An Informative System for Structured Data Management to Build a Cardiological Multidimensional Database

S Dalmiani, A Taddei, M Glauber, M Emdin

CNR Institute of Clinical Physiology, Pisa, Italy

Abstract

To collect clinical data generated during patient care in a multidisciplinary environment an informative system should manage each clinical aspect of involved disciplines. In this project we developed an application of what we call a Dynamic Dataset Definition (D^3) that allows rapid system deployment due to its capability of building a graphical interface directly related to a definition of collectable data. It is sufficient to initialize the system with a minimal data-set, not yet completely established, and let to successive steps to refine detailed information with current clinical use, where annotations are followed by an immediate program changes.

Moreover, in a multidisciplinary clinical environment, information is shared providing a partial view on the common defined dataset, in order to inherit data definitions of shared parameters and let them to be displayed and/or entered by several physicians. At last, the physician will retrieve information based on his personal choice or by a common scheme or calculated indexes, with a final generation of a WEB based report. This system has been implemented in a network environment to guarantee an effortless data sharing between physicians, and tested in a Cardiology and Cardiac Surgery departments at our institute.

Presently, 12 clinical subsystems are based on this model and more than 270,000 parameters have been collected on 1200 patients since August 2001 to August 2002.

1. Introduction

The main goal of each clinical information system is collecting clinical and demographic data, which generates administrative reports of clinical activities to deliver to patients and allowing other physicians to develop studies over these data at the same time. In a multidisciplinary environment as a cardiological department, each patient has a complex interaction with many physicians, from the cardiologists to cardiac surgeons: each specialist should interact one with each other in order to build a common database containing all desired clinical aspects of patient care, sharing collected data of common interest in order to reduce time consuming procedures, such as manual data entry or, even worst, multiple entry of the same data. The challenge is to develop a system capable of an easy data collection which should work as an interface between an electronic database and the final clinician users, guiding them in the definition of the dataset to be collected, or applying and expanding a predefined dataset. Finally, all data and reports should be available from remote workstations, even at physician's home through Internet connections.

To achieve these goals, an informative system called "*Matrix*" has been developed in the CNR Institute of Clinical Physiology for collecting data from in- and outpatient care process and from retrospective analyses, extracting features of interest from history, physical examination, humoral sampling and diagnostic procedures. Matrix system had to be integrated with an existing Hospital Information System (HIS), based on network infrastructure [1]; data sharing among physicians was easily obtained through network connection, and even remote data consultation and report distribution were developed through WEB interfaces. The first difficulty we had to face was the definition of data to be collected. Clinical-related information is represented as a great variety of data, from floating-point values to free text, from images to signals. Our dataset model is based on structured data of alphanumerical type or casting, leaving extraction of relevant parameters from images and signals to physicians or specialized application or systems. Aim of this project is to build an informative system for each clinical need, and, at the same time, to build a repository of data collected from other informative systems managing clinical activities that does not have inner statistical or research functionality.

2. Matrix system description

This adaptive system defines a dataset building pathway. Once a minimal data-set has been established, successive steps extend or update this basic set through dataset employment in current clinical use. In order to test each dataset change the corresponding program should change in a short time lapse. In our approach this time is reduced to a few seconds for each dataset change. This is achieved by an automatic generated interface, where, under simple graphical conventions, a redefinition of collected parameters in terms of primitive data, assumed items, range boundaries, measure units or other data value dependencies is succeeded by an immediate and automatic reconstruction of application graphical layout.

To obtain a friendly and flexible system and to follow an evolving environment like cardiological care research, a new development approach was adopted. Extreme Programming (XP) [2] is a software development approach that represents an effective method for building smaller systems in an environment where requirements are changing continuously. XP methodology was originally applied to develop in-house information systems projects with small developers team [3]. Nevertheless, XP is based on long standing industry best practices, including evolutionary prototyping, short release cycle and active end-user involvement in requirements definition. XP contributed to bundle and package specific practices to form a methodology: planning game, pair programming, small releases, collective code ownership, metaphor, continuous integration, simple design, 40 working hour per week, regression testing, on-site customer, continuous code refactoring, coding standard.

