Medical Informatics Europe 2002
G Surján et al. (eds)
IOS Press 2002

XML as Standard for Communicating in a Document-based Electronic Patient Record: a Three Years Experiment

Anne-Marie RASSINOUX, Christian LOVIS, Robert BAUD, Antoine GEISSBUHLER

Medical Informatics Division, University Hospital of Geneva, Switzerland

Abstract. During the past few years, the eXtensible Markup Language (XML) has experienced a growing use for accessing, representing and exchanging information, especially in the health care environment. This paper discusses the potentials of the use of XML for the electronic patient record (EPR) in two ways: first, as a format for the exchange of structured messages, and second, as a comprehensible way of representing patient documents. These statements rely on a three years experiment conducted at the Geneva University Hospital as part of its document-centred EPR.

1. Introduction

The eXtensible Markup Language (XML) [1] is emerging as a universal format for describing the content and structure of electronic documents on the Web. Its growing use is promoted by the fact that it is independent of any vendor, platform or application, together with the availability of tools for processing and browsing it. This text-based language allows users to define their own markup or tags for data description, thus enforcing the demarcation between presentation and content. As healthcare requires information structures that are highly flexible and evolutive, XML-based applications are gaining attention in the medical field [2, 3]. In order to ensure the interoperability and consistent representation of medical data and clinical information embedded into patient records, researchers [4, 5], as well as several working groups and technical committees related to CEN, HL7 or ASTM to name a few, are currently addressing XML standards within the healthcare arena.

At the Geneva University Hospital, a dedicated software tool called DOMED (a French acronym for "DOssier MEDical") has been developed in Delphi 5, for rapid access to patients' electronic documents. This application, in use since 1999, is installed on more than 1 000 client computers in the University Hospitals of Geneva (HUG), and has already been run by 1285 different users. In order to broaden its functionalities towards managing images, laboratory results and questionnaires, as well as allowing the edition of new documents, a new version of the application renamed DPI (a French acronym for "Dossier Patient Intégré" i.e. "integrated patient record") is currently set up in the HUG. The basic architecture of this application relies on a three-tier architecture. Indeed, three independent layers share the various tasks: first, a storage layer allows for the storage and retrieval of both data from a structured relational database, and of full text narratives from a file oriented database; second, a middleware layer, which is a business logic layer, carries out most of the functionalities of the application; and third, a presentation layer deals with the

user interface on the client-side application. This separation simplifies the overall developments and allows the best technology to be selected for each layer. In particular, the XML technology has been integrated into all layers, more specifically for the exchange of messages between layers and for the representation of patient documents. These aspects are specified and discussed in detail in the following sections.

2. Exchanging Structured Messages with XML

The relatively quick and easy way of using customized tags as an interchange format for communicating relevant information has promoted the XML language for exchanging structured data, thus leading to the emergence of new protocols such as SOAP [6]. Indeed, XML tags are likely to carry semantic interpretation about the data that they encapsulate. Such semantics must be properly understood by any application in charge of reading and interpreting the data. For this, the structure of messages as well as the various criteria selected for customizing the different tags must be precisely defined.

```
<?xml                                version="1.0"?>
<REQUEST    CLIENTVERSION="service    version"
         PROTOCOLVERSION="protocol version">
  <RQBODY>
    <LOGIN>    user    identifier    </LOGIN>
    <PASSWORD>  user  password  </PASSWORD>
    ... other information such as name,
          version of the client application
  </RQBODY>

  <RQHEADER>
    <SERVICEID>  identifier  of  the  asked
                service </SERVICEID>
    <SUBSERVICEID>possible identifier of the
                  sub-service</SUBSERVICEID>
  </RQHEADER>
</REQUEST>
```

```
<?xml                                version="1.0"?>
<RESPONSE  SERVICEVERSION="service  version"
         PROTOCOLVERSION="protocol version">
  <RPBODY>
   ... XML structure standing for the proper
     response to the solicited service
  </RPBODY>

  <RPHEADER>
    <STATUSID>     request     result     or
                 error         type </STATUSID>
    <STATUSLABEL>if STATUSID is an error,
          label of the error </STATUSLABEL>
    <STATUSCOMMENT>  an  optional  comment
    </STATUSCOMMENT>
  </RPHEADER>
</RESPONSE>
```

