IEEE Computer Society Press, Los Alamitos, Ca., 1990).
- [4] T.A. Landers and R.L. Rosenberg, An overview of multi-base, in *Distributed Data Bases*, ed. H.J. Schneider (North-Holland, Amsterdam, 1982).
- [5] T. Barsalou and G. Wiederhold, Knowledge-directed mediation between application objects and base data, in *Proceedings of the Working Conference on Data and Knowledge Base Integration* (October, 1989).
- [6] R.A. Greenes, Promoting productivity by propagating the practice of plug-compatible programming, in *Proceedings of the 14th Symposium on Computer Applications in Medical Care*, Washington DC, November, 1990, pp. 22–26 (IEEE Computer Society Press, New York, 1990).
- [7] P. Degoulet, F.C.J. Coignard, M.C. Laurent, L. Lucas, M. Ben Said, H.P. Meinzer, U. Engelmann, A. Springub, R. Baud and J-R. Scherrer, The HELIOS European project on software engineering, in *Proceedings of the IMIA Working Conference on Software Engineering in Medical Informatics*, Amsterdam, October, 1990, pp. 125–137 (North-Holland, Amsterdam, 1991).
- [8] L.F. Lunin, M.J. Ball, Perspectives on Integrated Information Management System (AIMS). *J. Am. Soc. Inf. Sci.*, 39, 3 (1988) 102–112.
- [9] S. Johnson, C. Friedman, J.J. Cimino, T. Clark, G. Hripsack and P.D. Clayton, Conceptual data model for a central patient database, in *Proceedings of the fifteenth Annual Symposium on Computer Applications in Medical Care*, Washington DC, 1991, pp. 381–385 (McGraw-Hill, New York, 1991).
- [10] J.J. Cimino, Representation of clinical laboratory terminology in the unified medical language system, in *Proceedings of the fifteenth Annual Symposium on Computer Applications in Medical Care*, Washington DC, 1991, pp. 199–203 (McGraw-Hill, New York, 1991).
- [11] J.S. Duisterhout, B. Franken B and F.S.C. Witte. Structure and software tools of AIDA, *Comput. Methods Progr. Biomed.* 25 (1987) 259–274.
- [12] W.J. Dixon, *BMDP Statistical Software Manual* (University of California Press, Berkeley, 1981).
- [13] E.M. van Mulligen, Migration of the MW2000 from HP9000 to a PC environment, October, 1991 (Internal report).
- [14] A.V. Aho, B.W. Kernighan and P.J. Weinberger, *The AWK Programming Language* (Addison-Wesley, 1988).
- [15] T. Timmers, E.M. van Mulligen and F. van den Heuvel, Integration of an object knowledge base into a medical



workstation, in Proceedings of the fifteenth Annual Symposium on Computer Applications in Medical Care, Washington DC, 1991, pp. 654–658 (McGraw-Hill, New York, 1991).

[16] L.A. ten Horn, Changes in tasks and task requirements of information technology experts and departments in view of end user computing, in: Proceedings of the 20th ICA Conference, Bonn, October, 1986, pp. 133–142.