E-Health in Belgium and in the Netherlands
F.H. Roger France et al. (Eds.)
IOS Press, 2002

Framework for a Clinical Information System

R. VAN DE VELDE, R. LANSIERS, G. ANTONISSEN
University Hospital Brusssels (AZ-VUB), Department of Medical Informatics
Laarbeeklaan 101, 1090 Brussels, Belgium

Abstract. The design and implementation of Clinical Information System architecture is presented. This architecture has been developed and implemented based on components following a strong underlying conceptual and technological model. Common Object Request Broker and n-tier technology featuring centralised and departmental clinical information systems as the back-end store for all clinical data are used. Servers located in the "middle" tier apply the clinical (business) model and application rules. The main characteristics are the focus on modelling and reuse of both data and business logic. Scalability as well as adaptability to constantly changing requirements via component driven computing are the main reasons for that approach

Keywords. Hospital Information Systems, Remote Clinical Workstation, Component model, Reusability, Scalability.

1. Introduction. The Dream of Software Engineering.

For more than a decade the dream of the software industry has been to achieve "software reusability" where standard components, selected from a catalogue, are customized "à la tête du client". The majority of "reuse" initiatives didn't succeed as the culture of reuse was undervalued insufficiently rewarded and tools were lagging behind. Nowadays we are still witnessing the so-called "software crisis", and software is never complete. To be successful with enterprise application development in the Internet age, IT developments must parallel hospital changes. Hospital information systems not only grow by the individual but through the computer network as hospitals merge with other hospitals. To face this increasing complexity, and responding to the need for a highly scalable, extensible and flexible infrastructure software design must be driven by a component-based approach. As many software packages are monolithic, sometimes difficult to meet the organizations needs and in order to integrate with existing systems for maximum operational efficiency within the organizational context we choose to develop our core applications. Although the cost of purchasing a package is less than the estimated cost of self-developing an application, a large cost is hidden in customisation and integration. Some publications show that the cost of customisation may rise to over 80% of the total cost of a package replacement project.

2. The Technological Environment

The goal to deliver reliable and scalable software urged us to develop a sound architecture based on the component paradigm [10]. Applications are designed as collections of collaborating components that provide the conceptual metaphor necessary for the assembly of systems, existing within a defined environment or component model. The hospital started in the 80's with mainframe based applications. In the 90's two-tier client-server architecture to run on Windows NT clients and Sun Solaris Servers replaced this mainframe environment. In 1998 ahead of its time, the Genesis project was launched. Applications were developed, based on multi tier architecture under an in-house framework using C++. In 2001 three-tier middle-ware applications gradually replaced the original client server architectures. Three particular component models currently dominate the market: Microsoft's COM+, OMG CORBA standard and Sun's Java Enterprise Beans (EJB). We decided to redeploy gradually our applications under the Java 2 Platform, Enterprise Edition (J2EE) architecture.

2.1. The Business Tier

Genesis is an in-house developed C++ framework [8] addressing the business layer. It contains a number of components allowing a developer to implements business classes and their relationships, according to a programming model that introduces the concept of reflection in a C++ system. The framework attempts to promote good programming practices by extensively making use of the C++ standard template library and so called design patterns, such as the observer, memento and proxy design patterns [4] to name only a few. At the user interface side, business models are accessible through a generic interface that is based on the framework's reflection mechanism and additionally supports generic method invocations. The simplicity of this interface significantly reduces the effort to connect user interface and business model, at the expense of trading compile time checks for run time checks. In fact, any software system can easily integrate with a Genesis based system as long as the former is capable of calling ordinary C functions.

The adoption of the J2EE platform in 2001 (an industry standard with wide spread support to ensure long-term viability) was a way to continue our experiences paved by the previous project such as (figure1):

