A New Approach to the Implementation of Direct Care-Provider Order Entry

Antoine Geissbühler, M.D., Randolph A. Miller, M.D.
Division of Biomedical Informatics
Vanderbilt University Medical Center
Nashville, Tennessee

Successful implementation of direct computer-based care-provider order entry traditionally relies on one of two different approaches: development from scratch or installation of a commercial product.
The former requires extensive resources; the latter, by its proprietary nature, limits extension of the system beyond capabilities supplied by the vendor. This paper describes an intermediate approach using the association of a locally-developed and controlled set of distributed microcomputer based applications and a commercial mainframe-based order entry application used as an order transaction processing system. This combination provides both an intuitive user interface and a platform for implementing clinical decision-support tools.

INTRODUCTION

Direct care-provider order entry was selected as one of the functions to be implemented early in the course of the IAIMS effort at Vanderbilt University Medical Center (VUMC) [1,2]. An intermediate approach, between development from scratch [3,4,5] and purchasing an "off the shelf" commercial system, for a "fast track" implementation of order entry was taken. VUMC entered into a joint development agreement with Shared Medical Systems (SMS) in 1992. Key components of the VUMC portion of the joint implementation included development of an external, relational database representation for orderable items, and a Generic Interface Engine that facilitated routing of information among the various VUMC "legacy" systems, the relational database system, and the order entry system. The initial clinical implementation of the SMS/Invision order entry application on two pilot units, in April 1994, relied heavily on the native SMS product. It met substantial resistance from the VUMC house staff, who expressed concerns about the lack of intuitiveness of the interface and the slow response time of the system. It also appeared that the product, by its nature as a part of a large transaction processing system, would be difficult to customize rapidly enough to respond to users' requests for changes, including clinical decision-support tools. WizOrder was developed to address these two problems, i.e., to provide a friendly user interface to the order processing system as well as a platform for integrating data and knowledge from various sources, thus enabling seamless access to clinical decision-support tools.

A plan was developed to keep SMS/Invision as the order processing system and focus VUMC efforts on the redesign of the order capture process. This paper discusses the architectural design of WizOrder, a complex set of distributed applications.

DESIGN CONSIDERATIONS

The "moving target" problem

Despite careful planning and detailed analysis, it is difficult to fully describe *a priori* the specifications of a clinical information system. A hospital is a complex and heterogeneous system where each category of users has substantially different views of the world. As the implementation of the clinical information system significantly alters the way the information is recorded and used, it is difficult for users to project their needs. Finally, the implementation of new features generates new ideas and new requirements, so that it becomes difficult to clearly separate the classical steps of software engineering: specifications, implementation and maintenance. From the developer's standpoint, this results in an unavoidable "moving target" problem, whose impact on the development of the application can be minimized by a proper design.

For the design of the user interface, a prototypical interface was rapidly evolved based on the comments of potential end-users [6] (i.e., medical house staff, nurses, pharmacists, medical receptionists).

At the application level, the various functions are separated into modules whose inter-dependence is minimized so that they can be evolved independently and, when needed, assembled in different configurations. A client program, running on the clinical workstation, provides both the graphical user interface and a communications module which provides the interprocess data exchange mechanism (Fig. 1). Each server process may exist in several instances, distributed over the network.

