XML: Defining the Transition from Paper to Digital Records

Save to myBoK

by Richard M. Peters Jr., MD

Storage, retrieval, and display of data are perpetual challenges for computer users who must also use paper. One answer to the challenge is XML, which can facilitate complex tasks for us. The author offers an overview of how XML can help us build better databases.

With each new trend in healthcare reimbursement, institutional reorganization, and computer technology during the last decade, it has been assumed that the computerized patient record (CPR) or electronic medical record (EMR) would "revolutionize" the management of medical records.

In reality, we have not seen a significant penetration of CPR/EMR systems in our healthcare organizations, but we have seen —and are confronted with daily—an explosion of additional paper and an ever-expanding volume of required forms, reporting, and documentation. We have also seen a growing interdependence between our own and other healthcare organizations to exchange patient and provider-specific information, and explicitly to justify and document each administrative and clinical interaction we undertake.

We are confronted with authorization requests, referrals, requests for additional or supporting documentation, encounter reports, utilization data and profiles—which are all derived from and interdependent with the traditional medical records documents we have long collected and managed. In addition, under the Health Insurance Portability and Accountability Act (HIPAA), we are seeing regulations changing the required format and submission of much of our health plan and billing information, from paper-based formats such as the HCFA 1500 and UB-92 to predefined electronic formats such as the X12 837.

Essentially, we have an unbearable and diverse paper collection and cataloguing task. We know that the management of much of this could be simplified if it were automated and computerized. But the computerization of that data has been piecemeal and the computerization standards, such as the X12 standards, apply only to a small part of the overall paper burden—that is, claims.

Meanwhile, the administrative computer system, billing, claims and accounts receivable, and CPR/EMR vendors have also provided us with partial solutions, all basically incompatible and all producing paper printouts that do not match the paper-based formats we have struggled so hard to standardize within our own institutions. We have standardized these paper forms and formats within our institutions for a number of practical and logical reasons. When systems vendors are asked to format their printouts to conform, however, we find it involves expensive and time-consuming customization—or we are told it is impractical or even impossible.

What we end up with is a marginal benefit from automation and an increased complexity and work load in manual cataloguing and retrieval. You have to wonder who is winning here—the computer with its incessant printing, or the human trying to make sense of it all.

What we need is a computer data format—a process or format that the computer can store its data in—that conforms to a number of requirements that will support and not hinder the inevitable computer-to-paper and paper-to-computer continuum. We are just not going to completely eliminate paper.

Our requirements are:

- the format within the computer systems has to be consistent across all data, form, and document types and across/between systems from all vendors

- the output from healthcare administrative and clinical computer systems to paper—the printouts, in practical terms—have to be fully configurable to local and internal formats and presentations
- parts of one data set or one document have to be easily extracted to form a part of another data set or document where the same data is needed for another purpose. In short, we need to seamlessly and painlessly combine data from multiple sources into multiple combinations and presentations
- local and institutional variability in methods of data collection and presentation must be preserved, but must not affect data exchange and interchange
- cataloguing, retrieval, reporting, and data analysis must use the same simple tools and methods across all data and document types, whether it is administrative, clinical, or reporting data

For all intents and purposes, this sounds wonderful, pie-in-the-sky, impractical, and unobtainable. But the reality is that it is far from unobtainable. In fact, these exact requirements are routinely met in the publishing business and are now being met in the expanding world of non-healthcare electronic data interchange (EDI), such as in the auto industry, and for data and document exchange within governmental agencies like the Department of Defense. The way these seemingly insurmountable requirements are being met is through the use of XML (eXtensible Markup Language).

Creating Clinical or Administrative XML

While we've all heard of miracle cures in medicine, we are not all believers in them; nor are we necessarily happy to hear that someone is promoting a new one. XML, however, is not a miracle cure. It is merely an extremely logical solution to the requirements defined above, which essentially confront all users of computer systems who must also interact with paper. It is also a very straightforward way to address these requirements across industries.

