

The Data Administration Newsletter (TDAN.com)

[Robert S. Seiner - Publisher](#)

[Main Menu](#)

[New Articles](#)

[Article Archive](#)

[Special Features](#)

[What's New](#)

[Contact the Publisher](#)

AND WHERE DO XML TAGS COME FROM?

Michael Gorman - Whitemarsh Information Systems Corp.

[Return to the Article Archive](#)

The Question

"Which is the better source for XML Tags, an IT System Centric or Data Centric Approach?"

Here are two alternatives. The first was a proposal that was to be adopted by their highest levels of management of an enterprise. The second was created by Michael Gorman as a preferred alternative.

Alternative 1: Information Technology System Centric

Alternative 2: Data Centric

IT System Centric Alternative

If you have an IT System (MIS, Package, etc.) and it does not have to share data with any other Information Technology system then, you, as the Information Technology System manager are 100% in charge of the XML Tags, and you are also in charge of all business fact value sets where ever the value sets are restricted.

Else, if the Information Technology System has to share data with another Information Technology system then for those shared data areas then the two Information Technology System managers are 100% in charge of the XML Tags, and also in charge of all value sets where the value sets are restricted.

These XML tags then are then imposed on all creators of the Information Technology system so that every data exchange must extract and transform the "raw data" into the XML Tagged sets.

"Data Centric" Alternative

All the data in all the Information Technology systems are to have their names and their restricted value sets defined and managed by the highest level functional experts in the enterprise such that when either the functionally defined names and values sets are employed, then, their XML tags and values are standard across all Information Technology systems that may employ them regardless of whether there are any plans for these systems to share data.

Names here include both container names (e.g., table or schema), or fact names (e.g., columns). All

fact names are taken from enterprise-wide standard data elements (now acting as business fact semantic templates for the fact names). Finally, that all "words" that comprise data element names are standardized within their domains (e.g., engineering, business, medical) via standard name-strings, definition fragments, and standard abbreviations. Finally, that all data element value domains are standardized.

"Highest level functional experts" are those persons designated by the enterprise to be the subject matter experts (SME) in that functional area, and thus, it should be their responsibility and authority to define the XML tags that should be deployed in what ever information systems that may employ their data.

Thus, by pushing to the responsibility and authority for XML tags to enterprise-level subject matter expert, you are explicitly removing XML tag development authority from the designers and builders of information systems that may employ data from multiple functional domains. This would have the consequence of having the same tags across information systems regardless of whether those information systems currently were exchanging data.

These XML tags then are automatically created by application programs that are processed through SQL:2003 DBMSs that conform to the soon to be finished and ratified SQL/XML Part 14 that is under very intense implementations by the current set of SQL vendors (read: IBM, Oracle, Sybase, and Microsoft).

Responses

Responses were received from the following individuals:

- Peter Aiken
- George Brennan
- Henry Feinman
- David Hay
- Corinna Martinez
- Terry Moriarty
- Dwight Seeley
- Paul Tiseo
- Richard Warner

Peter Aiken

In my humble opinion, there is the correct answer and a practical answer. First, the correct answer is, of course, Alternative 2, the data centric solution.

With all of its hype XML does permit organizations to realize two distinct advantages. First, wrapping data in metadata permits business users the opportunity to become more involved in the use, valuation, and understanding of how data is used to support their business processes. Second, because of the XML buzz these projects are now face easier approval.

Having data centrally managed helps organizations take advantage of the 20-40% savings in IT costs that can accrue from the data centric solution development - in this case, aided by XML-based integration.

The practical answer may be that the effort does indeed need to be grown from the bottom up. However, bottom up growth strategies must be done as if they were architected from the top down.

This is somewhat akin to (I think) David Parnas' admonition that program documentation should appear as if it was originally developed before the programs are written but knowing that documentation is always done after the fact.

I've witnessed many data groups who were inarticulate about the data centric approach. Their management then lost faith in their ability to deliver, causing the choice of an expedient solution.

In summary though, if there is a top-down requirement to be interoperable, then, given a quality approach for data standardization, and the ability to have an enterprise-wide multiple-level distributed meta data repository, then Alternative 2 will greatly enhance the definition and use of XML tags.

George Brennan

Ignoring for the moment the questionable logic of 2 people being 100% responsible for the same thing. At first, the system centric will be easier and the data centric will be more difficult. But enterprise wide, each additional system 'share' makes the system centric approach more and more complex.

Taking an enterprise view there can easily be a dozen or more systems sharing subsets of the same enterprise customer or client data elements and it becomes a major exercise just to identify shared ownership.

A key goal of an upcoming seminar where we expect 15 to 20 data architects to come together from 14 countries is to move away from country and system 'stovepiping' concepts into global enterprise concepts as would be espoused in Alternative 2.