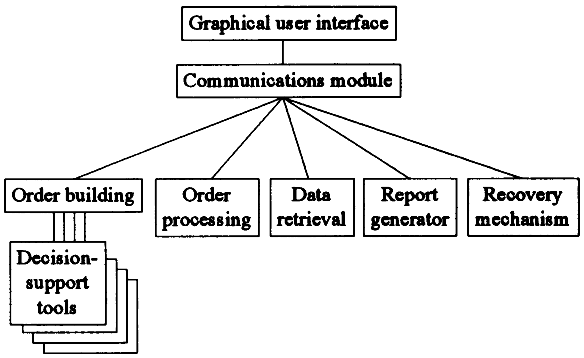

Figure 1: model for interprocess communications

Evolving towards a reference architecture

A target reference architecture for the IAIMS project at VUMC has been defined [1,7] and gives a framework for the design of new applications. However, given time constraints, users' pressure and the status of the system's infrastructure, it is often not acceptable to delay the implementation of an application until it meets all the requirements of the reference architecture. Our strategy has been to whenever possible develop components of the system designed to fit in the long-term plan, and otherwise to develop modules to serve as temporary place-holders, enabling the application to function, but ready to be replaced when the infrastructure and/or the development resources permit improvements consistent with long term objectives.

IMPLEMENTATION

Simulating the user to dissociate the order capture from the order processing

The only way of automating the process of entering data into SMS/Invision is by "simulating" a user, i.e., read and parse the screens as they appear on the terminal emulator, then fill them with data, like a user would, using a set of APIs (EHLLAPI) provided with the 3270 terminal emulation used to communicate with the mainframe. Although conceptually simple, this process requires a significant amount of effort to have a program be able to parse and format data that was originally designed to be viewed and entered by humans. It is also necessary to fully understand, and be able to reliably replicate, the order building process, most of which is not represented in externalized databases but embedded in the screen logic of the SMS/Invision system: at VUMC, fifty different screens (revise screens) with different layouts and fields define syntaxes for the capture of the various types of orders; each screen is associated with a set of routines (TCLs), written in a proprietary language, that define rules for data validation and concatenation and can dynamically modify the behavior of the succession of screens (the pathway). Fifty more screens are used for navigation purposes, including authorization functions, patient selection and data display. The order building logic is translated and encoded in the WizOrder server (Fig. 2), whereas the navigation and screen parsing is performed by the WizOrder client, thereby isolating the process of "how to build an order" from the process of "how to transmit it to the order processing system".

By simulating the user and replicating the order building logic, it is possible to dissociate the order capture and building process from the order processing. The original user-interface, hidden by a new one, still remains the final common path to the order processing system. Key features of the new interface, not available in the original system, include a free-text parser and completer, a single screen layout that remains stable during the whole session and includes the list of active orders, a geographically coherent organization of the information on the display, and single-key access to context-sensitive help, lab results and decision-support tools. The graphical user interface is provided by the WizOrder client, but the contents of the various regions of the screen is controlled by the WizOrder server.

Importing data and knowledge from other systems

Various computer systems at VUMC contain information that can be useful when made available during the order entry process. VUMC pharmacists have developed and maintain an extensive database of drug monographs and drug-drug interaction alerts used in the legacy pharmacy dispensing system. These databases have been made available to WizOrder so that this information can be displayed during order entry rather than having to be communicated by the pharmacist to the care-provider once the order reaches the pharmacy. Similarly, information about laboratory tests is imported. In both cases, an extension of the Health Terms Dictionary (HTD) is a key element in providing links between the databases.

A set of about 25,000 references from MEDLINE, arbitrarily defined as the four last years worth of references from fifteen major medical journals, is downloaded using the VUMC OVID search engine and matched with patient specific data to provide literature references relevant to the case. Tables from the UMLS Metathesaurus are used to translate drug names and diagnoses into MeSH terms.

