A System's Architecture Which Dissociates Management of Shared Data and End-User Function

William W. Stead, M.D., Ruby B. Borden, R.N., M. William Boyarsky, David S. Crow, Terry P. Mears, R.N., Alan A. Stone, Ph.D. and Peter J. Woods
Duke University Medical Center, Durham, North Carolina 27710

Abstract

An architecture for providing an institutional systems infrastructure is proposed. The architecture permits distributed applications while maintaining an integrated patient database.

Introduction

There are two predominant system architectures in use in hospitals today. Systems such as IBM's PCS/ADS [1] and the Technicon MIS [2] are examples of the first approach - the monolithic system. In this model depicted in Figure 1, one information system is used to meet the needs of all users. Although this approach has the advantages of economy of scale and provision of an integrated database, one set of programmers must develop both the systems infrastructure and all end-user applications. The task has become too large to be done this way in a reasonable time or at an acceptable cost.

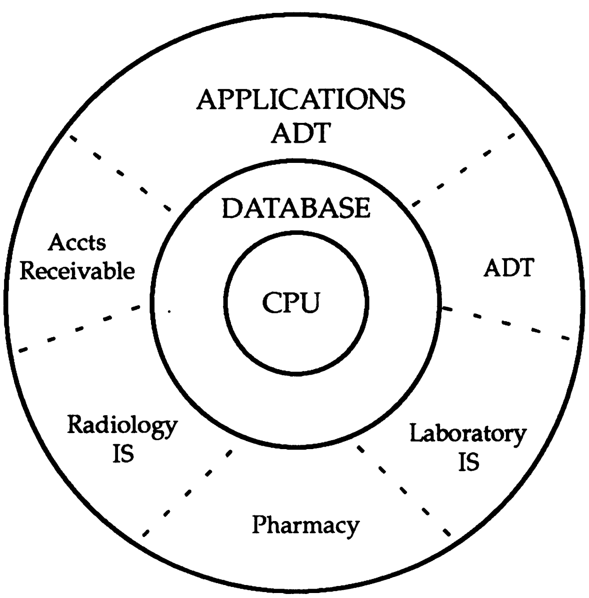

FIGURE 1: Monolithic System

The second approach, exemplified by Statlan [3], is to network autonomous systems as depicted in Figure 2. The patient identification function is provided centrally. The best system is developed/purchased for each major function and attached to a network. Each system is self sufficient and is operated independently of the other systems, but workstation technology is used to provide unified access to the data from the set of systems. Although this approach has the advantages of permitting major user groups to pursue their systems solutions separately, it does not provide an integrated patient database.

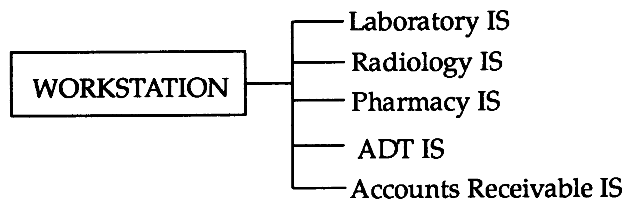

FIGURE 2: Networked Autonomous Systems

It is important to emphasize the difference between unified access and an integrated database. Unified access permits the user to see data from multiple systems on one terminal, in some cases simultaneously, in separate windows on one display. The user integrates the data in their head. An integrated database compiles data from multiple sources into a single record. The data may be presented or analyzed based upon its meaning or significance without regard to its originating source. For example, separate bills from a hospital and two clinics may be placed in a single envelope and mailed to the patient. Although this approach lets the patient examine all bills at once (unified access), it is not a single patient bill (integrated database). Automated monitoring of correct use of prophylactic and therapeutic antibiotics is an example of a clinical application which is not possible without an integrated database. The monitor must be able to integrate information from the pharmacy, operating room, and laboratory about drug orders, operating room schedules, culture results, and antibiotic levels.

Duke University Medical Center (DUMC) is seeking to fashion a new information system architecture (MCIS-1) which combines the best of the monolithic system and the networked autonomous systems approaches. The goal is to permit retention of existing systems and utilization of purchased software modules, where appropriate, without compromise to the overall effectiveness of the institution's systems infrastructure. The MCIS-1 architecture will dissociate the information system into components, specifically separating the applications programs which provide end-user function from the underlying databases. The idea is to centralize the management of data that is of general interest, where sharing is important, while allowing multiple distributed applications programs, where variation is important.

This paper begins by presenting pilot projects at DUMC which have demonstrated the capability of shifting from a monolithic system-base to use of distributed applications while maintaining an integrated patient database. The specifications for the database and communications aspects of the MCIS-1 architecture which have been developed on the basis of these experiences are outlined. Finally, the MCIS-1 approach is contrasted to other current work in the field.

