

XML - A P2P Technology

Michel **Vulpe** <mvulpe@i4i.com>

Abstract

The business of business is business. The business of authors is developing content. Dictionary.com defines an author as someone who is 'responsible for the content of a published text'. It can be argued that, under this definition, in a business where XML is a mandatory output format the author is responsible for the XML. The concept of 'XML authoring' is based on this wishful interpretation. In reality, for most authors, XML is an output format just like Word or HTML or RTF and therefore should be no more complex to be responsible for. When the United States Patent Office decided to adopt valid XML as an output format for patents it made this truism a central tenet of its technology strategy. Inventors, patent agents, lawyers and paralegals need patent authoring support tools not XML authoring tools. This paper will review some the considerations that informed the USPTO's decision-making process.

1. Delivering Competitive Services

The intellectual property industry is, not surprisingly in this information economy, growing at an annual compounded rate of 40%. For the United States Patent Office (USPTO), the dominant player in this industry with more than 40% of the business, this growth means that the average daily workload of 2,000 submissions in 2000 will grow to 2,800 in 2001 and 3,900 in 2002. The USPTO, although a U.S. government agency, is not a cost center. Instead, as with virtually all intellectual property offices in the developed world, it is a performance agency, a government euphemism for a profit center. As such, it is run and managed as the very large for profit bureaucratic monopoly that it is. As with most monopolies in today's global economy, though, it is under pressure to deliver competitive services to forestall interference by its political masters in its operations.

Being competitive, for the USPTO, means, among other things, being able to

complete a patent transaction in two years instead of the customary four. In today's high-speed economy, four years is simply too long to wait to establish the value of intellectual capital. That improvement in service is not only critical to U.S. inventors, it is also critical to maintain the predominant position of the U.S. in the intellectual property business. Inventors worldwide file there to leverage the legal and capital infrastructure in the U.S. that allows them to come to market first. However, if one of the cornerstones of the product strategy 45 'Symbol' patent protection 45 'Symbol' takes too long, alternatives will, out of business necessity, be found. The U.S. is not about to surrender its preeminent position as the world's leading center of innovation because bureaucratic processes cannot be improved.

2. Patent filing: a B2B transaction

The conventional definition of Internet age B2B is one of transactions that are effectively machine-to-machine (M2M). A computer at Aardvark generates a business message:

```
'<offer_to_sell offernum='1234'>  
<seller>Aardvark</seller>  
<quantity> 200</quantity>  
<part_number>A-130</part_number>  
<price currency='US'>$50.46 </price></offer_to_sell>'
```

This message is picked up by supplier FooBar's computer which sends a return business message:

```
'<offer_to_buy ordernum='1234'>  
<buyer>FooBar</buyer>  
<quantity>200</quantity>  
<part_number> A-130 </part_number>  
<price currency='US'>$50.46 </price>  
<delivery_date type='confirmed'>June 15, 2001</delivery_date>  
</offer_to_buy>'
```

If Aardvark's computer agrees that the delivery date, which was not specified in the initial message, is acceptable, a confirming message is sent to FooBar and the deal struck, untouched by human hands.

The system works because this type of B2B activity can be algorithmically described.

Of course, it needs security, non-repudiation and so forth, but at its core, the activity is a simple equation. What makes it work is not the XML, but the simplicity of the transaction. XML is a convenient way of defining the variables in the equation, not a revolutionary new technology that suddenly makes B2B possible. These types of machine-to-machine transactions predate XML by a few generations ATMs for instance.

The process of filing a patent is also a B2B activity. Inventor A is offering an idea/process to the public in return for certain legal protections that allow the unlimited exploitation of that idea for a set period of time. The transaction could be described as:

```
'<offer_to_sell offernum='1234'>
<seller>Tom Edison</seller>
<quantity>1</quantity>
<part_number>clever idea about how to illuminate a
building</part_number>
<price currency='all'>0</price></offer_to_sell>'
```

The problem with this transaction, of course, is what is meant by the content of <part_number>. In the previous B2B transaction, there is a consensus as to what 'A-130' means, in this transaction 'clever idea about how to illuminate a building', of course, is unknown, otherwise it wouldn't be an invention. This B2B transaction requires that P, people, get involved because only they can give meaning to content. It was only when people gave formal meaning to A-130 that the M2M B2B could take place.