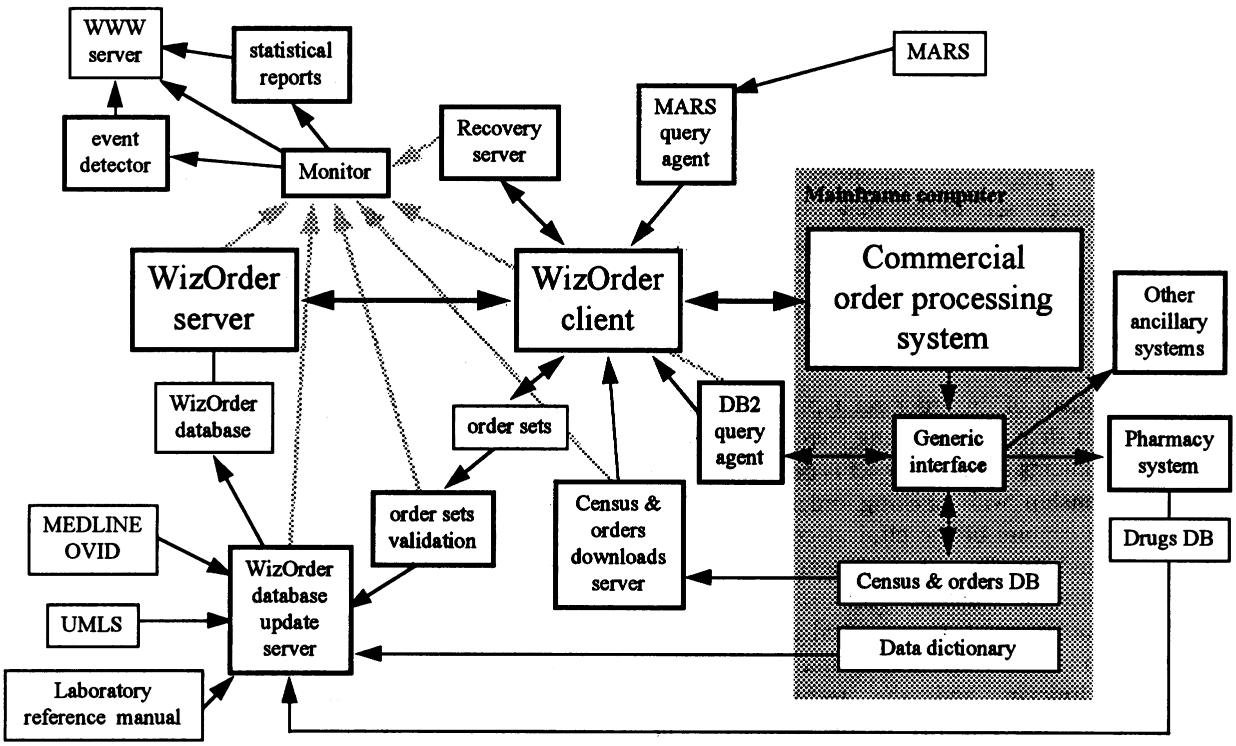

Figure 2: overall architecture of the order entry system

The Metathesaurus is also used to assist the encoding diagnoses and procedures in ICD-9.

To maintain synchronicity between the various databases, each provider of information regularly downloads the data, mostly through TCP/IP FTP, on the WizOrder server, where a process, the WizOrder database update server (Fig. 2) checks the integrity of the imported data and updates the WizOrder database. The frequency of updates is determined by the sending process based on the rate of change of the database (e.g., daily for the data dictionary and order sets, weekly for the pharmacy databases, quarterly for the laboratory reference manual). The reception of a new data set automatically triggers the appropriate process of the update server. Only successful conversions are incorporated in the WizOrder database. This process integrates data from multiple sources and generates lookup-tables to increase the performance of the WizOrder server. Detailed log-files of database assimilation activity are made available to the monitoring system.

Information retrieval and integration

Clinical patient data is stored into two different repositories. MARS [8] is a database management system based on distributed parallel processing and full-text indexing, which contains, amongst others, laboratory results, radiology reports, pathology reports and discharge summaries. Mainframe-based relational databases (DB2) contain patient demographics, allergies and orders. Data can be retrieved from both repositories to feed decision-support tools and report generators. The MARS query agent uses a library of routines developed for the VUMC MARS viewer. The DB2 query agent uses the same infrastructure as the WizOrder server, i.e., a multi-threaded server using IPX named pipes, able to service up to 15 requests simultaneously; it communicates with the generic interface using a TCP/IP socket; the generic interfaces actually runs the DB2 query. The communication module of the WizOrder client separates the data request from the actual database query. Both query agents reformat the retrieved data into a standard format, thus making its source transparent to the requesting process.