Fig. 1: Generic XML messages (request and response) exchanged between the client application and the middleware layer

DPI utilizes the XML format for structuring messages exchanged between the client application and the middleware layer, whether they are requests, responses, or even logs (see examples in Fig. 1). Each message usually consists of two parts. The body part (<RQBODY> or <RPBODY> depending on whether it refers to a request or a response) encapsulates the relevant data, which have to be correctly interpreted by the intended application. In the case of a request, these data represent information entities needed for the proper execution of a message, whereas these data constitute the returned answer when the message is forwarded by the middleware. The header part contains information that unambiguously identifies the request, or that directly deals with the execution of the message. Although the XML technology permits the use of both tags (i.e. any element content that is defined between '<' and '>') and attributes (that are in the form *name = "value"* and expressed inside tags), for describing information entities, tags are preferably used in our applications to delimit data. Indeed, the use of attributes is better adapted to the representation of fixed parameters (see the version features in Fig. 1), whereas tags are more suitable for delimiting unformatted full-text that can vary in size.

A special interest group has been set up as part of the Division of Medical Informatics to define the correct usage of XML within our institution. The final goal is to have a library of tags that is shareable and reusable by various software applications within the HUG. At the present time, no content format or Document Type Definition (DTD) is used for validating

the logical structure of messages as well as the contextual usage of tags. Indeed, the type and the structure of data are explicitly defined into our relational databases and thus, are implicitly reflected into messages that are automatically generated from these tables. However, a minimum of data typing in XML is introduced. For example, each tag, the name of which ends with 'ID', embeds numerical data as the patient identifier <PATIENTID>, or the message status <STATUSID>. Despite the fact that storage space is no longer a problem nowadays, and rapid and performing compression programs are available, the size of XML messages can become critical when repetitive data are transferred, such as laboratory results. That is why, if possible, the number of characters of any tag name should not exceed eight characters. Moreover, the more a tag is used, the more its name should be short. These size restrictions have direct effects on the expressiveness of tag names and it should be recalled here that readable tag names are one of the main features of XML. A name must not necessarily describe the goal of the corresponding tag but must be sufficiently explicit to clarify its correct usage. This viewpoint is nevertheless questionable as global constants with no limited meaningful names are used in the source code instead of the tag names themselves. Moreover, only one tag, with a unique name, should be used to refer to the same entity. In order to reduce the number of tags, generic tags, the meaning of which is fully determined by the nested context, are preferred to specific tag names. For example, a tag for the patient's last name should be expressed as <LNAME> nested as part of an element with the tag <PATIENT> rather than <PATIENTLNAME>. The full interpretation of a tag is therefore clarified by its nested context, thus making the task of analysing XML messages a little more complex. This drawback is however alleviated by the portability of generic XML tag libraries.

3. Representing Electronic Patient Documents with XML

Besides typed data, such as laboratory results, questionnaires or images, the patient record is mainly built upon textual documents that reflect the chronological medical history of a patient. These documents show different levels of structure that vary from relatively formatted documents such as discharge letters, through descriptive documents such as laboratory reports, to informal documents such as progress notes. More than four million documents, collected from various productive sources, are available through the Geneva EPR. These documents are essentially stored in the RTF format ("Rich Text Format") and in HTML. Nearly 50 000 documents are browsed per month and the possibility to capture new documents directly from the DPI application is currently in test in the HUG.

The structure of any new document edited in DPI is based on a template or model defined in XML format (see the left part of Fig. 2). These templates play the role of DTDs or XML schemas as they precisely define the structure and content type of each paragraph. Such a structure embeds a <HEADER> and a <BODY>. The header encapsulates the properties that are inherent to the new document and that will be useful to further classify it according to various criteria, such as the document type, the identifier of its authors, of the patient, or of the passage to which the document will be attached, etc. The body that encapsulates the content of the document is divided into two parts. The <STRUCTDOC> part describes the semantic entities that compose the document. The <FULLDOC> part embeds the document itself with its page layout information, which can be stored either as a draft, a temporary text or a definitive text. As long as the document is susceptible of being modified, it is stored in an internal format (proper to the commercial editor that has been chosen for editing and displaying documents in DPI) that guarantees the storage of dynamic and controlled fields. Once the document is no longer editable, it is definitively saved into the RTF format. A CDATA section is utilized for storing the rough document whatever its