The fundamental principles of XML were previously described in "XML Makes Its Mark" by Rachael Sokolowski in the November/December 1999 *Journal of AHIMA*. To review, XML is a syntax for computer data storage, retrieval, and display that involves the use of tags that both proceed and follow a specific part or piece of data (such as part of a form, a piece of text, or a document). In essence, the "tags" are a set of markers for the computer to read so that it can recognize that what is between those tags is a piece of data. In addition, the tags themselves have descriptive meaning, so that the computer, when it reads the tag, recognizes what type of data is between those tags.

Although XML is powerful and can facilitate very complex computer tasks, the principle of tagging is simple. XML's power lies in that simplicity. Let's take a simple example, such as a diagnosis—a diagnosis that could be part of a clinical note, a history and physical, a discharge summary, a bill or claim, an encounter report, or a patient statement. Taking the diagnosis of Myocardial Infarction, we can define and illustrate how XML tagging allows us to meet the requirements defined above.

To tag Myocardial Infarction in XML, let's first define it and what we are collecting in terms of data about it. First, Myocardial Infarction is just a set of words. In reality, it has some additional characteristics of practical and essential value to us in healthcare, specifically what type of Myocardial Infarction it is. These characteristics include its unique manifestations in this patient, and what code it might match to, such as an ICD-9-CM code. Let's take an ICD-9-CM string and tag the contents for computer storage, retrieval, and display in a step-by-step process.

Step One

Let's define the clinical case—a patient with an acute myocardial infarction, with evidence from the EKG that the infarct anatomically affects the anterior portion of the heart.

Step Two

Let's define the administrative situation—this is a patient seen in the Emergency Room, this is the initial episode of care for this condition for this patient, the physician is going to fill out a clinical note in the emergency department record or a "superbill" in his/her office record, and the clinical note is interpreted into the "primary diagnosis."

Step Three

The physician generates a clinical document (emergency room note, or an emergency department superbill) on a computer system. Note that this step, which involves entering the data into a computer, could also be undertaken by an emergency department clerk or by a billing clerk in the medical records or billing department.

What the physician, departmental clerk, or billing specialist enters is:

Diagnosis:	Myocardial Infarction
Date/Time of Onset:	October 10, 1999 0500 AM Pacific Standard Time (PST)
Location:	Anterior
Acuity:	Acute
Episode:	Initial Episode of Care
Status:	Primary Diagnosis
Code Type:	ICD-9-CM
Code:	410.11

Step Four

The computer "tags" the data in XML syntax, as follows, where a start tag is a descriptive term like "DIAGNOSIS" in between a less-than (<) and a greater-than (>) sign and an end tag is the exact same descriptive term, preceded by a back-slash (/) symbol in between a less-than (<) and a greater-than (>) sign:

<DIAGNOSIS>
Myocardial Infarction
<ONSET>
<DATE>
<Y>1999</Y>
<M>10</M>
<D>10</D>
<HR>5</HR>
<MIN>0</MIN>
<TMZ>PST</TMZ>
</DATE>
</ONSET>

<EPISODE>Initial Episode of Care</EPISODE>

<LOCATION>Anterior</LOCATION>

<ACUITY>Acute</ACUITY>

<STATUS>Primary Diagnosis</STATUS>

<CODE>410.11
<TYPE>ICD-9 CM</TYPE>
</CODE>

</DIAGNOSIS>

Note that the <DIAGNOSIS> start and </DIAGNOSIS> end tags surround all the other data, because all of the other data, such as <LOCATION> and <CODE> for example, belong to that diagnosis. Step Five The computer stores this data as a string of text. This string of text has no punctuation, carriage returns, or line breaks, and is literally just a block of text that, for our example, would look as follows:

<DIAGNOSIS >Myocardial Infarction<ONSET> <DATE> <Y>1999</Y> <M>10</M> <D>10</D> <HR>5</HR> <MIN>0</MIN ><TMZ>PST</TMZ> </DATE> </ONSET> <EPISODE>Initial Episode of Care</EPISODE> <LOCATION> Anterior</LOCATION> <ACUITY>Acute</ACUITY> <STATUS>Primary Diagnosis</STATUS> <CODE>410.11 <TYPE>ICD-9.CM</TYPE> </CODE> </DIAGNOSIS>

This "block" is not as simply laid out or as easy to decipher as it is in outline form as displayed in Step Four, but, as you can see, it is still readable by a human or computer. In fact, one of the beauties of well-designed XML tagging is that the data in any form—outline, text block or string—is still readable by humans and computers alike. This is in sharp contrast to other formats we are used to such as the HCFA 1500 electronic flat-file format, an X-12 837 electronic message format, or an HL7 electronic message format.