From experience, I'd have to say system centric wouldn't fly in any typically complex business, but data centric would.

Henry Feinman

Alternative 1 is simply requiring agreement on the format for "interface files." With Alternative 1, IT System Centric it looks like we're back to square-one. Because we hear buzzwords like XML, XQL, bandied about there is a belief that we are doing something fundamentally different from anything done in the past. This is really no different from when Information Technology first confronted with the problem of one program communicating with another program half a century ago.

What central data management made possible was the creation of flexible and resilient data structures. With these structures some of the business was removed from programs and declared using a centralized software whose job it was to ensure the declarations were enforced. We now have business constructs of large complexity and high accuracy enabled in many information systems, largely due to central data management. The theoretical potential of declarative business constructs through central data management has barely been tapped.

If central data management is bypassed by those keen to jump on the XML bandwagon, then the situation will be identical to that which has given rise to the enterprise application integration (EAI) industry. This industry arose due to the need to retrofit flexibility, resiliency and integration onto a fragile spider's web of interdependent activities communicating in unique languages developed for point to point purposes.

What if the schema declaration is bundled with the data? The problem remains the same. Those whose only concern is to get the point to point function operating are highly unlikely to construct a flexible, resilient, integrated information communication architecture. The business likely has professionals on staff trained to build such structures and should make use of them, if it values flexibility, resiliency, and integration.

David Hay

I don't really quite understand why the issue would come up. Actually, I do. XML has been greeted with so much hype that many people don't really understand what it is. It is a language for data interchange. It is not a database language. By definition, this means that to use it, you must have two parties that wish to send data back and forth. At its most primitive, that means that the two parties must agree on the definitions.

I have been amused by those who say, "Gee, all we have to do is to agree with our vendors on a format and we can send data back and forth!" If you could have agreed on the format, you could have sent data back and forth years ago. XML is not the issue here.

Actually, getting large numbers of people to agree on a format is hard. Especially since they are looking at XML code when they are discussing it. A better strategy would be to get the players to go in on the development of a data model encompassing the subject areas involved. This means that they have agreed not just on the terms, but on the meanings of the terms as well. Once they agree on that, the XML formats can be easily derived.

If XML is to realize its potential, it will inspire large groups of people (like entire industries) to agree on the meaning and the format of their communications. Some industries have already done this, to good advantage. To the extent that a company takes a more parochial view, it will encounter all the problems its parochial view generated in the past.

Corinna Martinez

I don't believe that it is an all or nothing option. Some systems are tactical in nature, and yes they seem to survive much longer than they should, but if the system is not required or expected to 'share' data then why would you go through all the trouble of making all its entities tagged and interchangeable?

I personally prefer Alternative 2, data centric, because more and more there are less tactical systems being developed but I wouldn't want to lock the option out.

My vote is for the Data Centric alternative without constraining the development effort too much. This can be readily defined by the organizations because not all businesses or business processes are complex.

If you truly embrace Alternative 2 then traditional ideas of workgroup must change from IT/User to Collaborative Teams that are comprised of people from various areas within the Organization with plenty of checks and balances being done throughout the effort with folks outside of the Collaborative Team. And, if you had collaborative teams, then making systems, whether they share data or not would require less training, quicker staff learning, and easier to construct documentation and user guides.

Ideally, the collaborative teams would consist of at least one of each of the following: information architect, Information Technology systems manager, enterprise architect, and Super subject matter expert who review the information, come up with the standards, make the recommendation on who should be the keepers of the tags, advocate their recommendation, teach the keepers of the tags how to do what needs to be done, and occasionally review that it is being done.

Terry Moriarty

I agree that Alternative 1 is a stove pipe mentality, which I believe is the type of thought process that has caused so many problems when the data warehouse and integrated view of customer came on the scene. No one probably thought to check if systems could share data until someone wanted to

build an integrated view of customer. Maybe there is actually no overlap in the actual data instances between the two systems, but there certainly is an overlap in business concepts. Maybe there currently are no customers that have both a checking account and a savings account (it could happen), but both checking accounts and savings accounts have current balances or tax identifiers or gender codes or open dates. These represent shared data.

Sharing data is really about sharing business concepts and designing our data to consistently support the shared business concepts, so if suddenly, in the future, the same person does open a checking and a savings account and we want to know that, we don't have to redesign all our applications to get the ability to share the data.

It's the sharing of business concepts that leads to the need for consistent naming, not the sharing of data.

One last point. I don't see how creating XML tags is any different than creating names for data in any programming language. Within a data exchange environment, XML can be viewed, primarily, as a data definition language that HTML (the programming language) uses in creating its forms, in exchanging data between web servers and clients, and in exchanging data, in general.

Through XML, HTML gets re-usable data structure definitions, just like COBOL gets when its DDL is placed in a copybook library or Java/C++/ <select your more modern language here> uses object class specifications from some sort of an includes library. So, I see XML as just another way of describing a data structure to a programming language. Yes, it is much more powerful than COBOL or SQL DDL, but it's still a DDL.