format, as it permits to disregard blocks of text containing characters that would otherwise be regarded as markup. The content of this section is then taken as the input source for displaying editable documents in DPI (see an example of a sheet of synthesis in the right part of Fig. 2).

```
<DOC>
 <HEADER>
   <DESCRIPTION>  Short  label  for  the
                document       </DESCRIPTION>
   <DOCTYPE>   Document   type   </DOCTYPE>
   <AUTHORID>       Author        </AUTHORID>
   <DOCID>   Unique   identifier   of   the
           document                  </DOCID>
     ...     Any     other     relevant     tags
 </HEADER>

 <BODY>
   <STRUCTDOC>
     <PARAS>
       <PARA      CONTENT="Content      type"
             STATUS="Present or optional" ...>
         <TITLE>Paragraph       title</TITLE>
         <CONTENT>Textual  content</CONTENT>
       </PARA>
       ...
     </PARAS>
   </STRUCTDOC>
   <FULLDOC>
       <![CDATA[.....]]>
   </FULLDOC>
 </BODY>
</DOC>
```

DPI editing user interface

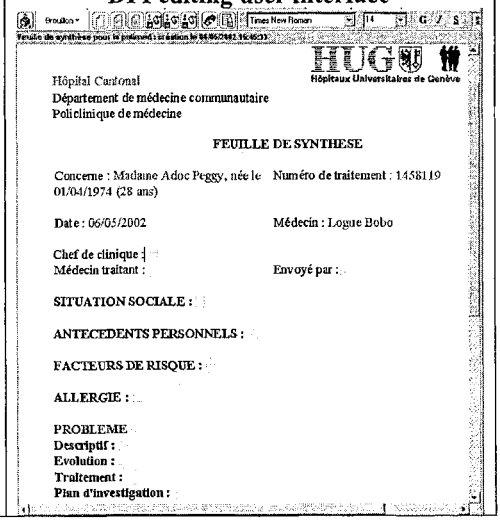

Fig. 2: Generic XML template for editing new documents in DPI

The <STRUCTDOC> markup is used for describing the semantic content of each document that is editable in DPI. The basic structure of medical texts relies on the notion of paragraphs. These blocks are natural separations that are likely to have some semantic meaning. A document is therefore made of a set of paragraphs that can be nested inside one another to multiple levels. Each paragraph is characterized by attributes inside the <PARA> markup, which are directly used by the client application to check the format of input data. For example, the attribute 'Content' clarifies the nature of the paragraph content that can be handled either as free text, as a numeric field, as a controlled field that must be filled with predefined values, or as a data field that is automatically filled by the system. Free texts and structured data can then coexist in the new document. Moreover, the attribute 'Status', which can take the values "present" or "optional", is useful to ensure the coherence of the editable document. Indeed, a document can only be definitively saved, if all the paragraphs containing the 'Content' attribute marked as "present" have been effectively filled in. Besides, the content of a paragraph is stored inside two tags. The <TITLE> markup, which is pre-filled in the template that describes the document, identifies the title of the paragraph. These titles are depicted in the document editor by textual areas that are not editable. The <CONTENT> markup encapsulates the textual content of the paragraph that is edited in DPI. This content is currently stored as a rough text. However, the works related to the automatic processing of medical texts, which have been conducted by the authors of this paper during the last few years [7], open the way towards structuring these texts thoroughly. Indeed, analysis of medical texts can be of great help for introducing semantic tags that emphasize meaningful textual information [2]. A light concept model for linguistic purposes, as proposed by the authors through an international initiative [8], is a solution for preserving the flexibility of free text documents while schematising the data they embed.

The question of directly tagging the textual document or creating a parallel document that embeds these semantic tags remains open.