The ability to integrate and abstract data from various sources has led to the development of various on-line and printed reports such as a widely used house staff rounding report that combines current medication orders and lab results; a nurse work-sheet that displays current orders and potential drug alerts; and a sign-out sheet that summarizes for a given set of patients, demographical information and relevant orders.

Order sets development

Order sets streamline the users' order entry process. They also provide a convenient way of displaying guidelines and can take into account temporal information and patient specific data (e.g., chemotherapy protocols). Questionnaires can be embedded in order sets so that they will dynamically adapt to the user's responses or to some data derived from decision-support tools (e.g., recommendation of cost-effective empirical antibiotherapy, drug dosing based on estimation of the renal function based on a serum creatinine result).

The development of an order set typically involves three different steps: the planning step during which the overall contents of the order set is defined, as well as its relationship with existing order sets in the database; the capture of the clinical contents of the order set; and finally, its technical review and integration into the database. Authorized clinicians "check-out" order sets from a central database. The planning step is performed at that time and involves both the clinician and the technical person responsible for the integrity of the database. The clinician then modifies the order set using a modified version of the WizOrder interface created for order set maintenance that allows for the entry of formulas for date, time and dose calculations, reminders, conditional logic and various presentation attributes. Once relevant clinical experts have reviewed the order set, it passes a technical review and is integrated in the database (Fig. 2). This method allows clinicians to have the ownership of the contents of order sets and still preserves the integrity of the database, and ensures a certain level of homogeneity of style and limits the proliferation of redundant order sets.

WizOrder-To-Go

Shortly before the implementation of the WizOrder system on the Coronary Care Unit, the medical team realized that the system would disrupt their normal morning rounds by having a house officer leave the team to enter orders at a fixed workstation. By assembling the WizOrder client and the WizOrder server and its database on a portable computer, it was relatively easy to design a system, WizOrder-To-Go, that provides the same interface and decision-support capabilities as the complete system. Before rounds, the portable computer is loaded with a "snapshot" of the patient information (demographics, orders, lab results). During rounds, incremental changes are written on a diskette that can be automatically played back in the live system under the supervision of the medical receptionist.

Similarly, the modular design of the system enabled to rapidly provide a solution to the problem of entering orders for patients that have not been admitted yet in the hospital but are being seen in the clinic.

Reliability, redundancy, automatic recovery

In this distributed system, each process can be implemented redundantly. Whenever possible, this is done on different regions of the network, to minimize the impact of failures of the underlying infrastructure. Multiple instances of the various programs run on five Pentium-based servers, and are distributed in such a way that three servers must fail at the same time to bring any major subsystem or the whole system down.

Another way of providing redundancy is by evolving different methods to perform the same task. In the case of WizOrder this was merely a by-product of the implementation method, where more efficient methods superseded previous ones. Previous methods are used as backups for newer ones, in case of failure. The methods used over time to retrieve the list of active orders provide a good example: in the initial implementation, the orders were "read" and parsed from the terminal emulator screen of the SMS/Invision system, a cumbersome method that could not always capture all data; the next generation used downloads of the externalized DB2 orders database that were sent to the servers (census & orders downloads servers, Fig. 2) at fixed 15 minute intervals. This was a faster method that had the drawback of providing data potentially out-of-date by a number of minutes. The current implementation (DB2 query agent) retrieves real-time data from the orders database through the generic interface (which originally did not provide this ability).

To prevent work from being lost when a user's session fails, the communication module of the WizOrder client stores a detailed history of the session on the recovery server (Fig. 2). This history can be replayed if the session terminates abnormally. The recovery server uses the same basic infrastructure as the WizOrder server and the DB2 query agent.

Monitoring

The availability of servers is monitored from the data center's master console by a program that polls each of the instances of each server at regular intervals and reports anomalies to the operators. Each server maintains a detailed log-file.

