Medical Infobahn for Europe
A. Hasman et al. (Eds.)
IOS Press, 2000

AN XML-JAVA SYSTEM FOR HL7-BASED HEALTHCARE INFORMATICS

Alwyn Goh, YL Kum, SY Mak and YT Quek
USM Computer Sciences, 11800 Penang, Malaysia

Abstract: Health-Level (HL) 7 message semantics allows effective functional implementation of Electronic Medical Record (EMR)—encompassing both clinical and administrative (ie demographic and financial) information—interchange systems, at the expense of complexity with respect the Protocol Data Unit (PDU) structure and the client-side application architecture. In this paper we feature the usage of the Extensible Markup Language (XML) document-object modelling and Java client-server connectivity towards the implementation of a Web-based system for EMR transaction processing. Our solution features an XML-based description of EMR templates, which are subsequently transcribed into a Hypertext Markup Language (HTML)-Javascript form. This allows client-side user interfaceability and server-side functionality—ie message validation, authentication and database connectivity—to be handled through standard Web client-server mechanisms, the primary assumption being availability of a browser capable of XML documents and the associated stylesheets. We assume usage of the Internet as the interchange medium, hence the necessity for authentication and data privacy mechanisms, both of which can be constructed using standard Java-based building blocks.

Keywords: HL7, medical informatics, XML, Java, Web-based transaction processing

1 Introduction

The proliferation of client-server transactional messaging applications is a technological reflection of the expansive tendencies of contemporary transactional enterprises. This enlargement in the number and complexity of interacting transactional entities and processes naturally places a substantial premium on automatic and computer-aided information handling, in addition to necessitating a common message descriptive framework. The traditional approach—as exemplified by HL7 [1] and the older Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) [2] standards—is to present an integrated specification for both message syntax and semantics, with the richness of the former tending to reflect the informational complexity of the latter. For healthcare-related applications; this often results in complex data structures (encoded within messages) which can be machine-processed relatively straightforwardly, but which are difficult to display in an ergonomic manner. This has rendered HL7 and EDIFACT-like transaction processing as something of an anachronism when compared with client-server services accessible through the World Wide Web (WWW).

The inexpensive and population-wide accessibility of the Internet coupled with the seeming evolution of the Web-browser towards a (perhaps over-hyped) role as the *universal* client-side Graphical User Interface (GUI) both constitute powerful motivations for the adaptation of transactional messaging frameworks for Web-mediated functionality. The main stumbling block has always been HTML's limited capacity—in spite of successive functional enhancements—to handle complex data modelling and transactional functionality. The advent of the XML [3] descriptive framework was therefore particularly important in that it suggests the decoupling of hitherto integrated mechanisms for the handling of: (1) message structuring and semantics, and (2) data structure modelling and presentation.

XML application developers are therefore provided mechanisms expressly designed to address the above-mentioned issues. Internet-operability with its manifest attractiveness unfortunately also necessitates certain precautions against message fraud and interception, especially given the ethically sensitive and commercially valuable nature of healthcare messaging. It is in this respect that the continuing evolution of Java as a robust and functionally versatile development/operational environment is particularly crucial, thereby allowing for the straightforward implementation of authentication and data privacy mechanisms. The combination of XML and Java seems to be a particularly promising technological confluence, hence the motivation for this paper as the logical follow-on to work previously reported [4, 5], including at the last Medical Informatics Europe (MIE) conference.

2 HL7 Transactional Model

HL7 was specifically designed to model the urgencies and complexities of contemporary healthcare environments, and in this respect can be considered to be a refinement over the healthcare-related variants in the EDIFACT protocol-set. Adoption of HL7 structure and semantics allows for the *closed-loop* handling of information relating to healthcare transactions ie encompassing patient admission/discharge/transfer (ADT), resource scheduling/ordering, clinical observations/results, billing/payment and insurance claims/settlements, in addition to queries and updates of back-end institutional databases. HL7 is also intended to be application architecture independant; hence its declared concentration on the codification of message generation, interchange and processing. Design of a HL7-compliant system would therefore require consideration of the following: (1) *Data elements*: representation of transactional information, (2) *Segments*: semantically meaningful aggregations of data elements, (3) *Messages*: PDUs exchanged among system components, and (4) *Trigger events*: occurences which necessitate message interchange; the last of which constitutes an important innovation over EDIFACT in that it allows for the explicit specification of transactional messaging with respect healthcare process workflows. A transaction process would therefore be defined by the event-driven and sequential interchange of a HL7 message-set, as demonstrated in Fig 1 below:

