HELP The Next Generation: A new Client-Server Architecture

Stanley M. Huff, M.D.[1,2], Peter J. Haug, M.D.[1,2], Lane E. Stevens, M.S.[2], Robert C. Dupont[3], T. Allan Pryor, PH.D.[1,2]

[1]Department of Medical Informatics, University of Utah
Salt Lake City, UT 84113
[2]Information Systems, Intermountain Health Care Inc.
Salt Lake City, UT 84111-1486
[3]Health Information Systems, 3M Health Care
Murray, UT 84157-0900

A new client-server based system which is centered around a lifetime data repository (LDR) is under construction. The goal of the new system is to maintain the patient centered decision support aspects of the existing HELP system while providing an open architecture that supports faster application development and allows execution of applications to be distributed across many computers. These goals are achieved by implementing the system with software components that are commercially available or by adhering to national and international standards for software integration. Keys to successful integration include the use of MS-DOS@, OS/2#, and UNIX§ as operating systems, Microsoft OLE 2.0 as a standard interface to the clinical database, the use of TUXEDO as a transaction/communication manager, and the use of ORACLE¶ RDBMS as the underlying database management system.*

INTRODUCTION

The HELP system has been an important part of patient care within Intermountain Health Care (IHC) for the past 20 years [1]. The system was designed with the goal of improving patient care and reducing patient costs by using the computer as a patient-centered decision support system integrated with the clinical care of the patient. The system was specifically designed to be used by physicians, nurses, and other clinicians at the bedside and at the nursing station for data entry, data analysis, and display. The system has been very successful in meeting its goals for improving patient care, as well as facilitating clinical research [2,3].

Over the years the HELP system has evolved new programs, new data structures, and new terminology as needed to keep up with the changing functional requirements. However, a new system is now being created (as a cooperative venture between IHC and 3M Health Information Systems) to meet the demands of a changing social/political climate in the health care environment and to take advantage of technological advances in both hardware and software development. The new system is not just a recreation of the HELP system using new technology, but includes a whole new philosophy of system architecture and approach. Before describing the new system it is important to first outline some of the specific goals of the new design philosophy.

- Ambulatory care is playing an ever increasing role in medical care and the outpatient environment provides many of the events and follow-up data that allow the determination of the long term outcomes of medical therapy.
- There is a need to integrate health plan coverage, scheduling, billing, and referral services across both inpatient and outpatient facilities within an enterprise.
- The increased complexity of protocol and alert logic that the HELP system performs creates a need for distributed processing.
- Some compute intensive applications like voice recognition and natural language processing tend to overload our current system.
- The development of high performance/low-cost workstations makes distribution of processing feasible.
- A faster software development cycle that uses standard languages and software development tools is a necessity in order to keep up with ever increasing application needs.
- The need to incorporate increasing amounts of textual information in the system, and the ability to do *ad hoc* queries based on word co-occurrence within a document.
- The need to enhance the HELP vocabulary (called PTXT for Pointer to TeXT), including the support of synonyms and homonyms, increased depth of the hierarchy, support of multiple hierarchical views, and support of a more formal definition of how

---

* HELP is a registered trademark of 3M Corporation.
@ MS-DOS, MS Windows, Windows NT, OLE, Word, Excel and Visual Basic are trademarks of Microsoft Corporation.
# OS/2 and IBM are registered trademarks of International Business Machines Corporation.
§ UNIX and TUXEDO are registered trademarks.
¶ ORACLE is a registered trademark of Oracle Corporation.

atomic codes are combined to make meaningful records in the database.
- Modern database management systems provide levels of data integrity, security, and internal consistency. It is desirable to use these tools as provided by commercial vendors rather than create and maintain these processes internally.

In the midst of describing the requirements for the new system there are a set of fundamental design considerations that were the foundation of the HELP system and that continue to be valid today. The new system incorporates these time tested assumptions. Specifically, the new system is patient centered and optimized for use by clinicians as an integrated part of direct patient care. This includes data entry at the point of care. Secondly, we are convinced that the best and most cost effective patient care requires the integration of alerts, protocols, and other decision support applications in the system. Thirdly, an essential component of the system is a well defined, comprehensive, coded patient database. Finally, the system must provide for the administrative activities of patient care (scheduling, billing, data security, administrative decision support, etc.) along with the patient care and clinical research aspects of the system.