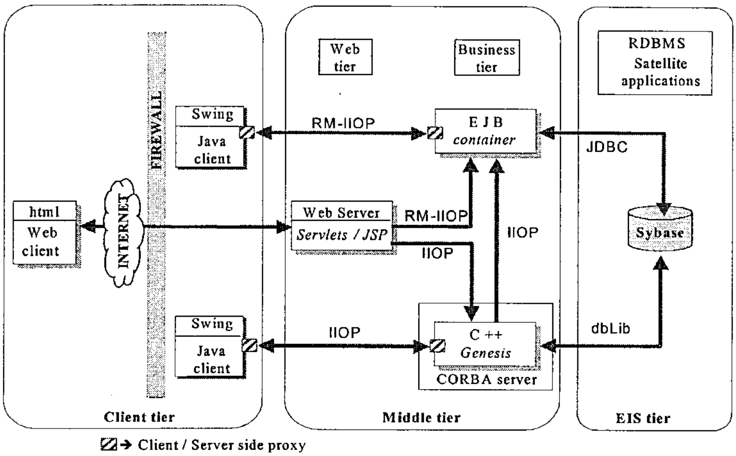

Figure 1: Framework for component software, or how to cut the dependencies between various components. The Web server tier is used to serve up the web based application and to run the servlets / Java Server Pages (JSP) that provide dynamic interaction in the web client. A Web browser makes a request to a Web Server via HTTPS to secure all communication via SSL.

- a clear separation of applications into multiple layers
- standardizing our development processes
- creating a services framework with generic components that could be reused in other applications.

In 2002 the patient scheduling and appointment system to be followed by the computerized physician order entry (CPOE) system based on J2EE using an application server (Web logic 7.0) built upon the existing RDBMS (Sybase 12.0) was deployed. An application server provides essentially the integration infrastructure in which components can be executed.

2.2. The Client Tier

The A.Z.-VUB system is accessible through two types of clients: stand-alone Java applications based on Swing classes and web-browser-based HTML clients. Full-blown Java Swing clients provide high interactivity, fast user response times indispensable in many heavy interactive applications where full application functionality is needed. This type of client can be downloaded, installed and updated automatically through the use of Java Web Start. On the other hand thin HTML-clients -aimed for GP's requiring remote, extra-firewall access to the clinical system- provide a subset of the clinical workstation functionality through a web browser. HTML is a mature cross- platform technology resembling the document metaphor with intuitive hypertext links to retrieve and present data but on the other hand characterized as slow to enter data.

Mobile Computing. Mobility within a health care building enables improved clinical care affecting patient safety. Health care workers need a tool for instant access to the patient record and to clinical reference information. Access to clinical information at the point of care by the use of bedside data input devices such as portable PC's or handheld computers (PDA) was launched in February 2001.

Besides desktop PC's, increased access is provided to every ward unit by means of a number of mobile computers connected over a wireless LAN to the system for order entry purposes, access to medical data, registration of other data. Nurses are equipped with a PDA (Palm pilot), which they use for administering medication, registration of vital signs... For the time being these data are communicated through a serial data communication link in an asynchronous way.

Internet Portal. An Internet portal gives GP's secure access to medical records of their patients from anywhere. Based on RSA-token (authentication), over a HTTPS connection a secured on-line access is provided. Referring physicians have a specific dedicated browser based access to validated medical data (only) characterized by its simplicity. For the time being they can get all the clinical information they need such as lab results, reports, history of patient visits, discharge letters and reference information. The next step will be to make the clinical information system accept data about their patients in order to give feedback and to request examinations, make appointments and so on.

3. The Information System Components

All applications are logically integrated around six basic "main" healthcare common components (HCC): Subject of care, Activities, Health Datum's, Resource, Knowledge Components and Authorization. They are briefly described hereafter:

3.1. The Patient Component

The patient component is the cornerstone of every clinical information system and provides a unified repository for storing Patient identification and demographic data and a number of functionality for the Management of patient, contact data and management of insurance data

3.2. The Act Component

This component provides the communication vehicle between requester and performer by managing (the status of) acts related to a patient. It starts with writing patient orders electronically (medications, diagnostic procedures...), scheduling, performing medical procedures and finally reporting and billing. It is linked to a clinical decision support system (CDS) as part of the knowledge component.

Many approaches of writing orders are available such as fill in the blanks, templates, complex orders (linked with protocols such as radiotherapy...). The Act component also encapsulates a workflow engine and manages a unified repository of elementary and aggregated acts (e.g.: protocols defined for clinical and organizational purposes), organized in taxonomies of semantic links and supporting inheritance. Consumable objects related to certain acts, or other acts triggered by the execution of certain acts are also defined. An order is not only limited to one service, but may span several clinical services or HC-actors. The most difficult to design and implement were medication orders although a good basis for implementation of CDS systems.