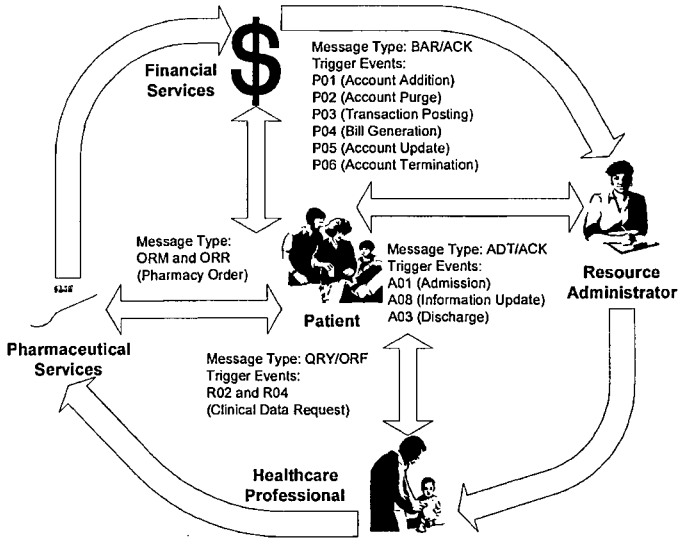

Fig 1: Healthcare Transactional Workflow

for the generic transactional episode initiated by patient entry into the healthcare infrastructure and concluded by the satisfactory resolution of finance-related issues. Information flow for this particular process is managed through interchange of ADT/ACK, QRY/ORF, ORM, ORR and BAR/ACK messages; with other processes also characterised by the message-set employed.

HL7 message structures are broadly similar to those encountered in EDIFACT, hence our continuing adherence to the XML-based document-object modeling (DOM) approach previously outlined. The interchange of XML-encoded messages in conjunction with Java-based components constitutes a client-server framework able to deliver high-value transactional functionality through Web-browser based clients connected to a public-access WAN medium. A medical informatics solution with such attributes would certainly be interesting, particularly in light of the geographically distributed and cross-organisational nature of contemporary healthcare enterprises.

3 XML-Java Distributed Application Framework

HTML forms provide a *classic* Web-based mechanism for structured document interchange, with its major shortcoming being the absence of any mechanism to define and determine semantic validity. This notion is analogous to class membership in an object-oriented programming (OOP) framework, and would allow: (1) declaration of document-object classes, with inherent reusability and extensibility; (2) determination of class membership, through the document-object parsing with respect a specified class declaration; and (3) transactional manipulation of semantically valid document-derived objects

The arbitrary complexity of the modelled data structures also necessitates an independant mechanism for document-object presentation within specific client-side environments ie HTML-Javascript for Web-browsers. This separation of data modelling and presentation functionalities can be viewed as a design-level compromise between two competing requirements in transactional computing: (1) compact encoding of complex document-objects—as in *traditional* EDIFACT or HL7 messaging—with emphasis on automatic machine-processing of transactional information; (2) ergonomic presentation of comparatively simple document-objects—as in most transactional applications with Web-browser client-platforms—with emphasis on human comprehensibility and interactivity; both of which would (in retrospect) be useful in any practical transactional application due to the likely presence of both manually-processed (ie biographical data) and automatically-processed (ie cryptographic parameters) information on the same transactional message.

The XML document-object declarative framework and stylesheet-driven mapping of document-object encapsulated information addresses both these issues, to the net effect that an XML document—representing a defined instance of some declared document-object class—is commonly associated with two additional documents: (1) *document schema*: containing document-object class declarations, (2) *stylesheet*: containing document-object presentation instructions. The interplay between these descriptive elements is illustrated in Fig 2; and the Document Type Declaration (DTD), XML and Extensible Stylesheet Language (XSL) code segments in Table 1 below.

Note the XSL-dictated transcription for data elements in the AL1 segments—ie message element DA into combination-box option "*Drug Allergy*"—as shown below in Fig 3, which also illustrates the similar result for the more complex Personal Identification (PID) segment. The flexibility of this presentation mechanism allows a particular XML-encoded message to be transmitted with client-specific and functionally-specialised stylesheets. Stylesheet-based mechanisms can also be used to support multilingual transcription of

document-objects, which would be of considerable practical untility in a multicultural environment such as Malaysia.

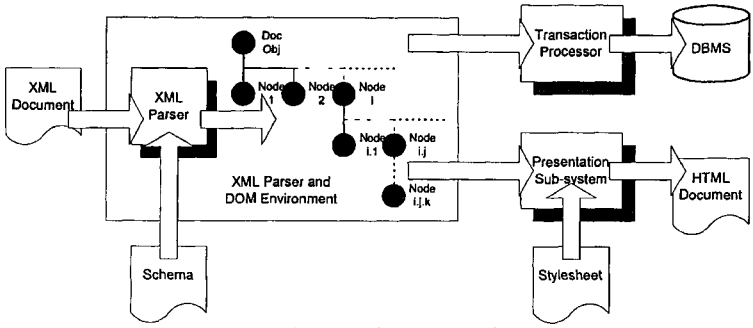

Fig 2: XML DOM Framework

<!DOCTYPE Patient_Allergy_Info [<!ELEMENT Patient_Allergy (..., Allergy_Type, ...)> ... <!ELEMENT Allergy_Type (#PCDATA)> <!ATTLIST Allergy_Type Allergy_Type_Val(...|DA|...)> ...]>	<xsl:template match = "Allergy_Type"> ... <select datafld = "Allergy_Type_Val"> ... <option value = "DA"> Drug Allergy </option> ... </select> ... </xsl:template> ... <html><head><title> Patient Allergy Information </title></head> <body> ... <xsl:for-each select = "Patient_Allergy"> <table ... datafld="Allergy_Type" ...> <xsl:apply-templates select = "Allergy_Type"/> </table> </xsl:for-each> ... </body></html>
<Patient_Allergy> ... <Allergy_Type Allergy_Type_Val = "DA"> Drug Allergy <Allergy_Type> ... </Patient_Allergy>	