Matrix system consists of four subsystems: a DB model system, a graphical interface engine, a WEB report engine and a statistical system.

2.1. System structure

This scalable system starts from a basic configuration consisting of a workstation populating a local database with JDBC capability [4]. All text reports are generated locally, allowing document generation and retrospective analysis for both clinical and administrative purposes. Although this basic configuration could be a solution to all specific needs, overall research is currently driving the integration of local data collections, through network connections. Locally collected data may be transmitted in case of direct database integration, or published in case of decopuled Publish & Subscribe (P&S) protocol [5], according to a defined time scheduling policy or data amount driven policy. The protocol used for data extraction and transmission/publishing is HL7 [6], also embedded in a document definition table of XML protocol, thus gaining a large scale of compatibility with other external archives used for data collection.

In each configuration of Matrix, a good practice is to define a central archive containing all collected data and acting as a central fail-protected repository.

At a last glance, the local configuration is used when stable network support is missing or minimum requirements of speed and safety are not satisfied. The same configuration is also used when data flow level does not reach a minimum level of cost/effectiveness ratio as concerns equipment and investments.

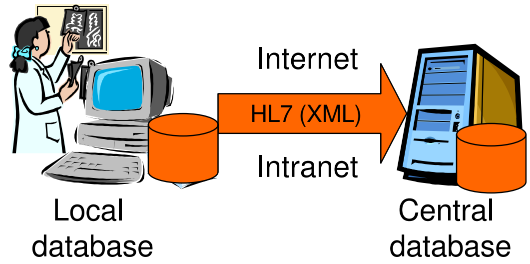

Figure 1. Basic system configuration, local database

When an Intranet is available, to optimize data-flow thus avoiding P&S overhead, we could choose a client-server configuration with a central database server housing all data concerning Matrix (Figure 2).

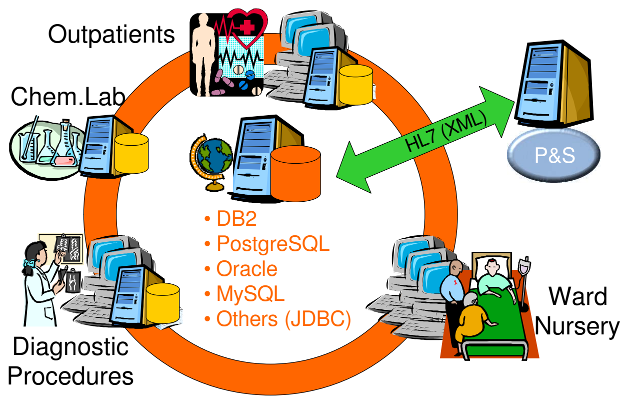

Figure 2. Network (HIS) configuration

This database contains demographics and administrative (admission and discharge) data, in addition to Matrix specific informative content, directly sharing information common to the various workstations collecting clinical data.

This configuration can be functionally extended up to the integration of the Matrix system with an Hospital Information System (HIS) and a WEB server (to allow external consultations) including other archives representing data source for booking, examination data or features, administrative data, etc.

This is the actual configuration implemented at our institute, where the Matrix has the role of a functional island as a data generator, too. The Hospital Information System contributes to increase information collected in the daily clinical routine, merged in an overall electronic medical record, because integration with other systems enhances basic data with novel annotations and examinations results, coming from ward nurses and physicians. This amount of information is used only for automatic extraction of features or data defined in Matrix.