Preliminary Work

The Duke Hospital Information System (DHIS) is an interactive mainframe-based system, originally implemented in 1976. Versions of DHIS have been marketed by IBM as the Patient Care System (PCS/ADS) and as Patient Management (PM) and Orders. DHIS was designed to handle institution-wide functions such as patient admission/discharge/transfer, order entry/result reporting, nursing care plans/medication administration reports, message communication, medical record abstracts/case notes analysis, specimen/x-ray/document tracking, and automated charge capture. An integrated patient accounting system manages accounts receivable. The DHIS hardware configuration includes an IBM /3090 300-S with 95 gigabytes of on-line storage and over 1,600 direct-attached devices. A Common Services Network supporting ethernet and token ring connections provides access for the more than 2,000 microcomputers in the Medical Center. DHIS supports 52 nursing stations in two hospital buildings, 18 service departments, and all outpatient clinics handling 33,500 admissions and 474,000 visits per year.

By the mid 1980s, DUMC recognized that multiple systems and hardware platforms would be necessary to provide the requisite level of end-user function. In addition to serving as an application providing shared function, DHIS was designated as a data hub to interconnect distributed applications. A communications management facility (MAPS) [4] was developed to allow a remote system to obtain data from, or insert data into, DHIS's mainframe databases.

MAPS observes the LU6.2 communications standard and is currently operational between the IBM 3090 and IBM personal computers, VAXs, and an AS-400 via locally developed APPC, Ethernet APPC, and token ring protocols respectively. Operational MAPS data transfers include history number/name look-up; admission/discharge/transfer, demographic, diagnostic study, and laboratory data downloads; and appointment data, laboratory data and radiographic report uploads [5,6,7]. Eighteen thousand queries are processed via MAPS at DUMC per day, and forty-five thousand data packets are generated daily either proactively or in response to queries.

The various systems at DUMC which are being connected through DHIS in its capacity as a data hub were developed independently and utilize their own names for medical data items. When it is feasible, databases are being restructured to utilize common terminology [8], but immediate conversion to a single data name per data item across all systems has been judged impractical. An institutional data dictionary is being developed in an evolutionary fashion. As each new piece of data is prepared for availability through the data hub, it is assigned a unique identifier and the identifiers of that piece of information in different systems are associated with that identifier. Efforts to date have focused upon defining laboratory tests, and 1,069 procedure names together with 2,652 result names have been mapped to 441 test names, representing 30% of the tests currently available for performance at DUMC.

MAPS and the developing institutional data dictionary have been used to incorporate workstation-based applications, the TMR [9] computer-based medical record, the Cerner LIS, and the IDX professional billing package into the DUMC systems infrastructure.

These pilot efforts have involved modifying the DHIS application to recognize a characteristic of a datum, such as a patient's location, to determine that data distribution needs to occur. Problems occur when that characteristic changes and an update needs to be made to previously transferred data. A new architecture is proposed which is designed to solve the problems identified during the pilot projects and to extend the underlying concept.

Proposed Architecture

The MCIS-1 information system architecture (Figure 3) will support a systems infrastructure consisting of integrated, but distributed, applications. DUMC master databases of items such as patient or provider identification and dictionaries of data item definitions will be maintained as resources independent of any information system. Generic functions ranging from order capture to electronic signature will be provided institution-wide. Distributed systems will meet local need for internal workflow processing such as the handling of a culture sample. Such systems may record data of local interest independently, but data of general interest will be defined according to the master databases and will be passed to a central transaction database/data hub, which will in turn be responsible for getting the information to any other system that needs to be updated. Any system which communicates with the data hub will be able to get data from any other system without a multitude of special purpose interfaces.

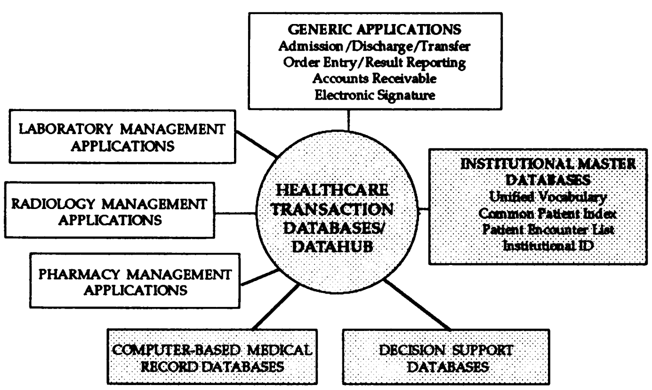

FIGURE 3: Integrated Database with Distributed Applications. Shaded components collectively provide the database.

