

XML.

Even If It Is Snake Oil, You'll Still Feel Pretty Good

By Richard Hammond

If this were a "spaghetti Western" movie instead of an information professional journal, the opening scene might look like this: The trail of dust from a wagon forms a sinewy, sinister ribbon across the prairie. The leather creaks; the horses groan and snort. Entering the frontier town, the snake oil salesman's beady eyes glare at the gathered populace like a buzzard circling for his inevitable meal.

Friends! Gather round one and all! I bring the greatest gift every conceived! XML! That's right, friends. It will link all your data together, and, even better, it'll produce knowledge from that data! Yes friends, just input your data in one end and out the other side will be born Wisdom! What have you got to lose but some more resources, another trip to the boss's office to explain how great this new software really, really is, and how it'll pay for itself in only one Ice Age (if we're lucky)! Come on now, step right up! Who'll be the first to buy a magic XML machine?"

BUT XML NEEDN'T BE SNAKE OIL

From that rather dark and fanciful introduction, it might be difficult to tell that I am actually a big proponent of the idea of markup languages. The eXtensible Markup Language (XML) is no exception. It is a powerful, robust tool specifically designed for manipulating information via digital networks such as the World Wide Web. Although the XML concept sounds intriguing, it

took me a while to fully comprehend just what all that meant. As I have learned about XML, and explained its value to various audiences, I have encountered an understandable lack of clarity regarding the role and purpose of XML as a part of the larger interconnected digital network. After all, the XML standard [www.w3c.org] was not public until 1998, less than a decade ago, which is particularly remarkable considering how has XML changed so many facets of information management.

Although touching on the technical aspects of XML, this article's main focus revolves around managing an XML project. Effective project management does not require expertise in XML. However, because XML requires the development of specific and concise business rules, which in turn requires individuals to discuss specific aspects of their job functions, I have found that the social management of the team is perhaps the most critical aspect of a successful XML project. People react with a wide range of responses when probed about exactly what it is they do all day. Regardless of whether you



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



bring in outside resources or work internally, to complete a successful XML project you will have to form and lead a team—a real team, not the snake oil salesman and his shill in the crowd.

XML INGREDIENTS

While it's unnecessary to be the world's expert in XML technology to lead a successful project team, you should know the rudiments. In the simplest terms, XML provides a mechanism for describing information. When I first encountered XML and markup languages, I described XML as the library card catalog (OK, most people today haven't encountered an actual card catalog, but it made sense at the time). XML isn't the book; it's information about the book. In the database world, this is generally referred to as metadata.

Using the idea of a library, markup language tools would provide the following:

Example 1

```
<book>
<title></title>
<author></author>
<publication_date></publication_date>
</book>
```

Example 1 is the metadata for the data. A great number of books have this basic information attached to them. The <book> is called a "tag." Each tag must be opened <book> and closed </book>.

Before continuing, I want to make a strong note that the examples are the code (pseudo code, but pretty close) and you are looking at the code. One of the basic requirements of XML is that it is "readable," meaning that humans can make sense of it pretty quickly. There is nothing below the text. If you're one of those readers who is saying, "Well, that doesn't concern me. I'm not a programmer," keep reading.

COMBINING PROGRAMMING AND SUBJECT EXPERTISE

Programmers, like computers, have no particular expertise about the topic for which they are programming. Rather, the success or failure of a technology project is weighted heavily toward the topic expert. Only the topic expert can fully describe the products that will best alleviate the problem. In earlier times, it would have been essentially impossible to become an expert programmer just for the purposes of overseeing a couple of projects. The blessing of XML is that applications can be developed very quickly and very specifically. The curse of XML (although I know you'll come to love it as I do) is that the topic expert will have to climb in under the hood to understand what the programmer and the programming can and cannot do. The greater the interaction between the two, the more successful the project will be.

Let's look at Example 1 with information inserted.

Example 2

```
<book>
```

```
<title>This Is a Great Book!</title>
<author>Imso Humble</author>
<publication_date>January 1, 2005</publication_date>
</book>
```

```
<book>
<title>This Is an Even Better Book!</title>
<author>Mr. Mark Markup</author>
<publication_date>January 2, 2006</publication_date>
</book>
```

Hopefully, without needing to understand the mechanics, you can see that you can query the file list. “Show me all books by Imso Humble.” “How many books are there in total?” “Which books were published on January 1, 2005?”