SYSTEM ARCHITECTURE

The new system is currently under construction and applies an open system approach [4] to the system architecture. The elements of the overall software architecture are diagrammed in Figure 1, and individual aspects of the system are described in the following sections.

Client Environment

The client part of the system has been left as open as possible so that many application development strategies can be employed. Clients may exist on an IBM PC with Microsoft Windows or OS/2, a Macintosh♀, or a UNIX workstation. Three layers of software participate in the functions provided by a client: the application presentation layer, the business logic layer, and the communication layer.

The presentation layer provides the actual interface to the user. It presents and collects data from the user. This layer can be a GUI (Graphical User Interface), forms based, character oriented, or pen-based. The business logic layer enforces presentation and collection rules and supports application specific logic and navigation. Visual Basic was chosen as the language of choice for our first applications , but Visual C++ and Power Builder are equally supported.

The communication layer gives the client the ability to communicate with the transport component of the system to request services. We have chosen

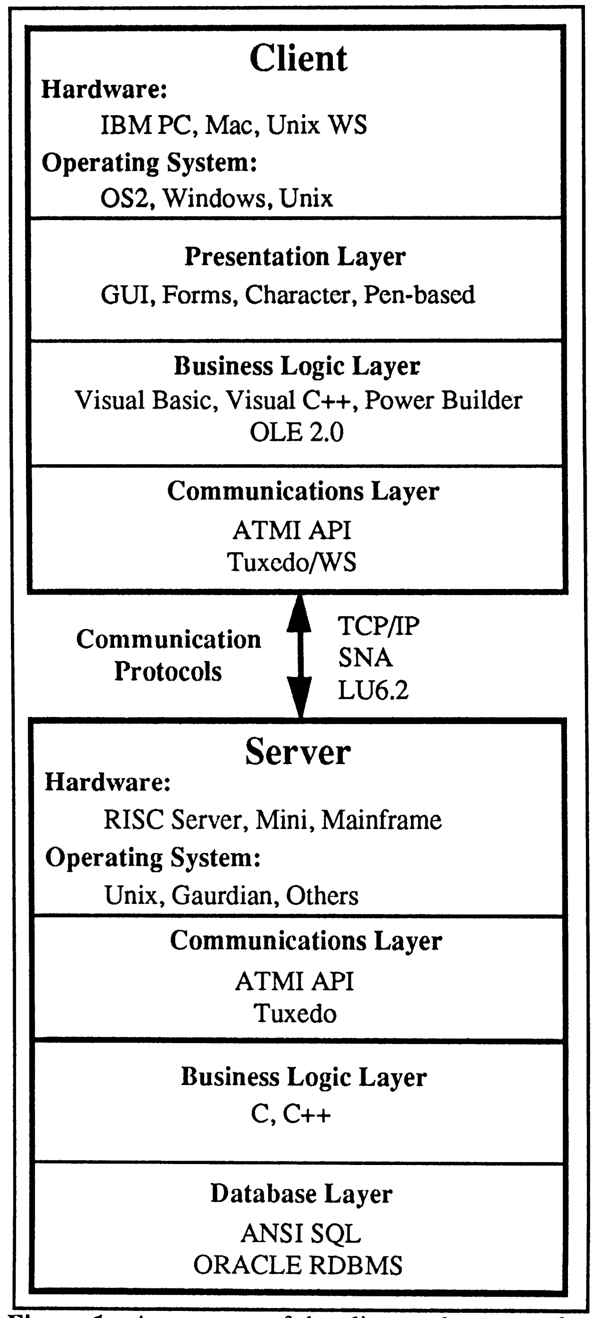

Figure 1: A summary of the client and server software layers.

TUXEDO, a commercially available UNIX-based transaction manager [5] to provide this function. A component of TUXEDO called TUXEDO/WS resides on the client while a matching counterpart of TUXEDO resides in the server. The Application Program Interface (API) to TUXEDO is ATMI (Application Transaction Manager Interface) which is a superset of the XATMI standard [6]. XATMI is supported by the X/OPEN group as one of three standard APIs for open systems.