MCIS-1 can be seen as having three types of nodes: MCIS-1 nodes, the central communication hub, and participating nodes. Each of these components may be a site of databases and associated applications. The communication hub is used to move messages between the MCIS-1 nodes and participating nodes, and may optionally be used to communicate between MCIS-1 nodes.

MCIS-1 assumes that applications will store local copies of non-owned data. This design choice typically balances the trade-off between the burdens of local storage versus response time. One of the services provided by MCIS-1 to participating nodes is the delivery of updates of locally held data, thus enabling the application to keep its database current. MCIS-1 will also provide query support for data not stored locally.

Along with the communications hub, four types of MCIS-1 databases collectively present themselves to participating nodes as a unified central resource. The four kinds of databases are: transaction, definition, institutional primary key, and derived. The transactional database is the site of institutional data verification prior to communication to the other three and participating nodes.

The primary purpose of MCIS-1 communications is the proactive distribution of new and updated data from an originating application through a central institutional distribution hub to all participating applications who have an interest in that data. Participating applications will be loosely coupled, in that they will not require 'instantaneous' data update. However, they will want to maintain MCIS-1 database synchrony in that they will want 'timely' serialized updates of any data they are storing. MCIS-1 will insure delivery of that data, thus enabling applications to maintain MCIS-1 database synchrony. Two databases are in synchrony with respect to a datum if they hold duplicate copies of that datum which are at the identical stage of update. This view of the database includes any data currently in its input queue.

MCIS-1 Databases

Definition Databases: The definition databases maintain the institutional vocabulary and rules required for the MCIS-1 nodes to communicate with each other. These databases will be relatively static.

ML (Medical Language) is the institutional data dictionary. The ML provides a unique identifier for each datum; information necessary to translate a datum between systems or databases; definition of semantic relationships that exist between data to support automatic mapping of external datasbases to the ML; and static attributes of the datum. To support the hub communications, the semantic mapping amongst data must provide hierarchical levels (all lab data, chemistry, serum sodium).

CR (Communication Rules) is the database that, for a given datum or aggregate data, specifies which node owns it and which nodes need to be sent copies of its update. This database is the registry of applications interested in receiving proactive data for a given set of patients. It will house node specified criteria for data distribution such as by patient (ex: mrn=xxxxx), by datum (ex: serum chlorine), by category of data (ex: all lab results), or by datum value (ex: blood type AB Negative).

Institutional Primary Key Databases: Institutional primary key databases maintain the permanent identifiers and demographics for patients, staff, and MCIS-1 nodes.

The CPI (Common Patient Index) database houses data required for making a positive patient identification and tracking the history of change in those fields.

The PEI (Patient Encounter Index) will house the institutional encounter list for a patient, together with the index of applications holding data on a given patient for each encounter.

The III (Institutional Identification Index) is the unique identification for employees, students, and medical care providers. Part of this database will hold a hierarchy of services and roles that personnel hold and mappings between individuals and those roles. These databases will support MCIS-1 node determination of authorization to use MCIS-1 applications or data.

The Institutional Node Index (INI) is the unique identification for MCIS-1 nodes and MCIS-1 participating nodes, together with routing information.

Transaction Processor and Databases: The transactional processor is the application that manages the transaction database, and it can be seen as a worklist processor. Each logical unit of institutional work to be done must be processed by this application and stored in the database. An incoming transaction can generate other transactions, each of which is a loosely coupled logical unit of work and which can be consumed by other worklist applications. A critical component of this scheme will be the handling of error conditions in an error worklist.

One of the functions performed by this processor will be to apply edits and rules to the data as a prerequisite to communication to other MCIS-1 nodes as well as the derived databases. During the life of a worklist item, the record will be updated with respect to status, results, timestamps, etc. Status thresholds will define the starting point for proactive MCIS-1 data distribution. Another function performed by the transaction processor will be to assign institutionally unique identification data to all data items being processed. This identifier must accompany data throughout all communications and storage, and must be utilized in the handling of all communicated data edits.

Each record in the transactional database will contain all data associated with the transaction as well as the information needed for the transaction processors to perform its function. Examples include: information surrounding data acquisition; complete edit history of any field; store status of the 'edits and rules' such that any 'replay' of the transactions need not reapply them.

The data retention of the transactions in the transaction database will be very short. Upon completion of the transaction, it will be eligible for archive. The archive is the permanent audit trail of institutional worklist activity.