That’s about all there is to the act of creating tags. Now, let’s “extend” (thus the extensible in XML) the code to create more metadata tags. Suppose every electronic book or information resource was tagged internally like this:

Example 3

```
<book>
<title/> (this is just shorthand for opening and closing
the tag)
<author/>
<publication_date/>
<chapters>
<chapter 1>The Story Begins</chapter 1>
<chapter 2>The Story Continues</chapter 2>
</chapters>
</book>
```

TAKING TAGGING TO EXTREMES

To be a bit more radical, suppose it was tagged like this:

Example 4

```
<book>
.....
<chapter 1>
<sentence>The beginning was just the start</sentence>
<sentence>Then the story continued</sentence>
.....
</book>
```

Of course, there is nothing to prevent this:

Example 5

```
<book>
.....
<sentence1>
<POS = "article">The</POS>
<POS=" noun" (gerund?)>beginning</POS>
.....
</book>
```

In fact, with tools such as Xpointer [www.w3.org/TR/xptr-framework], I can find each letter and the space between each letter!

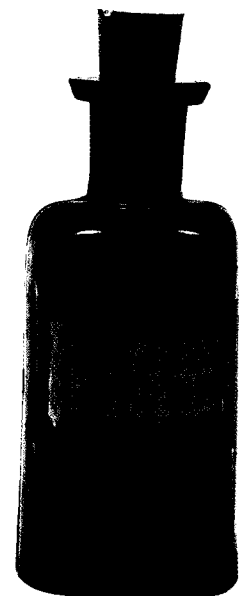
MAYBE THE MEDICINE'S TOO POWERFUL

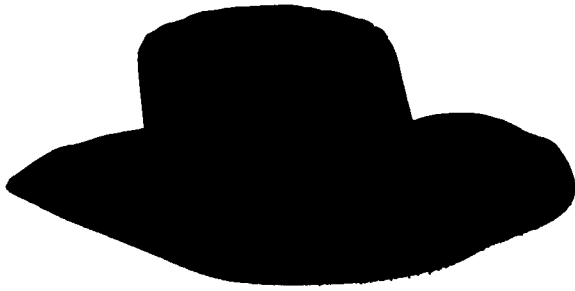
At what level of information “granularity” did you realize that the cost to obtain such precision exceeds the benefit? If you did not reach that point, you could be obsessing on the possibilities of the technology and should consider an immediate vacation! But if you *did* reach that point, the snake oil part of XML should be clear to you. We are using words to describe words. The information is words and the XML metadata is words. The question then arises, if we sit on the back of a turtle that sits on the back of a turtle, what is at the bottom? Who gets to define the words?

As professionals who deal with the categorization of data and information each day, you may long ago have recognized that “bounding” the information is not always easy. In the same way, creating the logic for the tags to bind data into categories ranges from not so easy to downright difficult. The U.S. federal government is currently developing the Data Reference Model, or DRM [www.whitehouse.gov/omb/egov/a-5-drm.html]. Perhaps it should be called *The DRM* because its purpose is to tag every activity of the federal government.

During the difficult tag development phase of an XML project, it is often disconcerting to realize that there is very little output from the team. Team members have a high level of commitment to the ultimate success of the project and must be prepared to engage in “creative abrasion,” a term first coined by Gerald (Jerry) Hirshberg, the founding director of Nissan Design International, and expanded upon in his book, *The Creative Priority: Driving Innovative Business in the Real World*. The idea is that having people with divergent views work on the same project will result in a creative friction. Their disagreements will actually result in a superior final product.

Creating the logic
for the tags
to bind data
into categories
ranges from
not so easy to
downright difficult.





Although rules development is an essentially different exercise than tag development, both share in common the need for the programmer and the topic expert to work side by side.

The final outcome of such an effort will represent the consensus of the team, effective in the broad scale, even if not efficient in the traditional business sense. One role for the project manager and team leader will be to defend the resources required to develop tags for each part of the business process. You will also likely have the role of cheerleader, encouraging your team to see it through to the end, regardless of how much they fight over details.

THE TAGS ARE DONE, MY LITTLE CHICADEE

Once the tags are done, the team can relax, or can they? Your team has figured out what you're going to call each and every item. Ready to go reap some of those great benefits from XML? It's not quite time. However, here's the good news—so far, your only investment is time and effort. The entire project has been on the back of a napkin as far as hardware and software go.

At this point in the process, you have developed a data dictionary, a list of all the words your language will allow. If I were to provide a translation dictionary, say from Spanish to English and back, I would have at my fingertips the definitions for every word I could possibly need. The thing I could not do would be to speak Spanish. Without grammatical rules, the dictionary is only so useful. I tend to call these rules business rules. Regardless of their label, one certainty of grammatical rules is that there are a very few compared to the number of possible useful and useable combinations of words in the dictionary. What rules lack in number, they compensate for in power. Each rule dictates essentially infinite combinations of useful structures. It is no surprise that de-

veloping the syntactic rules for the words is perhaps even more strenuous than the original tagging project. Business rules frequently describe what individuals do for their profession. Different individuals react differently to this process.