So, if your organization has a standards approach for naming (creating tags) for other technologies, why is a special approach needed for XML? Shouldn't the existing naming process be extended to support XML? If your organization doesn't have a standard approach for naming (creating tags) for other technologies, creating one for XML is a good start. But if it's only done for XML and not in conjunction with how this data is named and defined in other technologies that may interact with XML (generate it or read it), then you're still going to have a disparate data problem.

In short, if the subject matter experts have participated in the common business concepts, then they should also govern the technical exposition of the XML technical constructs.

Dwight W. Seeley

Under every scenario I'd have to support Alternative 2, the data centric approach. The reason is that tags may be used for more than just an exchange of information. Tags may be used by search engines to associate information.

In Database Management Systems, we centralize management of the corporate lexicon in the Data Architect function. In publishing environments (where SGML tags are used extensively), the lexicon is such an important management function that it is placed under the control of the Editors. In almost every instance, management of the human lexicon has a centralized aspect.

We do this because we know that perhaps someday we will need to find something and when we do, we will need to know what that thing was called, and what it meant. What things get called/named is important. While the actual name of a datum or a name contained in a tag used to identify some larger abstraction (i.e., a paragraph, a graphic, a book, et al), should be created by a Subject Matter Expert, that name must be maintained within a taxonomy that allows for grouping and association of like concepts. This allows for information exploration from the general to the specific. This activity should be under some centralized authority.

The Data Centric view is the only avenue that minimizes the impact of the inevitable: change.

Paul Tiseo

Certainly there are a couple of short-term, tactical projects where the need for getting something out quickly out-weighed many other requirements such as the burden of imposing enterprise-wide standardization just to support short-term, transient data exchanges.

Notwithstanding, I lean towards the data-centric approach as a general policy that I'd like to see in place in any medium or large organization with the caveat that the "...all the data in all the Information Technology systems..." stipulation might end up being very constraining if the process used to account for and to define names and value sets (DTDs) isn't responsive (agile?) to enterprise needs.

If any architect or implementer doesn't have to agonize over data definitions or business rules that have already been defined by someone else in the organization, then yes, there is a definite advantage to that. It should be obvious to anyone who works in implementing more than one business system for one enterprise.

My caveat is of a broader perspective. The scenario I see is this: If someone is the first to perform the investigation into a set of data items and their system-based rules, then, is the cost of whipping those out, quick-and-dirty, outweighed by the cost of having the system development effort delayed by the process required by enterprise-wide level definition and recording effort? For sure there will be a delay, the question only is, by how long will it take and how much will it cost?

The enterprise-wide-perspective would definitely be good for subsequent rounds of development. But, engaging that process might not be cost-effective for the first-time implementation. Thereafter, however, both the subsequent rounds will be delayed and also the first round will have to be changed and retrofitted. Maybe there could be a compromise between the delay and cost of preparing in advance for data sharing and the cost of retrofitting systems when the need for data sharing arises.

Richard Warner

The specification of the tagged data in XML document specifications is, like the data requirements for relational or dimensional databases, based on the business needs as elicited from the SME. The design of the tags themselves is the responsibility of the information architects responsible for developing the specifications, This is not a job that can be dumped on subject matter experts alone.

In addition the business requirements, considerations such as the existence of relevant industry standards (the CIDX Chem eStandards for those of us in the chemical industry), reusability (ideally, core components), well-documented metadata, normalization of the tags within their documents, and alignment of the documents themselves to their business function. These are considerations which need to be addressed at an enterprise level, but not (shudder!) left solely to subject matter experts, let alone (shudder, shudder!) to the Information Technology System Manager, who, in the reasonable effort to optimize the XML specification for the system will make minimize its usability by every other system in the business.

Michael M. Gorman, President of Whitemarsh Information Systems Corporation, has been involved in database and DBMS for almost 30 years. Mr. Gorman has been the Secretary of the ANSI Database Languages Committee, X3H2 for 19 years. X3H2 standardizes SQL. A full list of

Whitemarsh's clients and products can be found on the web site, www.wiscorp.com. The goal of the web site, WisWeb, is to make data management books, courses, methodologies, software, and metrics available to the database community through electronic publishing and downloading. WisWeb memberships are very reasonable and are designed for the individual, the ISD organization, universities/colleges, and professional training organizations.

*Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716-1518 USA
Phone: +1.301.249.1142
FAX: +1.301.249.8955
Email: mmgorman@wiscorp.com
WWWeb: <http://www.wiscorp.com>*

[\[Return to the Article Archive\]](#)

**The Data Administration Newsletter (TDAN.com)
Robert S. Seiner - Publisher - rseiner@tdan.com**