Using XML Data Once It Is Captured and Tagged

Now that the Myocardial Infarction data has been captured and stored in the system in XML, we want to retrieve it and use it. Examples of uses for those data are:

- to generate and print a paper copy of the clinical note or superbill
- to feed the diagnosis into a proprietary rules-based claims engine
- to generate a paper-based HCFA 1500 or UB 92
- to generate an X-12 837 electronic claim

What we want to do in each of these scenarios is to retrieve the Myocardial Infarction data from the computer, change its format and structure, and then use it for something else. Traditionally, this means that for each of these desired end results or outputs, we have to explicitly program the computer to read the data, convert it into something else, and then have the computer display or print it for us. We would do this for each of the instances outlined above, and for any new required output we would need to program the computer yet again.

Using XML, the situation is quite different. Unlike the traditional situation, in which we are constantly reprogramming the computer, XML allows us to convert data from the stored format to any other desired storage or display format without reprogramming.

In XML we use a set of translation files or scripts to convert data from one form to another. These translation files are simple text files that the computer can read on the fly and that convert the data into or out of one format and into or out of another. These translators are known as "style sheets," and under XML will most likely be XSL (eXtensible Style Language).

For the four output examples listed above, we would have four separate XSL files that would each read the same original XML text string for Myocardial Infarction and generate four different presentations. Again, what is important here is that to get different outputs, either those we want now or those we will need in the future, all that is required is to define a new XSL style sheet. Reprogramming the computer is not necessary. The profound importance of this to medical records management is that it frees us from the tyranny of computer systems that define a specific output when our needs and requirements are always changing (specifically when it comes to new requirements and regulations, such as HIPAA).

To convert data, a computer system would take the XML block of text and the pre-defined XSL file and complete the following steps:

1. use an XML parser to read the XML text block and catalogue its parts—in effect, it would recreate the outline view for itself
2. parse the XSL file, which would then tell it how to format the XML string
3. match the tags, one by one, with the description of what to do with that tag and its data, as described in the XSL file

4. generate the required output. Note that this output could be to a computer screen, a printer, or to another computer system as a message

Let's take a look at Conversion 1, as itemized above. We defined Conversion 1 as the generation and printing of a paper copy of the clinical note or superbill. In this situation, we could define in XSL for the computer to convert the tags to labels and to format the data within each string. Let's first display the XML block of text:

<DIAGNOSIS>Myocardial Infarction
<ONSET>
<DATE>
<Y>1999</Y>
<M>10</M>
<D>10</D>
<HR>5</HR>
<MIN>0</MIN>
<TMZ>PST</TMZ>
</DATE>
</ONSET>
<EPISODE>Initial Episode of Care</EPISODE>
<LOCATION> Anterior</LOCATION>
<ACUITY>Acute</ACUITY>
<STATUS>Primary Diagnosis</STATUS>
<CODE> 410.11
<TYPE>ICD-9 CM</TYPE>
</CODE>
</DIAGNOSIS>

Then we will display the desired output:

Myocardial Infarction - Onset October 10, 1999 5:00 AM PST Acuity Acute **Episode** Initial Episode of Care **Location** Anterior **ICD-9 CM** 410.11 **[**Primary Diagnosis**]**

Several things are apparent:

- not all of the data in the string was displayed
- the order of the data has changed
- some tags have been "converted" to bold titles, while some have been ignored
- bold brackets ([ ]) have been added around "Primary Diagnosis"
- the data for "Month," which was "10" in the XML string, has been converted to "October" in the output, and the time of "5" hours and "0" minutes has been converted to "5:00 AM"

You can see the power of a simple set of text files. The XML data file holds a significant amount of contextual data, while the XSL file gives us almost unlimited formatting control over the data. To meet the requirements of Conversions 2 through 4, we would just write additional XSL files.