2.2. Database

The Matrix core system has been designed for a direct integration with other systems where the same types of data definitions are present. Those data, extracted from other systems, are called "source data" and can be retrieved automatically from Matrix, or regenerated by a manual command in case of local data corruption or typing errors. Data are stored in a meta-data definition tables representing the dataset, where each feature is recorded as an independent data definition, in terms of:

- data description and assigned name, as external unique identifier;
- data type (string, integer, decimal or Boolean value, etc.);
- content definition: data length for string definitions, range for numeric definitions, list of possible values, even brokered by a trans-codification layer;
- data default value, if applicable;
- default graphical component to use in graphical interface;
- tips and help text, displayed during the use of the item;
- data input policy: optional, mandatory or conditional;
- data update permission: not allowed (read only), user locked, timeout, free, versioning;
- external data sources and priority order;

Use of this meta-data definition allows to override the maximum number of column permitted in a normal relational approach (typically 255 fields), supporting a logical dataset definition with more than 65000 fields. To define a minimum-fixed data-set, some experience is necessary, coming either from physician visit or diagnostic procedures. During the development of a Matrix dataset, the definition of a single collected feature followed a bottom-up approach, where common clinical data (administrative and identification data) were first inserted, and next by other advanced user-selected features. Anyway, relevant information to be collected could be changed immediately; from an initial minimum data-set, a feature enhancement followed a short-term experimentation pathway, which led to larger and more consistent data definitions.

2.3. Matrix application

With the widespread of computers in daily activity, both in hospital and in research environments, one of the most common requirements for new software or systems is the capability to run on different machines and under different operating systems. In order to comply with these specifications, Matrix was developed in Java language, which guarantees a free integration on many popular operating systems and platforms and allows, through the JDBC protocol, an easy integration with many different database systems.

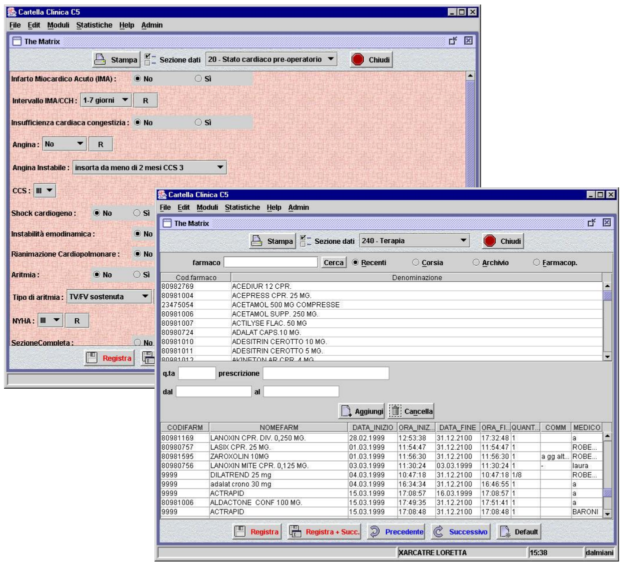

Figure 3: automatic interface and custom panel

A new interface type was developed, which does not require any specific programming, by the use of a component dynamic instance, assigned in data definition. This interface follows immediately data-set changes, due to its data-component definition, and thus called "*dynamic interface*". Besides this interface any other "classic" interface can be developed and used in the same graphical structure (Figure 3). The system defines a pathway to follow in data collection. The initial step is the patient selection with his date of characterization or session, where "session" is a counter identifying a period of time, like a ward period stay. In the subsequent steps data are grouped in subsets of clinical-related paragraphs, concerning usual custom work-path or defined entry pattern. Besides formatted data presentation and graphic data display, the main interface allows the manual input or editing, in case of missing or incorrect data. Some safety systems ensure data quality, such as exact patient/session identification, specific dictionary verification for all structured data insertions or changes, or range matching for numeric parameters. This validation is aimed not only to ensure the quality of data collected, avoiding major oversights, secondary to inaccurate data collection.

To allow remote and protected consultation, maintaining an open architecture, a WEB based report engine was developed to generate an HTML based printout, for patient or general practitioner, containing tests results or any other document based on defined dataset. In a local configuration this engine is used directly from graphical interface and generates a local HTML file containing the final printout. On the other hand, in a network configuration this engine is used by a Servlet container [8] and manages the interaction with a WEB browser. Each document is based on an HTML model previously defined and recorded in the database, for easy creation and update. The model definition contains position and format of displayed data, in a conventional and easy pattern, and is linked to the SQL query that collects desired data.