4. Discussion and Conclusion

We have reported here on the integration of the XML technology in the EPR, both as a format for exchanging messages between different system components, and as a flexible means for representing patient documents. XML has been utilized successfully in both arenas and is likely to be more broadly used in our healthcare environment in the coming years. The adopted solution with XML can be suitably used within other application domains, as only tags referring to patient data are specific to the clinical domain.

The relatively quick and easy implementation of XML for messaging is a major asset, despite the fact that a common agreement on element content is still expected. The Division of Medical Informatics has gathered more than 300 XML tags within the HUG. These tags are mainly used for describing structured messages that transit between the middleware and the different client applications. The management of a common dictionary of tags ensures that the work is done in a consensual manner. Indeed, the creation of a new tag cannot be made in an isolated and independent way. Any addition must be the fruit of a common agreement between the different users, and, above all, must be coherent with the existing dictionary. This is the price to pay for converging towards the compatibility of data and services that will lead to the uniform use of message standards not only within our institution, but also with other institutions.

The desire for an EPR that is XML-structured stems from the need of sharing the clinical content, which is too often locked into free text clinical documents. The definition of templates for editing new documents offers flexible management of clinical data by introducing more structures into medical documents, thus contributing to the production of uniform and coherent documents. The software developers currently handle these models, in compliance with the users, using a commercial tool that provides an easy way to work with XML documents. Currently, about 10 templates are tested in several units of the HUG. Subsequently, an editor, able to easily create and modify these templates, will be made available for the different editorial medical services. Such an approach can lead to a complete clinical document repository in so far as each service is responsible for providing its own domain templates. XML can thus be seen as a means for introducing structures into domains that were previously poorly structured. The endeavours for structuring the patient record will lead to an internal representation of textual documents that is more adapted for further semantic indexing of electronic documents. This will also facilitate semantic-driven browsing and further retrieval within the EPR, as paragraphs can be stored separately into relational databases.

Semantic tagging of the different kinds of data (e.g. administrative and clinical), embedded into medical documents, while preserving the context annotation of medical data, should be largely extended within our institution [9]. Medical documents will thus become the new way of sharing clinical content.

References

[1] The World Wide Web Consortium (W3C) is responsible for the development and maintenance of the emerging Web standard XML: http:// www.w3.org/XML.

[2] C. Friedman, G. Hripcsak, L. Shagina, H. Liu, Representing information in patient reports using natural language processing and the extensible markup language. J. Am. Med. Inform. Assoc. 6, 1999, pp. 76-87.

[3] R. Schweiger, T. Bürkle, S. Hölzer, J. Dudeck, XML structured clinical information: A practical example. In: A. Hasman et al. (eds.). Proc. of Medical Infobahn Europe. IOS Press, Amsterdam, 2000, pp. 822-826.
[4] R. Sokolowski, J. Dudeck, XML and its Impact on Content and Structure in Electronic Health Care Documents. In: N.M. Lorenzi (ed.). Proc. AMIA Symp. 1999. Hanley & Belfus, Inc., Philadelphia, 1999, pp. 147-151.
[5] Rossi Mori, F. Consorti, Structures of Clinical Information in Patient Records. In: N.M. Lorenzi, (ed.). Proc. AMIA Symp. 1999. Hanley & Belfus, Inc., Philadelphia, 1999, pp. 132-136.
[6] The Simple Object Access Protocol: http://www.w3.org/TR/SOAP/
[7] A.-M. Rassinoux, R.H. Baud, B. Trombert-Paviot, J.-M. Rodrigues, Model-driven Medical Language Understanding: An Application for Mediating between Language Expressiveness and Formal Definition of Surgical Procedures. IMIA-WG6 Conference, Arizona, 1999, pp. 168-182.
[8] R.H. Baud, C. Lovis, P. Ruch, A.-M. Rassinoux, An Initiative to Develop an International Recipient for a Multilingual Dictionary in the Medical Domain. Submitted to IJMI, 2001.
[9] P. Ruch, J. Wagner, P. Bouillon, R.H. Baud, A.-M. Rassinoux, J.-R. Scherrer, MEDTAG: Tag-like Semantics for Medical Document Indexing. In: N.M. Lorenzi (ed.). Proc. AMIA Symp. 1999. Hanley & Belfus, Inc., Philadelphia, 1999, pp. 137-141.