In addition, XSL can convert from XML to something else such as X-12 837 or a HCFA 1500 flat-file format, but it can also do the exact reverse—convert any file from something else into XML. What that means is that our older computer files and the files in our legacy computer databases and systems can be converted from a proprietary format to an open format such as XML. Once the data is in XML, we gain all the advantages inherent in the XML format. We have, therefore, converted old, virtually unusable data into a format we can use to do all kinds of things.

The Economics of XML

While XML and XSL allow us great flexibility in data storage and output formatting, they also have some very significant economic advantages. It is generally well known that XML and XSL are powerful tools. What is less well known is that the software needed to use them is essentially free to anyone or any company that wants to incorporate XML and XSL into its systems.

In a nutshell, virtually all of the code necessary to generate XML and XSL and the parsers and translators needed to use XML and XSL to convert data are available from a number of reputable sources. For example, IBM provides an XML parser to the industry completely free of any licensing fees—and theirs is just one of many.

As Sokolowski mentions in her article, the standardization of XML and XSL is managed by a cross-industry consortium. This means that tools built for one industry, such as publishing, are applicable across all industries, including healthcare. This broadens the competitive pressure to develop and implement cost-effective and highly featured XML and XSL tools.

These two economic advantages alone make XML a compelling solution to computer software developers.

Further, XML is emerging as the preferred syntax in all Internet and World Wide Web transactions, to eventually replace HTML (HyperText Markup Language), which is currently the preferred syntax. For example, both the newest versions of Netscape's and Microsoft's Internet browsers support XML. If you have Microsoft Internet Explorer Version 5.0 on your computer and you wish to see an illustration, take the Myocardial Infarction XML text block above and type it as displayed under Step Five into a text window in your computer, like Notepad in Microsoft Windows. Save it as a text file with the extension .txt. Next, go into your file manager and change the extension from .txt to .xml and then launch the renamed .xml file. On a Microsoft Windows machine, this will automatically launch Microsoft Internet Explorer Version 5.0, and you will see the file parsed automatically into outline form.

The Adoption of XML in Healthcare Systems

Few healthcare information systems vendors have adopted XML for storage and manipulation of data within computer systems at this point, but there is a strong industry push to adopt XML as the preferred format. But even if there isn't widespread adoption among the vendors, as long as standards organizations agree to use XML to exchange data among systems, data that comes from non-XML compliant systems can be converted to XML.

Among most of the newer systems vendors, specifically those working on Internet and Web systems, XML is the preferred format syntax. This trend will expand as the use of XML expands in the non-healthcare computer industry.

As for data exchange, both HL7 and ASTM are defining comprehensive standards for XML in healthcare, as are the European countries through standards groups such as CEN TC251 and the international community through the new international healthcare standards group under ISO TC215. HL7 and ASTM have closely coordinated efforts to define how documents and messages would be tagged and formatted under XML, and HL7 has defined XML as the syntax for HL7 Version 3.0. These standards efforts will strongly influence healthcare systems vendors and accelerate the adoption of XML.

The Future of XML in Medical Records Management

What remains for XML is to begin to define the XSL display and printing format issues. Ideally, we could begin to define a set of standards for document formatting and display within the medical records environment. HIM professionals would be the ideal source for these standards and would have a significant effect on the correct and standardized formatting of XML. If anything, XML and XSL are so powerful that you can almost do too much with them—and if we are not careful, we will end up with an unlimited set of outputs and formats that add to the confusion rather than reduce it.

In this way, XML is like computers—it can simplify or it can complicate. As we have seen, we probably have more paper now that computers are "helping" us, rather than less. In this situation, however, XML may help us find a way out. If we can influence the direction of XML and XSL, we might put an end to the deluge of paper.

Reference

Sokolowski, Rachael. "XML Makes Its Mark." *Journal of AHIMA* 70, no. 10 (1999): 21-24.

***Richard M. Peters** is president and CEO of iTrust, San Francisco, CA. He can be reached at rpeters@itrust.net.*

Article citation:
Peters, Richard M., Jr. "XML: Defining the Transition from Paper to Digital Records." *Journal of AHIMA* 71, no.1 (2000): 34-38.