3. Patent documents and XML

Patents applications are formal documents. They are highly structured, being made up of numerous content objects that have formal meaning and roles. The validity of a patent application is determined in part by its conformance to structural rules and in part by the meaning of the content. Structural rules are necessary to ensure that the patent application is complete and can be judged on its merits. In order to make that judgement the patent reviewer needs a complete information package that places the invention in context, describes it in a general manner and finally articulates the value propositions in clear and unambiguous terms. Over time the patent industry has

agreed on what it is that comprises a complete information package. This consensus on at least the structure and the identification of the components has allowed the patent industry to develop data models for these documents. Not surprisingly, the industry has adopted XML as the modeling technology of choice.

XML provides the patent industry with some of the same benefits as transactional XML. The patent document can be validated for structural completeness that is, all the required components are present and in the correct order. It can also provide workflow support. Individual components can be identified, extracted and subjected to business processes using a parallel processing model that is more timely than the serial processing model. XML eliminates the need to transform approved patents into a form that can be used by publication systems. Of course, the problem is how to get the patent into an XML form in the first place?

The initial model that the USPTO adopted was the very traditional transformational one. Documents were submitted in whatever form, which converted them into XML a complex, time consuming and costly process. In fact, it cost the USPTO \$25M (US) per year to do those conversions. But there was no other choice because there were no means by which the tens of thousands of patent application creators could create XML. The fact that XML authoring tools were available on the market was irrelevant simply because they were XML authoring tools rather than business process tools.

4. Patent authoring vs. XML authoring

The inventors, lawyers, paralegals, etc. who author patents recognize and accept the fact that they are developing complex documents, both in terms of structure and of content. They are always on the lookout for technologies and methodologies that will ease their workloads and improve their efficacy. XML and its predecessor SGML have, of course, been considered. The technology itself has been accepted. However, the implementation, in the context of authoring patents, has not justified the investment. The explanation for this is simple; the author wants support for complex document authoring, *not* XML authoring. The USPTO estimates that an XML authoring solution, as opposed to a complex document authoring support tool, would increase authoring costs to the patent filer by 50%. This is because XML authoring focuses on XML and thereby requires the author to become XML literate.

Patent authors typically are not interested in the science of documents, or the intricacies of metadata languages; they are interested in authoring patents. To them XML is a background, not a foreground, technology.

The USPTO has no choice but to accept this argument. The support for XML is driven by their internal needs. The argument that it makes their processes more efficient, while interesting, means little to the patent filer. Filers appreciate that it is beneficial to have the USPTO more efficient, but that efficiency cannot be gained at the cost of making the filers more inefficient. It has to be a win-win situation. The USPTO, confronted with this hard-nosed realism, determined that the problem had to be restated. The restatement is framed as an improvement on current business processes, not as XML authoring. That is, the USPTO agreed to provide the filer with a tool that would improve their patent authoring process. In short, it had to give the filer something that directly benefited the filer. That tool is the Patent Application Specification Authoring Tool, or PASAT®, a software application which provides patent authoring support inside the Microsoft® Word framework.

PASAT is built on i4i's S4/TEXT® product, which distinguishes itself in a number of ways. For the USPTO the most important distinguishing characteristic was that it is not an XML authoring tool. Rather, it is a framework for developing complex document authoring support tools that work inside Microsoft Word. XML is part of the framework, not the end game. The PASAT product hides all the XML from the user and provides, through the services of Microsoft Word and the Office Assistant, prompts and guides based on the business rules for creating a patent application. The fact that the S4/TEXT underpinnings to PASAT provide real-time XML support for a complex DTD is totally transparent to the end-user.

To make XML user friendly, the focus has to be on supporting the activities of the user, and those activities are not XML authoring but complex document authoring. The key is to frame whatever XML dialogue cannot be hidden as appropriate questions which support the users perceptions of what they are doing. Very simply, if the business problem is that graphics are required to illustrate a point, the dialogue and behavior of the system should be structured around the business problem of selecting the appropriate graphic, not around which XML tags and attributes to enter.