Derived Databases: The derived databases are the site of long term storage of data being managed through MCIS-1. These databases will be organized to support health care decisions, computer based medical records, administrative decisions, medical education, and other institutional query needs. Their contents consist of subsets and summary data from the transaction database. They optionally may be organized to support data change history. Because derived databases will be focused on varying applications, data will be redundantly stored, and have varying retentions. These databases will not only entertain queries, but will also provide direct support to MCIS-1 applications.

MCIS-1 Communications

Communication Hub: The communication hub houses the institutional communication queues and the router application. It is architecturally distinguished from other nodes in that it owns multiple communication interfaces to MCIS-1. The router's general function is to shield nodes from needing knowledge of other nodes' existence, location, or specific data holdings. It is the implementor of communication logic based on data content of the messages.

The datum owner sends to the hub a notice of a newly created value for a given patient. It is routed to the transaction processor which performs its tasks. Once accepted, the transaction processor returns it to the communication hub for MCIS-1 distribution. The router will consult the communication rules database, and other definitional and institutional primary key databases to determine which nodes need to be informed of the results of the transaction. The router will send an outgoing update message to each appropriate node.

A node requesting data will send the request to the hub. The router will determine which MCIS-1 nodes own that data and forward the request. The requesting application may optionally declare a threshold time after which it is no longer interested in the response to the query.

The router will monitor and communicate to nodes measures such as average message latency time for any given communication channel and queue size. It will recognize danger thresholds and communicate warnings of conditions that exceed them. It will also maintain the current status of any given node ('up' or 'down').

Communications Subsystem: MCIS-1 node-to-node relationships will be loosely coupled. This means that one application need not require acknowledgement from another that it has processed a message in order to complete a transaction. It will only require acknowledgement from the communication subsystem that it has taken responsibility for that message. To be reliable, each message must be retained until the consuming partner explicitly states that it has taken full responsibility for it. The next requirement is that each participating application correctly sequence message processing, message processing failure, and message receipt acknowledgment.

The communication subsystem will not require knowledge of the message data content to perform its function. The messages that flow between MCIS-1 nodes are encapsulated logical units of work being directed to a worklist processor. The worklist name is the logical destination of the message. The communication subsystem will manage the flow of messages between applications using a queue based approach to deliver data in the same order it was produced. This approach will buffer one node from any impact of another being down. At the same time, it will ensure that the downed node has an orderly way to become 'synchronized'. Temporal synchrony within this construct then reduces to purchasing enough performance and reliability to keep the queues empty.

MCIS-1 Applications

Primary MCIS-1 Nodes: An MCIS-1 application is an application on an MCIS-1 node supporting an institutional function (one that crosses intra-institutional boundaries). It will have direct access to MCIS-1 definition databases, as well as the transaction processors. These applications will, over time, replace and supplement all DHIS functions.

Participating Nodes: An MCIS-1 participating node is any autonomus application that has a single communication interface to the central communications hub into MCIS-1. It participates in the MCIS-1 architecture as a peer. Participating nodes will be required to route all communication of institutional data updates through the central communications hub. They will have to respect institutional guidelines for defining data and for updating data that they do not own. In the latter situation, participating nodes will send the updated data in an update request to the datum's owner and handle that data as conditionally updated if kept locally. They will be prepared to honor ad hoc queries for data that they own.

An application should only know about two types of data: that which it owns and that which it does not. For the latter data, it should view the source of it, both for incoming updates as well as outgoing queries, as the hub. The participating node will not be required to know details such as who the actual owner is, or what the retrieval details of the actual data owner's local database are.

Discussion

The distinction between the word interface and the word integration is critical to understanding the MCIS-1 architecture. Although data can flow between systems across an interface, each system manages its database independently, and there is no way to insure that the specific data elements match across systems. Integration implies a much tighter link between systems. A datum is not changed without insuring that all copies of it are updated, with data transfer taking place under active control. Changes to data content are verified by processes acceptable to managers of each of the integrated systems. With a set of integrated systems, each unit can restrict its efforts to performing its unique functions, knowing that the central transaction database/data hub will maintain a datum time interlock throughout. The developing application data exchange standards address the interface problem. Additional architectural elements are necessary to support application integration.

Other groups are trying to build an integrated patient database from data derived from multiple autonomous applications. Notable examples include the work at Boston Children's Hospital [10], Presbyterian Hospital in New York [11], and the effort at Sioux City, Iowa [12]. In each of these cases, the integrated database is either constructed on the fly from the databases which support the autonomous systems or it is created as a by-product of updating those databases. The databases which underlie the autonomous system function as if the integrated database did not exist. MCIS-1 is unique in the degree to which the data hub, definitional databases, transactional databases, and derived databases are viewed as the data structure underlying the distributed applications. Local storage is viewed as an application enabler, not as a data repository.