A further constraint placed on the communication layer is a requirement that clients access the clinical

♀ Macintosh is a registered trademark of Apple Computer, Inc.

database using Microsoft Object Linking and Embedding (OLE 2.0) standard [7]. OLE 2.0 is an application interoperability standard that is being integrated into all Microsoft products, including operating systems (MS Windows and Windows NT), and applications (Word, Excel, etc.). The goal of providing access to the clinical database via OLE objects is two fold. First, the OLE objects can be accessed by any OLE enabled application such as Excel or Access. Secondly, since the database and data transfer mechanisms are hidden from the application programmer, he/she can use the properties and methods of the OLE objects without worrying about changes to the structure of the underlying database files.

The first clients are applications that manage clinical data in the outpatient setting. These applications are the most urgent since the current HELP system has a good set of programs for the hospital or inpatient setting.

Server Environment

Servers in the new system may exist on mainframes, minicomputers, or UNIX-based RISC machines. Our initial servers are UNIX-based RISC machines. As in the client environment, the servers consist of three software layers: the communication layer, the business logic layer, and the database layer.

The communication layer in the server is provided by TUXEDO and is the complementary part of the same product that was described for the client environment. The server part of TUXEDO provides two important functions. First, it handles communication protocol support for TCP/IP, SNA, LU6.2, etc. Second, it provides a transaction monitoring service. This service maps a service request to the correct server and balances the load among servers. With hundreds of concurrent requests for the same service, this layer allows multiple copies of the same server to respond to requests and then distributes the requests appropriately.

The business logic layer supports application specific logic and accesses the database. We have chosen "C" and "C++" as the programming languages for the business logic layer.

The final layer of the server architecture is the database layer which includes the database files themselves and an API to access the database. For the initial UNIX-based servers the ORACLE RDBMS product has been selected. The API to ORACLE from the business logic layer is ANSI standard SQL (Structured Query Language) for most tables, while the "text database" capabilities of ORACLE will be used for manipulation and retrieval of full-text documents in the remaining tables.

Some of the most important services provided by the UNIX-based servers are described below, along with a description of the structure of some of the tables that exist in the clinical database. Figure 2 depicts an example of how a working system might be configured and the types of clients and servers that exist in the system.

Security

An essential aspect of the new architecture is preventing unauthorized persons from reading or modifying the clinical database. The security services are used by both clients and database servers to prevent unauthorized access. Where possible these services are built on the underlying security capabilities of ORACLE. The most important piece of the security system is the logon method which provides for positive identification of the user. The initial logon method provides for authentication by eliciting a secret password from the