3.3. The Medical Component

The medical component provides tools for capturing, clustering and presenting medical data, aggregated in different structures following the users perception. The central repository contains multimedia patient data covering normalized data elements, strongly and poorly structured documents and image data, gathered from various departmental/satellite feeder systems.

Medical documents characterized by a strong structure they are stored in a relational database in XML format, while narrative reports are stored in HTML format. XML is used not only to exchange data between various satellite feeder systems but also to store data in the medical server and to send back the appropriate information either in the form of a Java serialized object or converted into an XML document. Every document is composed of a number of predefined sections in a specific order. The content of some sections can be pre-loaded (e.g. Lab results, radiology reports,...). In the literature [6] three types of documents are found varying from strong structured to poor structured documents. Flexibility is available through multiple methods of data entry for users with varying experience. Accordingly, three main types of medical documents are supported:

Strongly structured documents. Consisting of normalized data elements very precisely, giving birth to the use of form based screens. Those screens cannot be used during a patient contact as writing such documents are times consuming and a major burden for physicians which time is precious [1]. They are found in technical environments: medication prescription, lab analysis request...

Semi structured documents. Offering more flexibility and freedom to capture information compared to the previous one and are more suited to capture a patient encounter [9] [7] and *Poorly structured documents.* They do not tag each information chunk, bur only the paragraphs in the documents. This type of document does not allow a deep modeling of medical knowledge and does not include any conceptual representation. In fact the last two types of data capturing are more suited for narrative text reports. The status of the document -wether the document is a draft, verified signed or transmitted- can be traced. When signed, a document cannot be modified anymore but replaced by another version.

As part of the medical component, access is also provided towards bibliographic and other medical knowledge systems such as Medline, Poisson index etc.

To cope with a diversity of medical users and the specific information they use the creation and maintenance of a controlled vocabulary to which semantic layers can be added is necessary in an early stage. Although we chose to use ICD-9-CM as basis for our vocabulary server we extended this classification to incorporate our locally preferred terminology as the ICD-9-CM has severe limitations and fails to capture the clinical content. Specific functions were developed that assist users to introduce new medical concepts and links them where necessary with the appropriate sections [2] [3].

3.4. The Authorization Component

The interaction between a user and the clinical information system is controlled by the authorization component. This component provides a unified support to the declaration, definition and rights of individual users according to legal, medical an organizational requirements. They guarantee for each user profile:

- The security of specific data
- The access to a number of services requested by users based on a number of internal rules within the organization.

3.5. The Knowledge & Terminology Component

Knowledge is difficult to define and is a messy and complex concept. Although we understand the difference between data, information and knowledge the definitions become blurry. Knowledge is everywhere in the organization. Clinical information systems have 2 major roles [5]: first improving the efficiency of a process that surrounds care, second improving a knowledge support system the latter is covered by this component. A clinical system deals with three types of information. Information about the patient embedded in his medical record aims to reduce the likelihood of a poor decision that results from incomplete information. Local knowledge (e.g. organizational procedures of the hospital) and general knowledge are embedded in the middle tier. Third, global knowledge such as reference information, guidelines, expert systems.... are found anywhere: literature, internet. The boundaries between those different roles are getting more and more blurred and the various types of information become increasingly intertwined.

Depending upon the "domain" different knowledge applications are created. They provide the following intelligent services:

- On the medical level: real time assistance by providing prompts and alerts based on rules-based logic.A decision-making engine maintaining so-called "business rules" of the health care environment is implemented as part of the CPOE system to detect adverse drug events (ADEs) and nosocomial infections in inpatients and report them, using the data buried in the existing electronic medial record. To improve quality in an area, it is critical to be able to measure the frequency of these events.
- On the operational/organizational level:optimising resource scheduling management, and conflict management.
- On the administrative level: an ES-billing system based on the Belgian social legislation system. Legislation "rules" are implemented and trigger consequently the appropriate billing rules using forward and backward inference mechanisms.