For example, in the case of patent applications, artwork to substantiate claims is, according to editorial rules, always at the end of the document. The XML DTD is

structured to reflect that. However, from a business process perspective, artwork is a part of individual claims. The XML DTD is consistent with the editorial rule, but inconsistent with the business process.

To overcome this, the business dialogue presented to patent lawyers when they are developing claim content includes the option to include appropriate artwork. The system in this case, PASAT creates the valid XML and opens the appropriate File/Finddialogue box on behalf of the users. At no point in the process are users presented with <artwork> tags or any of the XML logic. The behavior of PASAT is such that the question INSERT CLAIM ARTWORK? is a business question, not an XML question. To users this simple expedient significantly changes the perception of the value proposition. Patent lawyers are not interested in the <artwork> tag; they are interested in inserting the artwork to substantiate a claim. The PASAT complex document authoring support system assists in that process.

PASAT provides, within the Microsoft Word GUI, a business interface to the patent author. PASAT uses the S4/TEXT platform to access the XML logic that drives the business interface. S4/TEXT is able to provide the XML access in a tagless mode, thus saving the end-user from becoming an XML author. Traditional XML authoring tools, because of their XML focus, have an unfriendly user interface that is, the XML tags are displayed in-line with the content, creating a visually unappealing interface. The expectation is that the user wishes to interact with the XML. The business interface is one that frames the user interaction as business decision support. As such it does not slavishly follow the XML rule set. It recognizes that the business processes can involve data identified by numerous XML tags which, for editorial or other reasons, are not serial, and that the reasons are not consequential to the content creator. Most importantly, it recognizes that the content creation system, not end users, should bear the burden of the XML

5. Conclusion

i4i calls this type of interface 'Tagless XML'. From a technical perspective, this is totally misleading. XML is about tags and rules for the application of those tags. Tags provide computers and software applications directives about content. This is the central value proposition of XML and why it has been so successful in B2B, B2C, and

C2B applications. It is worth noting, however, that all these applications are primarily C2C computer to computer. When computers talk to computers they do not care about look and feel, visual clutter, or inference. Precision takes precedence over these human issues.

Tagless XML is for humans. Humans care about look and feel, visual clutter, and imprecise business processes. Tagless XML recognizes that if humans are to interact with computer systems using XML as the language of communication, then XML had better learn to present itself in a people friendly way. The USPTO selected i4is S4/TEXT solution because it concentrates on making XML people friendly by:

1. connecting Microsoft Word user interface events to presentation-orientated XML, thus eliminating the requirement that users concern themselves with this level of detail
2. combining business process logic and XML logic into a single business support interface which delivers business value to users;
3. eliminating the visual clutter of XML by configuring the business support interface to recognize the imprecision of content development; and,
4. never forcing users to become the machine.

This approach goes a long way towards making XML people friendly.

6. Closing Comments

The official electronic filing system (EFS) went alive on October 27, 2000.

The first electronic filing of a patent using EFS was filed on Monday, October 30, 2000.

Total seats to date: 24,806

Biography

Michel **Vulpe**

CTO and Founder

i4i

Toronto

Canada

Email: mvulpe@i4i.com

Michel Vulpe - Michel Vulpe invented the patented S4 Technology in 1993 and is the founder of i4i. A native of Montreal, he attended the University of Victoria and then graduate school at the University of Toronto, where he studied political economics. He first learned computer programming in the R and D group of Geac Computer Corporation Limited in Vancouver, and helped further the understanding of distributed text and transaction processing over distributed databases.

Michel is an acknowledged leader in creating innovative products permitting real-time collaboration and interaction in the workplace. During a two-year sabbatical at the Smithsonian Institute in Washington, D.C., he developed a system to track all 120 million items in the various collections. In 1989 he authored 'CMASS,' a process methodology for complex data, and in 1996 he wrote the international standard SEMI E36; both are standards for business-to-business activities in the semiconductor industry