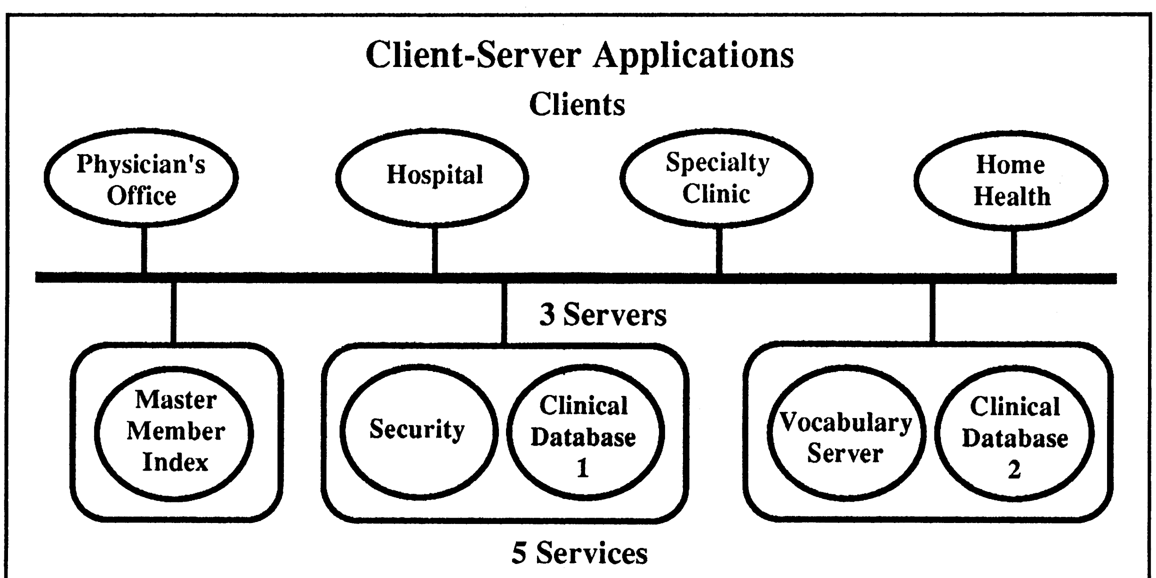

Figure 2: An example configuration of the system showing clients and servers attached to a wide area network. Only one client per facility is shown, but there would typically be several clients per facility. There may be one or more services provided by a server depending on the load balancing requirements.

user. Following identification of the user, services are invoked which indicate the functions and data tables that the user can access. Multiple levels of security are provided, with the first phase of security allowing restriction of access based on which data tables a user can read and modify. Future levels of security will be added as the client applications become more sophisticated and include restrictions based on columns within tables and on values within columns.

Patient Identification

A second essential part of the clinical database is the Master Member Index (MMI). The goal of the MMI is to provide an index of all patients or members of the health care enterprise. The MMI provides the following services:

- Establish an enterprise-wide member identifier that crosses both inpatient and outpatient settings.
- Establish a context-specific member identifier for each location within an enterprise.
- Provide search capabilities that link a member with all records in the LDR.
- Provide storage of member demographic information.
- Provide a log (audit trail) of changes to member names and demographic information.
- Provide facilities to detect and merge duplicate member records.

Patient Database

Nearly all clinical data will initially be stored in a single table, though this table will ultimately span more than one CPU. The table structure can be divided into two segments. The first portion of the table consists of fields that identify the patient that the data belongs to and other fields that are used for indexing and managing the data. Fields that are part of this fixed "header" section of the table include:

- Master Member Index Number - the unique identifier of the patient within the enterprise.
- Encounter Number - the unique identifier for a patient encounter or visit at a specific location within the enterprise.
- Event Identifier - a link to the clinical data dictionary that describes the structure and content of the data that is in the data portion of the record.
- Data Class, Type, and Field Code - a link to the first three parts of the PTXT codes that encode the data that is in the data portion of the record.
- Observation Date/Time - the date and time that the clinical observation was made.
- Store Date/Time - a system time stamp indicating when the data was stored in the database.
- Source of Information - indicates the user identifier or internal process identifier of the person or process that recorded the clinical data.
- Audit Information - several fields that are used in combination to create an audit trail of deletes and modifications to records in the database.

The second part of the table is a single relational field of variable length that contains the actual clinical data. The contents of this field are a packed PTXT string, similar in structure to the current PTXT strings used in the HELP system.

The patient data table, with its fixed relational portion and its packed PTXT string portion, is an intermediate step in the further evolution of the structure of the LDR. The exact next step is unknown, but two possible scenarios have been considered. The first option would be to evolve to a truly relational form where all the data is represented in normalized tables, possible table names would be Coagulation, Complete Blood Count, Blood Chemistry, etc. A second option would be to evolve to an Object Oriented Database (OODB). The innate structure of clinical data invites an object oriented approach, but these systems are not yet as stable as relational database platforms. The advantage of either of these models over the initial model described above is that they could be manipulated and viewed by off-the-shelf software solutions.

PTXT/Vocabulary/Data Dictionary

One of the key elements in the new architecture is an object oriented structure for the clinical vocabulary (clinical data dictionary). The new capabilities are implemented in a series of files that are too complex to present in this overview article. Many of the characteristics of the vocabulary are inherited from PTXT, but others are an adaptation of the Unified Medical Language System (UMLS) Metathesaurus structures [8] and still others are outgrowths of collaboration with the Canon Group [9]. Besides encompassing the current functionality of PTXT the new clinical data dictionary provides for:

- Representation of synonyms and homonyms.
- Representation of atomic to molecular concept mappings.
- Cross referencing of different medical encoding systems.
- Addition of site-specific terms for centrally defined concepts.
- Multi-lingual representation of concepts.
- A place to record a definition for each concept.
- A "string dictionary" that defines the component PTXT codes that can be legally combined into a valid PTXT string (record).
- Multiple hierarchical views of a given concept.
- A semantic network for non-hierarchical concept relationships.
- A link between selection lists used in client applications and the codes that are used to represent the clinical data in the database.