Table 3.1: XML document-object declaration, definition and stylesheet for data elements in the Patient Allergy (AL1) segment within the ADT/ACK message-type

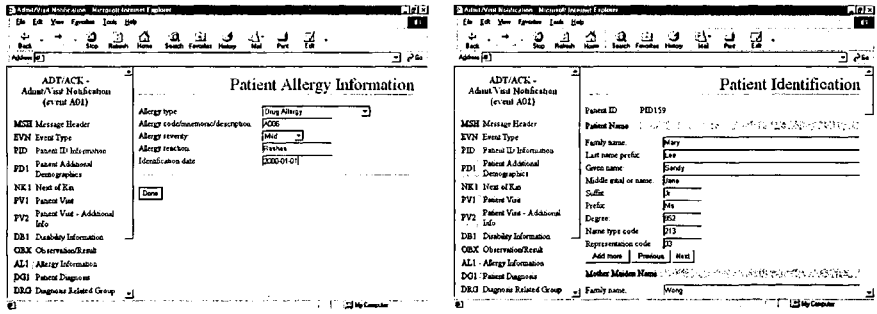

Fig 3: HTML-Javascript user interfaces for AL1 and PID segments

Our prototype HTML-Javascript interface features a simple framed design; with the left-hand frame indicating the segment-level composition of the HL7 message, in this case of type ADT/ACK. Completion of each segment-frame automatically triggers segment-level validation, with message-level validation executed prior to transmission. The work featured here utilises the International Business Machines (IBM) DOM Java application-programming interface (API), which is specifically intended to support XML-Java functionality. This is a somewhat natural combination given the strategic objective of Web-browser empowerment [6] with respect heavy-duty transaction processing.

The usage of Java client-server functionality is also useful in that relatively sophisticated (but non-standard) cryptographic mechanisms can be implemented so as to ensure secure operability over the Internet. These mechanisms include: (1) hash-tree message-authentication for the structure-sensitive transaction document-objects, (2) workload-optimised Rabin signatures with greatly reduced computation associated with message-author verification, and (3) bi/multilateral key-negotiation for establishment of secure client-to-server unicast and server-to-client multicast channels; details of which are provided in a companion paper, and which would constitute a cryptographically advanced realisation of EMR informational security requirements [7, 8].

4 Concluding Remarks

We consider movement towards enhanced Web client-server environments to be very much an industry-wide trends. The classic Web (as it existed in the early 1990s) was not capable of significant transactional complexity, since then various technologies—ie XML for data modeling and Java for client-server functionality—have emerged so as to enable delivery of services equivalent in sophistication to those previously available only to Value-Added Network (VAN) connected clients with custom-designed application software. The straightforward implementability of authentication and data privacy mechanisms based on strong cryptography would likewise tend to position the Internet as a suitable medium for healthcare informatics.

Our developmental philosophy emphasising the usage of openly available building blocks could probably be translated to significant cost advantages if adopted on a large-scale basis; due to the the manifest implementability of a high-value Web-accessible solution, and its commercial attractiveness deriving from the advantages of a potential customer-base that is both numerically large and geographically distributed. Such economies-of-scale would be strategically important for any attempt to develop and deploy a health informatics solution which is simultaneously population-wide and transparently-accessible.

References

[1] HL7 Inc. *HL7 Message Development Framework.* http://www.hl7.org. (1999)

[2] UN Economic Commision for Europe. *UN Rules and Directories for EDIFACT.* http://www.unece.org/trade/untdid. (1998)

[3] World Wide Web Consortium (WC3). *Extensible Markup Language (XML).* http://www.w3.org. (1999)

[4] A Goh. *A Java-based Framework for Medical Informatics and Electronic Commerce.* Proc of Medical Informatics Europe (MIE) Conf: Ljubljana, Slovenia. (1999)

[5] A Goh, M Mylini & AR Riza Shaharudin. *Internet-Mediated Medical Informatics using Java Client-Server Components and XML Data Modelling.* Proc of Intl Conf on Medical Imaging and Instrumentation Tech (IMIIT): Kuala Lumpur, Malaysia. (1999)

[6] J Bosak. *XML, Java and the Future of the Web.* http://sunsite.unc.edu/pub/sun-info/standards/xml/ (1997)

[7] Computer-Based Patient Record Inst (CPRI). *Security Features for Computer-Based Patient Record Systems.* http://.www.cpri.org. (1996)

[8] A Goh & WK Yip. *Security Mechanisms for Telemedicine and Medical Informatics over Insecure Public-Access Networks.* Proc of Intl Conf on Medical Imaging and Instrumentation Tech (IMIIT): Kuala Lumpur, Malaysia. (1999)