To simplify the reviewing of these files, they are summarized by a program in HTML format, then centralized on a Web server. The most lengthy log-files are processed by an "event detector" that extracts abnormal events for a quick review.

Another tool generates a statistical view of the log-files to help optimizing issues such as servers load balancing. Monitoring of the usage of newly implemented features, of common errors and of computer-generated advice or alerts has also helped in better understanding the users' needs and in fine-tuning the application.

CURRENT STATUS

As of July 1996, the described order entry system is used on sixteen wards (400 beds) of Vanderbilt University Hospital, including the ICUs, bone marrow transplantation unit, dialysis unit, general medicine and surgery wards.

Each day, 3000 orders are entered, 70% by physicians, 20% by medical receptionists, 10% by nurses. The housewide deployment is planned to be complete for inpatients by the Fall 1996. About 500 order sets are currently available of a projected total of 1000.

Except for the 2000 lines of the graphical user interface, written in C++ to use the object-oriented IBM User Interface Class Library, most of the 40,000 lines of code are written in ANSI compliant C.

The "institution-specific" code represents 20% of the total: 2000 lines for the communications module and 3000 lines for the EHLLAPI navigation and screen parsing in the WizOrder client, 3000 lines for the data conversion in the database update server.

NEXT STEPS

Future plans include further development of optimized, task-specific interfaces, integration of other sources of data, such as bedside monitors, and, most of all, the implementation of more advanced clinical decision-support tools based on the available infrastructure and additional knowledge bases, such as diagnostic-support databases, clinical pathways and cost information.

CONCLUSION

This paper describes a novel fast-track approach to the development and implementation of direct care-provider order entry, based on a set of distributed server applications that provide the functionality for the order capture logic and clinical decision-support, a workstation based client that handles the graphical user interface and the communications with the servers, and the commercial order processing system. This architecture provides an efficient way of developing and evolving complex, integrated, task-specific interfaces to the clinical information system by assembling minimally inter-dependent modules.

Acknowledgments

This work was supported in part by the Vanderbilt University Medical Center operating funds, in part by the NLM IAIMS grant 5 G08 LM05443-04 and by a grant from the Swiss National Research Fund.

References

1. Stead WW, Sittig DF. Integrated Advanced Information Management Systems: Strategies and Architectures for Fast Track Development. *Hospital Information Systems: Design and Development Characteristics; Impact and Future Architecture*, eds. Prokosh HU, Dudeck J, Elsevier 1995;313-330
2. Sittig DF, Stead WW. Computer-based Physician Order Entry: The State of the Art. *J Am Med Informatics Assoc*, 1994; 1:108-123
3. Tierney WM, Miller ME, Overhage JM, MacDonald CJ. Physicians and patient order writing on micro-computer workstations: effects on resource utilization. *JAMA*, 1993; 379-383
4. Teich JM, Hurley JF, Beckley RF, Aranow M. Design of an easy-to-use physician order entry system with support for nursing and ancillary departments. *Proc Annu Symp Comput Appl Med Care*. 1992; 99-103
5. Gardner RM, Golubjanikov OK, Laub RM, Jacobson JT, Evans RS. Computer-critiqued blood ordering using the HELP system. *Comput Biomed Res*. 1990; 23:514-28
6. Geissbuhler A, Miller RA. WizOrder, a User-friendly Interface for Order Entry and Clinical Decision Support Tools. *Proc Symp Comput Appl Med Care*, 1995; 1002.
7. Stead WW, Borden RB, Boyarsky MW et al. A System's Architecture Which Dissociates Management of Shared Data and End-User Function. *Proc Symp Comput Appl Med Care*, 1992; 475-480
8. Yount RJ, Vries JK, Councill CD. The Medical Archival System: an Information Retrieval System based on Distributed Parallel Processing. *Information Processing & Management*, 1991; 27(4):379-389