- Mechanisms for auditing and managing changes made to the terms in the dictionary.

Decision Support Services

Alerting and execution of other decision support logic is supported from both the client and server environments. Application and context specific rules that require foreground execution will be accessed in the client via an OLE object. System wide types of alerts and logic will be executed by the database server in response to the addition or modification of data in the database (i.e. data driven), or time driven based on elapsed time in the server.

DISCUSSION

We are not the first group to propose or build a client-server system for patient data management [10, 11]. Indeed, due to the prevalence of low cost clinical workstations most new patient care applications could be considered as client-server. What makes our experience somewhat unique is that we are migrating to a new architecture from a well established and successful existing system. The problems that this creates are much more difficult than creating a new system from scratch. A major part of our design must include plans for moving nearly 15 years of patient data from existing hardware and software to a new platform. Additionally, hundreds of user applications that operate against the old database must also be rewritten. Since neither the data transfer nor the application rewrites will happen instantaneously we have created a plan for coordinated use of both databases as work progresses. The migration plan is summarized below.

When the new MMI services are completed, a central MMI database will be loaded with our past and current patients and existing HELP system software will be modified to access the central database for MMI services instead of reading local files. A prerequisite for this change will be the reconciliation and merging of all duplicate patient identifiers.

A second major transition will occur when the clinical database services are available. At this point clinical data from existing HELP systems will be loaded into the new LDR. We currently have about 8 gigabytes of data that will need to be loaded. However, existing HELP systems in the hospitals will continue to operate against their own local databases. The data drive mechanism of the system will be used to cause any new data to be written to both the local HELP database and the new LDR. Because the data will exist redundantly in both databases, old HELP applications can continue to run while new versions are rewritten that access the new database directly. One reason that the packed PTXT strings from the old system were retained in the new architecture was to allow new and old applications to more easily coexist during what may be a rather long transition period. Also, further research is needed to test that any further evolution of the patient database can provide the same hierarchical inferencing functions as the current string structure while maintaining adequate performance.

In parallel with the activity taking place in the hospitals, outpatient clinics and physicians offices within IHC will begin using MMI services for identifying patients and the LDR for storage of ambulatory data. The outpatient activity will grow in volume as the available outpatient applications grow in number and sophistication.

In conclusion, we are optimistic about our new architecture. However, its success is by no means assured. Some potential problems include unacceptable response time of the new more atomic database, downtime or unreliability of the network servers, overload of the network due to greatly increased network traffic, and problems orchestrating the interfaces between the many commercial software packages that we are using. We hope to be able to report findings based on use of the new architecture in the near future.

Reference

[1] Warner HR, Olmsted CM, Rutherford BD. HELP - a program for medical decision making. *Comput Biomed Res* 1972;5:65-74.

[2] Kuperman GJ, Gardner RM, Pryor TA. *HELP: A Dynamic Hospital Information System*, Springer-Verlag, New York.

[3] Pryor TA. The HELP medical record system. *M.D. Computing* 1988; 5:22-33.

[4] Zimmermann H. OSI reference model - the ISO model of architecture for open systems interconnection. *IEEE Trans Commun* 1980; 28:425-32.

[5] TUXEDO. Novell Inc., 122 East 1700 South, Provo, Utah, 84606, USA.

[6] XATMI. Design Principles for Vendor-Independent OLTP Applications. Tandem Computers, Cupertino, California, 95014-2599, USA.

[7] Brockschmidt K. Inside OLE 2. Microsoft Press, Redmond, Washington.

[8] Lindberg DAB, Humphreys BL, McCray AT. The Unified Medical Language System. *Methods of Information in Medicine* 1993; 32:281-91.

[9] Evans DA, Cimino JJ, Huff SM, Hersh WR. Position statement: Toward a medical concept representation language. *Jour Amer Med Inform Assoc* 1994; In Press.

[10] 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.

[11] Chueh HC, Barnett GO. Client-server, distributed database strategies in a health-care record system for a homeless population. *Jour Amer Med Inform Assoc* 1994; 1:186-98.