3.6. The Resource Component

This component provides tools to support the managerial activities for financial, organizational and medical purposes. A resource is something an organization needs to reach its objectives and can be tangible or intangible. Tangible resources are medical departments, ward units, rooms, buildings, personnel, equipment and consumables. Intangible resources are in fact linked to the former ones and can be an agenda (available time slots), services provided,... Especially for the act component dealing with performance and scheduling of all sorts of activities the resource component is an important cornerstone.

4. Conclusions

The availability of a full range of patient data and services originating from a single source - the clinical WS (CWS) running on a variety of platforms - based on a horizontal component based multi-tiered architecture proved to be a key factor in efficiency, usability and user acceptance. The system has been going live in September 2000 but we are still refining our homegrown software.

When moving to a three-tier philosophy -besides the advantages described elsewhere a number of problems stood out [11]. Learning to cope with this new complexity is required. Gone are the days that a dominant vendor was holding your hand. When a problem occurs in a 3 Tier- environment a lot of different intervening pieces may cause the problem, due to a complex system of hardware and software originating from different vendors that must fit together. Second our development and implementation-project was complicated by the lack of (system management) tools as compared to the client server world. One of the biggest challenges was to optimise the speed of our applications in order to achieve a 1 to 2 second response time on average. A limited number of fragmented tools are appearing nowadays.

Figures. The CPOE and scheduling system as well as other components are deployed in all inpatient and outpatient clinics. We generate over 500.000 orders online on a monthly base and over 2000 appointments are scheduled on a daily base.

The number of medical reports (lab results, surgical procedures, admission notes, etc) discharge letters, monthly validated surpasses 60.000 and the patient database captures over 3.000.000 updates a month. The results and patient discharge letters are sent on-line to referring physicians on a daily base. We count more than 2000 active users in our hospital registered to the system in over 30 medical services. The number of concurrent users at specific moments is over 450.

As the system is based on a number of common components for reusability purposes it provides various advantages especially important in an academic institution such as reduced training costs (less than 4 hours), needless integration. Moreover parts of the software can be replaced as necessary without disrupting our user community or affecting other components. From the user point of view the GUI is characterized by its simplicity and uniformity. On the other hand as we aim to provide the best of breed, some functions are less specific and need to be adapted locally, by service.

References

[1] [Bates 1994] Bates DW, Boyle DL, Teich JM. Impact of computerized physician order entry on physician time. Proc Annu Symp Comput Appl Med Care. 1994; 996

[2] [Benesch 1997] Benesch C, Witter DM Jr, Wilder AL, Duncan PW, Samsa GP, Matchar DB. *Inaccuracy of the International Classification of Diseases (ICD-9-CM) in identifying the diagnosis of ischemic cerebrovascular disease*, Neurology, 1997;49(3): 660-4

[3] [Chute 1996] Chute CG, Cohn SP, Campbell KE, Oliver DE, Campbell JR. The content coverage of clinical classifications. *JAMIA*, 1996; 3: 224-33

[4] [Gamma 1995] Gamma E, HelmR, Johnson R, Vlissides J. *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing Company, 1995

[5] [Glaser 2002] Glaser J. *Experiences with knowledge applications in Medical Care*. Journal of Health Care Information Management 2002;16 (2):23-34.

[6] [Laforest 2001] Laforest F, Flory A. Medical Record and Electronic Documents: A proposal. in Proc *MEDINFO 2001*. Patel V. et al (Eds) Amsterdam: IOS Press 2001; pp.

[7] [Henry 1997] Henry SB, Morris JA, Holzemer W. Using structured text and templates to capture health status outcomes in the electronic health record. *Joint Commission Journal on Quality Improvement*. 1997; 23(12): 667-77.

[8] [Lanssiers 98] Lanssiers R (1998).The Genesis Project Internal Publication, AZ VUB, Brussels 1998;1-22.

[9] [Van Ginneken 1999] Van Ginneken AM, Stam H, van Mulligen EM, de Wilde M, van Mastrigt R, van Bemmel JH. OCRCA: the versatile CPR, *Methods of Information in Medicine*. 1999; 38(4-5): 332-8

[10] [Van de Velde 1997] Van de Velde R. Towards a component driven infrastructure for integrated health care systems. *Stud Health Technology Inform*. 1997; 45:119-27

[11] [Van de Velde 2000] Van de Velde R. Framework for a clinical information system. *International Journal of Medical Informatics*. 2000; 57(1): 57-72.