PASCAL

The Wrong Stuff

A few years ago I gave a seminar for IDUG in Toronto. It was just about when the XML fad started, and vendors were beginning to step over each other, claiming to be *the* XML company. Waiting in the hall for my session to start, I overheard a keynote speech by IBM's Nelson Mattos, including XML, whose pronouncements made my eyes roll.
Recently, C. J. Date drew my attention to *The Right Stuff*, an interview in *DB2 Magazine* (quarter 1) with "Nelson Mattos, Ph.D., an IBM Distinguished Engineer and newly minted vice president of information integration (a recent promotion from his role as director)". No surprise there, after all, jumping on fad bandwagons is a sure way to promotion. The purpose of the interview was "to help readers understand just what the DB2 II v.8.2 release means to developers and DBAs, and to give us a preview of what's yet to come." This should be debunked, said Chris.

Now, it is always a bad idea to put engineers in charge of languages, integration, modeling, design, or anything of the sort; that's like hiring a building contractor to architect your house (if you want an example of a language designed by engineers, look no further than SQL). What is more, integration is the motherhood and apple pie holy grail of the IT industry – everybody is for it, even though few can specify with any precision what they really mean – so marketeers have a field day with it. And engineers learn marketing jargon quite fast: fuzzy, often meaningless terminology, with a plethora of acronyms to impress the uneducated masses (see *Comments On a Jim Gray Interview* (www.dbdebunk.com/page/page/638948.htmI) in the past one could expect a bit more from a graduate education, but that is no longer the case (see *The Chasing of Mayflies* (www.inconcept.com/JCM/October2003/Pascal.htm), and *The Myth of Market-Based Education* (www.dbdebunk.com/page/page/622648.htm).

Consider the following from his latest interview:

Masala delivers, with the OmniFind Edition, a brand new interface to give users plain-text access to information across the enterprise. We significantly enhanced the performance of federation, in typical scenarios, by several orders of magnitude. We delivered a brand new architecture for replication focusing specifically on low-latency integration. We expanded information integration with event publishing, making the platform support not only a "pull" model but also a "push" model: DB2 II Event Publisher can monitor events and push those events to subscribers using an XML messaging infrastructure.

Also, jointly with the delivery of Masala, we closed the acquisition of Venetica, which gave us a new set of interfaces so that content-centric applications can be built using information integration. It also gave us connectivity to primarily non-IBM content repositories, such as Documentum, FileNet, Interwoven, and so on.

Got that? Doesn't this "help you understand"? Isn't "integration" great?
Data ownership is a major concern, particularly in the large IT infrastructures. One of the biggest benefits of DB2 II is that it doesn't need to take over control of the data. With federation, you're not copying the data; you just need to access the back-end system to deliver the data to a specific application. So you basically eliminate any concerns about data ownership.

Power equivalence between SQL and XML cannot be true

And if you believe that, I have a bridge in Brooklyn to sell you.
As you probably know, IBM has been investing in enabling DB2 products with XML technology. First, we delivered the DB2 XML Extender; more than a year ago, we delivered some XML capabilities in alphaWorks that gave customers the ability to map XQuery to SQL and run it against both DB2 and DB2 II. And, we will be announcing the beta of our XQuery implementation, as well as a brand new, fully native implementation of XML storage in our DB2 infrastructure. So, that will give DB2 UDB customers the ability to very efficiently store and manipulate XML data in their databases.
Because DB2 II uses the DB2 infrastructure, it will also include the XML storage and XQuery capability once it's available in DB2. Our plan for follow-on releases will allow DB2 II customers to use an XQuery data access paradigm to access any information. The native storage of XML documents can be used by DB2 II customers to provide efficient caching of data in back end systems in native XML.

Points arising:

- If you ever wondered what XML is good for, here you have it: to map XQuery to SQL, so that you can query databases of *two different versions of the same product by the same vendor*. Now, *there's* real integration and progress for you!
- "XML *storage*"? "Store and manipulate *efficiently*"? Do I

detect the logical-physical confusion (manipulation is logical, storage and efficiency are physical)? But then, shouldn't we expect just that when what is essentially a physical data exchange format is shoved where it does not belong, in data management?

- Data management is *logical structure, integrity and manipulation* – that is, some *data model* – and is about *inferencing*; text is about searching. *What is* the structure and, therefore, integrity and manipulation, of any text? Are these the same for contract, email, or article text? Codd was smart enough to realize this, cohorts at IBM – the present, and even the past – are oblivious to it.
- Brother, can you spare me a paradigm? How indiscriminate is the use of big words these days. If Kuhn is dead, he is probably rolling in his grave.

Because DB2 is used in the DB2 Content Manager products, the XML infrastructure can very efficiently store metadata about the content maintained in Content Manager repositories. This is a major breakthrough in technology, and one that will speed up convergence of the world of structured and unstructured data. I think this XML technology will speed up that convergence even further, because XML gives the ability to effectively describe structured and unstructured data. XQuery is also richer than SQL, which is restricted to the relational model. And XQuery provides some of the same concepts that you would find in content repositories today, like hierarchical data structures. XQuery allows you to navigate through hierarchical structures in the same way that a content application would navigate through folders. XQuery has all the power that you find in SQL.

At some point in during his doctoral studies somebody ought to have told Mattos that there is no such thing as "unstructured data"; anything unstructured is random noise, *not* data – see *Unstructured Thinking* (www.dbdebunk.com/page/page/626997.htm) and *More About Unstructured Thinking* (www.dbdebunk.com/page/page/766119.htm).

"Convergence" my foot. XML "richer" my foot. The relational model is restricted to SQL, not the other way around. Is "access" to documents the same thing as "access" to data? XML-formatted data is not just text, but rather *hierarchically structured* data, which *can* also be text content. Now there are *two data models* (SQL's poorly specified, chockfull of relational violations, and XML's hierarchy) and *two data languages* to learn, choose from, and use, and to what advantage? And if XQuery can be mapped to SQL, why do you need both of them? Anyway, C. J. Date points out that power equivalence between SQL and XML cannot be true.

During my graduate studies, nobody paid any attention to what you were saying or writing if you did not familiarize yourself with the history of the subject first; that's how scientific *progress* is achieved. That is no longer the case, but had it been, Mattos would have known that there was a *reason* to Codd's invention of the relational model *just when the reigning products were hierarchic*. Shouldn't somebody involved in data management technology be aware of, and understand that reason? *At IBM*, of all places?

The consequence of these new technologies and what we have delivered so far is an increased virtualization. We will get to the point where consumers and applications developers working in an II infrastructure will have literally no need to know where the data is coming from.

Ah, "virtualization". Another Brooklyn bridge for sale. I can only quote Dijkstra again, who deplores
... the widespread sensitivity to fads and fashions, and the wholesale adoption of buzzwords and even buzznotes ... promis[e] salvation, make it a "structured" something or a *"virtual"* something, or "abstract", "distributed" or "higher-order" or "applicative" and you can almost be certain of having started a new cult.

To which list I would add integration, convergence, infrastructure, architecture, paradigm, federated, seamless, and more.

Be suspicious of those who insist on attaching a degree to their name.

Fabian Pascal
Fabian Pascal is onafhankelijk IT-analist, consultant en auteur gespecialiseerd in data management. Zie ook zijn website www.dbdebunk.com