Here are a couple of examples of business rules. It is important to recognize that the tools you are reading about are less than 10 years old and have many different proponents working from many different angles. The development process is in full flux. The principles, however, are solid and, in general, will not change operationally for this type of application. The basic tool for adding rules to XML dictionaries is the style transformation sheet, or xslt.

```
<book>
<checkout_date/>
<due_date/>
</book>
```

```
<xslt: <due_date> + 5days>Send email to desk</
due_date>
```

IMPOSING RULES ON THE FRONTIER

Although rules development is an essentially different exercise than tag development, both share in common the need for the programmer and the topic expert to work side by side. Let's try a bit different type of rule just for one more example. Suppose you decided that you want to add a tag to each book that says something like "Age Restricted." The first action is to use the extensible part of XML.

```
<book>
<title>
<age_restricted>Y|N</age_restricted>
...
```

It is then pretty straightforward to write the rule:

```
<xslt: if <age_restricted> = "Y"
confirm age of customer/>
```

What is not so straightforward is what value goes with "Y." The only team member who will know that is the topic expert. It may be obvious, but it always bears repeating. Human capital and social capital are the underlying foundations to any successful technology implementation. What is somewhat unique about XML is its open declaration of that principle. From my view, it is the revolt of the programmers. No longer will programmers try to make subjective decisions about subjects for which they have no expertise! On the other hand, programmers must learn to explain the technologies in meaningful ways as well as really listen to the customer's wants and needs. As well, programmers must do a better job of differentiating between wants and needs. As the customer,

I *want* my computer to give me tomorrow's winning lottery numbers, but I *need* my computer to produce this article.

MOVING TO THE NEXT CURE-ALL

Now that the words are defined and the rules are developed, what's next? To put it simply, what you have is the railroad tracks (the business rules) and the train cars (the words) to run your project. If you did nothing else, you would have established a digital record of attributes about your business, not just the taxonomy of "things" but also the rules by which those "things" relate, the topology of the business. Nothing shady or suspicious about that.

While there are still a myriad of decisions to be made, it is relatively straightforward for the programmers to step in at this point and begin to rapidly produce data visualization tools, potentially using other parts of the XML family such as scalable vector graphics (SVG). Your input is still vital, but at this point, you have defined your business. Mapping it back out is not as procedurally challenging. The end-user tools are developed in weeks, not months or years, and can be altered very readily as your needs evolve.

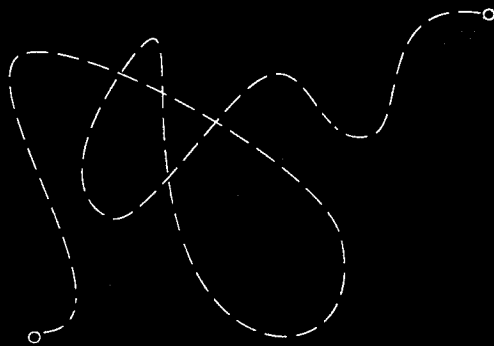
The subject of XML ranges far and wide, from the deeply philosophical meaning of language to the rather mundane moving of digits around the big system. Managing an XML project is a challenge, since you are forced to deal directly with these issues. In order to succeed, you must declare your position on many of them. Additionally, my experience indicates that topic experts are not natural project managers. It can be unnerving to lead a team for the first time. However, by asking for advice and including all stakeholders throughout the process, you will reach a valid consensus between the operational and the technological that will produce a useful and useable product. Don't over-promise, though, lest you be the one accused of selling snake oil. Be realistic ... and maybe there's a bonus in there for you!

Richard Hammond [rhamm123@bellsouth.net] is knowledge management team leader, EPA Region 4. He writes here in a personal capacity. Comments? E-mail letters to the editor to marydee@xmission.com.



e-Resource Management

Integrated Solution:



-
-
-
-
-

TDNet provides technology to help libraries effectively manage, search and utilize their electronic resources at Academic, Research, Medical, Corporate, Public & Government facilities.

TDNet manages collections in hundreds of libraries, single and multi sites, and consortia in eighteen countries.

TDNet partners with ILS vendors, content providers, and content management companies world wide.

Contact us for a free trial: tdnet@tdnet.teldan.com

www.TDNet.com