The transition from the DHIS architecture to the MCIS-1 architecture will be evolutionary over a five year period rather than revolutionary. In other words, we will not stop and build a new system from the ground up. Development of a Common Services Network was the first step toward the new architecture in that it provided system-independent connectivity much as MCIS-1 will provide system-independent databases. Elements of the new architecture which are contained within the existing DHIS application will be progressively isolated into stand-alone resources. For example, the Common Patient Index will be constructed by isolating identification data from the DHIS patient database. Similarly, those items which should be in the medical language will be isolated from the screen definitions, edit tables, and programs which make up DHIS. Additional fields will be added to the DHIS transaction-oriented databases to support new requirements. The duration of retention of the detailed transaction information will be reduced to the length of time necessary to handle the transaction because that subset of information which is needed for long-term query capability will be passed to new derived databases.

References

1. Mishelevich DL, Van Slyke D: Application Development System: The Software Architecture of the IBM Health Care Support/Dl/1- Patient Care System. IBM Systems Journal 478-504, 1980.

2. Childs BW: El Camino/National Institute of Health -- A Case Study. In Towards New Hospital Information Systems, ed Bakker AR, Ball MJ, Scherrer JR, and Willems JL. Amsterdam: Elsevier Science Publishers, 83-89, 1988.

3. Simborg DW, Chadwick M, Whiting-O'Keefe QE, Tolchin SG, Kahn SA, Bergan ES: Local Area Networks and the Hospital. Comput Biomed Res 16:247-59, 1983.

4. Kirby JD, Pickett MP, Boyarsky MW, Stead WW: Distributed Processing with a Mainframe-based Hospital Information System: A Generalized Solution. Proc 11th Symposium on Computer Applications in Medical Care, ed Stead WW, NY NY, IEEE, pp. 764-770, 1987.

5. Boyarsky MW, Schneider DE, Lindbloom H, Pickett MP, Schneider KA, Stead WW: Towards an Integrated Laboratory Information System. Proc 13th Symposium on Computer Applications in Medical Care, ed Kingsland LC, NY NY, IEEE, pp. 654-658, 1989.

6. Rabold JS, Baysden GS, Blunden PS, Califf RM, Fang WC, Hammond WE, Pryor DB, Stead WW: The Practice Management Workstation: Providing Incentive Across Subgroups of Users. Proc 14th Symposium on Computer Applications in Medical Care, ed Miller RA, NY NY, IEEE, pp. 760-763, 1990.

7. Hammond WE, Straube MJ, Stead WW: The Synchronization of Distributed Databases. Proc 14th Symposium on Computer Applications in Medical Care, ed Miller RA, NY NY, IEEE, pp. 345-349, 1990.

8. Pickett MP, Laco RJ, Stead WW: Developing Common Data Definitions for Distributed Systems. Proc AAMSI Congress 1988, ed Hammond WE, Washington, DC, American Association for Medical Systems and Informatics, pp. 212-216, 1988.

9. Stead WW, Hammond WE: Computer-based Medical Records: The Centerpiece of TMR. MD Comput 5(5): 48-62, 1988.

10. Margulies D, McCallie Jr. D, Elkowitz A, Ribitzky R, and the Information Systems Department: An Integrated Hospital Information System at Children's Hospital. Proc 14th Symposium on Computer Applications in Medical Care, ed Miller RA, NY NY, IEEE, pp. 699-703, 1990.

11. Johnson SB, Cimino JJ, Friedman C, Hripcsak G, Clayton PD: Using Metadata to Integrate Medical Knowledge in a Clinical Information System. Proc 14th Symposium on Computer Applications in Medical Care ed Miller RA, NY NY, IEEE, pp. 340-349, 1990.

12. Andrews RD, Beauchamp CO: Integrated Clinical Applications Using a Uniform Database Prototype within the VA's DHCP. Proc 13th Symposium on Computer Applications in Medical Care, ed Kingsland LC, NY NY, IEEE, pp. 939-940, 1989.

Acknowledgements

The majority of the staffs of the Department of Medical Center Information Systems and the Division of Medical Informatics at DUMC paricipated in the MCIS-1 planning process.

This research was supported in part by National Library of Medicine Grant GO8 LM 04613, awarded by the National Institutes of Health, Department of Health and Human Services; and a grant from IBM to support the requirements phase of a Joint Study on Hospital Information Systems.