The Feasibility of an Operation-Centric Environment for Processing XML Documents

Michael Champion
Mike.Champion@softwareag-usa.com
26 January 2001

Background

1. RDBMS theory has proven as successful as it has in large part because it is "operation-centric", i.e., depends solely on operations on a well-defined mathematical structure, the relation. It was a breakthrough compared to the “structure-centric” approach based on hierarchical or network databases. The operations and operands of relational theory are mathematically well defined, so mathematics can be rigorously applied, and this has allowed a consistent body of rigorous methodologies, query languages, and development practices to become almost universally understood.

Before the "relational revolution", developers had to understand the details of the data structures on which they operated in order to access the content. While it is quite true that B-Trees (or ISAM files, or whatever) could be represented with the mathematics of graph theory, they had no mathematically well-defined operations -- for example, a formal operation such as "next" is meaningless unless one understands the structure being traversed.

Relational theory introduced the concept of "data independence" in that the select, project, join operations in the relational algebra do not need to know anything about the structure of the data (the order of the columns, for example) in order to work. This allows analysts, designers, and programmers to do their jobs without reference to the underlying physical data structures, and for database administrators to reorganize, optimize, and distribute physical databases without "breaking" applications. It is the job of an actual database management system (Oracle, DB2, SQL Server, etc.) to map the abstractions of the relational model and SQL onto actual data structures and machine code.

But RDBMS theory and applications only provide data independence for the user, it does not offer data independence to implementers for operating on data in secondary storage. The theory only applies at what might be called the "view" level seen by users, whereas the details of how the operations are implemented on a computer are strictly up to the human programmers developing the DBMS itself. Relational theory only models the user view of data outside the machine, and says nothing about data within the machine.

DBMS systems are still plagued by performance problems because of the rigidly structured nature of the underlying physical representation of data. They work as well as they do in the real world partly because of much "voodoo" being performed by implementers and administrators, not to mention constant tweaking of queries and schemas by application writers. Furthermore, the “object-relational” systems in actual production use today often encourage de-normalization of data via support for features such as “repeating frames” that sacrifice the formalisms of the relational model in the name of performance.

2. The mathematics of the relational model is based heavily on classical set theory, CST, and this is both its strength and its weakness. For example, limitations such as the ability to talk meaningfully only about flat tables, stem from the fact that classical set theory blurs the distinction between sets with “ordered” elements and sets with “nested” elements. Thus, operations become ill-defined when extended to represent and manipulate sets with both ordered elements and nested elements. D L Childs developed an "extended set theory", XST, more than 30 years ago that adds an additional parameter known as a "scope" to the membership condition of classical set theory. In CST, membership is based on only an element component; in XST membership is based on both an element component and a scope component. This extended membership condition can be used to model ordering and containment relationships that are simply too "messy" to handle in classical set theory and the formalisms (such as relational algebra) that are based on it.

More specifically, XST allows much richer mathematical objects to be modeled, to the point where XST can be used to formally represent and manipulate the "real" operations of a computer in a useful way. This allows us to rigorously map the operations and operands of relational algebra onto a model of a physical computer to extend the concept of data independence from the level of the user all the way down to the level of the secondary storage.

To put it somewhat differently, classical set theory deals with a membership concept in which something is either there or not there, and this is simply not rich enough to provide a useful model of an *ordered* environment such as a computer. Extended set theory gives an additional membership dimension that allows one to model structures and operations both at the level of abstraction provided by relational theory and at levels of detail required to represent what actually happens on a computer.

With this mathematical foundation, one can model and control performance, rigorously optimize the speed/space tradeoff rather than relying on programmer's expertise, partition processing across processors, etc. Strong data independence between abstract operations and physical structures even allow one to change the underlying structures dynamically ...without changing the top-level operations. Over the years, [XST-based] mathematical tools have been developed and implemented in software. Many companies have used these over the years to achieve optimal performance and resource usage in large-scale applications.

XSP Technology for Processing XML Documents

XSP Technology is a formal system for specifying mathematically sound operations and operands that can be executed by a digital computer.

XSP Technology consists of three separate formal specifications:

> **XST: Extended Set Theory** - Formal axiomatic specification of extensions to the foundations of Classical set theory that are necessary to support the modeling of computer based operations and operands.
>
> **XSN: Extended Set Notation** - Formal set theoretic notation expressing operations and operands of XST that preserve the mathematical identity of all conceptual and computer-based structures being considered for a system.
>
> **XSP: Extended Set Processing** - Formal specification of a system of XSN defined operations and operands that can be executed by a computer.

In applying XSP technology to real-world problems, the "well-formedness" criterion of XML means that XML data can be effectively modeled with extended set theory (XST). In other words, an XML "document" is, by its very nature, a well-defined extended set. Unfortunately, the current processing models applied to XML are “structure-centric”, i.e. they require the accessing of the content through knowledge of the structure. With this semi-structured approach, users and developers are not able to exploit the underlying mathematical integrity of XML in a powerful way. Indeed, "old-timers" introduced to XML tools for the first time sometimes observe that XPath reminds them of the pre-relational days when one *had* to understand the structure of a dataset in order to effectively query and manipulate it.

Nevertheless, the world is producing more and more XML data, and the limitations of relational theory for modeling are becoming more and more apparent. RDBMS theory says, "look at the world our way (as normalized tables) and you can use our math". If, for whatever reason, one is forced to look at the XML world through RDBMS lenses, this doesn't help much. XML's data model strength, in which ordering and containment relationships are indeed significant, does not map easily onto the relational view of the world. Recall, however, that these are exactly the kinds of relationships that extended set theory has proven itself to handle. In other words, XSP technology says, "here's some math that defines what a computer does, it can be used to model your view of the world".

Previous XSP research has shown the feasibility of preserving the operation-centric and data independent advantages of the relational data model (RDM) by formally modeling XML document types as extended sets (Xsets). XSP technology is a practical reality to provide the operational capability to capture and process XML document types as extended sets.

XSP Technology also goes beyond the formal underpinnings of the relational model to describe the physical implementation of the operations and the representations of data within a real computer. This has great potential to bridge the mathematically clean, but reality-challenged world of the “pure” relational model and the messy, but more pragmatically rooted world of XML. As the concepts of XSP technology become more widely known, they can be applied and expanded by widespread experience and experimentation.

In a nutshell, XSP technology offers the equivalent advantages of mathematical formalism, which enabled the "relational revolution" over 20 years ago, to be applied to the XML world of semi-structured documents.