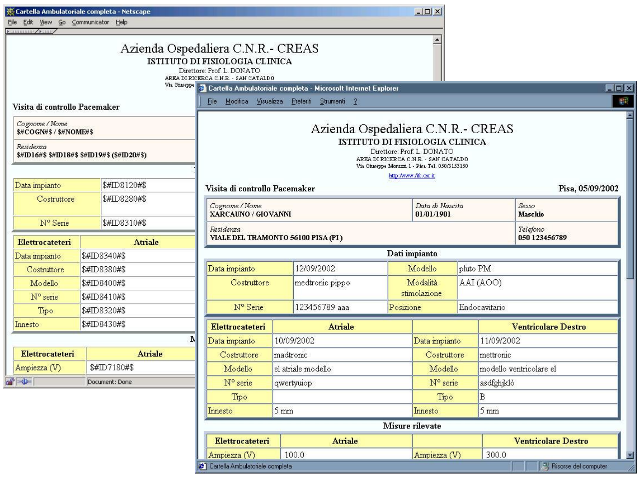

Figure 6 . Matrix WEB report engine: HTML model and HTML page generated for printout.

Cascading servlet container with open source external module of document type conversion, we can publish directly high quality printout in a stable and protected form like PDF Acrobat [9].

The last system functionality involves a statistical system that allows an user-defined data extraction and data manipulation. In those operations a data extraction or processing should take a long time, for large dataset definition, due to Matrix architecture. To avoid this time consuming operation an alternate statistical database was set-up, containing classical relational tables called "*Combined*" table. Each table is definable by end-users, directly through the graphical interface, selecting among their dataset definition in a desired order. Night-time or after a manual command, a system module updates each aggregate table gathering data from Matrix repository.

On each combined table and query can be defined from the end-user to generate indexes or tables to be graphically viewed trough an internal chart generator. Finally, any recorded data related to a dataset or a defined subset, can be exported into Microsoft Excel format, or text (CSV, plain text) for further processing.

3. Conclusion

The developed system allows a scalable configuration in Hardware support and in data collection definitions. The integration of Matrix with different data sources is essential to the achievement of a comprehensive patient record. Easy data input and consultation are obtained by data section subdivision, summary graphs and typing aids avoiding logical input mistakes. This system has an open architecture and represents a platform for successive integration and development in a short-term initiative. This project was initially developed to collect data concerning National Cardiac Surgery registry, and recently has been structured to deploy any kind of clinical data treatment. Presently 12 clinical subsystem are based on this core system, and more than 270,000 parameters have been collected on 1200 patients from August 2001 to August 2002.

References

[1] Taddei A, Carpeggiani C et al. Development of an electronic medical record for patient care in cardiology. Computers in Cardiology 1997; 24:641-644.

[2] Beck K, Extreme Programming Explained: Embrace Change. Addison-Wesley, 1999; ISBN 201-61641-6.

[3] Fowler M, Planning Extreme Programming. Addison-Wesley, 1999; ISBN 201-61641-6.

[4] Sun Microsystems. J2EE architecture JDBC protocol documentation. http://java.sun.com/products/jdbc.

[5] Sun Microsystems. J2EE architecture JMS documentation. http://java.sun.com/products/jms.

[6] Rishel W HL7 Modeling & Methodology and Control ver.3. HL7 Standards Developing Organizations 1999;

[7] Macerata A., Landucci L., et al. Networking for health care administration and delivery: the information system at the CNR Institute of Clinical Physiology. In: Proceedings of Health Telematics '95.Pisa: Area di Ricerca CNR, 1995, 113-18.

[8] Apache Software Foundation, Tomcat documentation. http://jakarta.apache.org/tomcat/index.html

[9] Easy Software Products, HTMLDOC documentation. http://www.easysw.com/htmldoc/documentation.html.

Address for correspondence.

Stefano Dalmiani.
CNR Institute of Clinical Physiology
Via Giuseppe Moruzzi 1
56124 Pisa, Italy
Phone: +39 050 315 2427
Fax: +39 050 315 2311
E-mail: dalmiani@ifc.cnr.it