The Myths of “Standard” Data Semantics

William C. Burkett
Product Data Integration Technologies, Inc.
444 W. Ocean Blvd., Suite 1800
Long Beach, CA, 90802, USA
August 2002

Introduction

Much of the literature heralding the benefits of XML has focused on its application as a medium for application interoperability. With (a) the Internet as a platform, (b) web services as the functional building block components of an orchestrated application, and (c) XML as a common data format, applications should and will be able to seamlessly and transparently communicate and collaborate without any human intervention. All that is needed to bring this capability into reality is (d) for everyone to agree on and use XML tags the same way so that when an application sees a tag such as <firstName> it will know what it means.

This intuitive understanding makes a lot of sense – which is why so many organizations have sprung into existence to create their own vocabularies (i.e., a set of tags) to serve as the “*lingua franca* for data exchange in the <*insert your favorite industry, application, or domain*>.” This intuitive understanding is so pervasive that it is even a key part of the U.S. GAO recommendations [1] to Sen. Joseph Leiberman (Chairman of the Committee on Governmental Affairs, U.S. Senate) on the application of XML in the federal government. This report warns of the risk that:

> “…markup languages, data definitions, and data structures will proliferate. If organizations develop their systems using unique, nonstandard data definitions and structures, they will be unable to share their data externally without providing additional instructions to translate data structures from one organization and system to another, thus defeating one of XML’s major benefits.”

The perspective of these efforts is that the standardization and promotion of the data element definitions and standard data vocabularies (SDVs) will solve the application interoperability problem. Unfortunately, this intuitive understanding – like many intuitive understandings – does not survive the trials of real-life application because important (and seemingly trivial) assumptions are poorly conceived. The purpose of this paper is to examine some of these assumptions and articulate several myths of “standard” data semantics.

For the purpose of this paper, the term “namespace” will be used to denote a domain in which a vocabulary (i.e., a set of data element names or tag names) is used with a particular meaning and purpose.

Context of article

"Context" plays an important and largely unrecognized role in the data semantics (a topic which is discussed further, below.) Similarly, context plays an important role in understanding the position expressed in this article. Therefore, in order to set the context for the comments expressed herein, let me briefly summarize the perspective from which they originate.

These comments come from a different perspective than that of most practitioners in the field of system integration and application interoperability. Rather than computer science, my background is as an industrial and systems engineer. I am more concerned about the overall "system" behavior of integration and interoperability, and about the human element within the system. I also have 20 years of experience in the development of schemas and the standardization of data exchange schemas within the International Standard Organization (ISO). This latter experience, in particular, has made me acutely aware of the pitfalls confronting schema developers and the "standardization of semantics".

Myths of Data Semantics

Several authors have described the dangers and problems of standardizing data semantics. Perry [3] warns of the willful misuse of SDV by "gamesters" that exploit the semantic play in language (e.g., Bill Clinton's use of the term "sexual relations"). Knox [2] cites the same problem with the GAO report as described above, observing: "Except for the most rudimentary concepts, the assumption that XML-defined models will be the same from one industry to the next, from one country to the next or from one point in time to the next is naïve."

The notion that data semantics can be standardized through the creation and promulgation of data element names/definition or vocabularies is based on several assumptions that are actually myths. These myths are:

***Myth 1*:** Uniquely-named data elements will enable, or are enough for, effective exchange of data semantics (i.e., "information").

***Myth 2*:** Uniquely-named data elements will be used consistently by everybody to "mean the same thing."

***Myth 3*:** Uniquely-named data elements can exist[1].

Many will readily acknowledge that these *are*, in fact, myths and that they don't really hold these assumptions. However, it seems to me that the users of namespaces and the developers of SDVs and metadata registries are pursuing their work as if these assumptions were true. No mechanisms or strategies have appeared in the extant literature that acknowledge, explain, or address the challenges that arise due to these faulty assumptions.

The reasons that these assumptions are faulty – making them myths – fall into three facets of SDV development and use:

- Scope

[1] Uniquely-*named* as opposed to uniquely-*identified* data elements – see note on Namespaces and URIs below.

- Natural language use
- Schema evolution

(There are undoubtedly other facets that contribute, but only these three will be addressed here.)

Scope

One of the principal reasons that these assumptions are myths is due to the issue of scope of control. The assumptions can be *made* true (i.e., be made "un-myths") with the proper oversight e.g., the assignment of a data steward with the authority and scope of control to police the uniqueness of the names and the semantics of their use. However, the scope of control of such a steward is necessarily limited. There will never be a data steward with a global reach. In fact, it's unlikely that a data steward with these responsibilities can be effective (or even exist) in any but small data-intensive enterprises.

Standards development organizations like OASIS (www.oasis-open.org) and the use of registries (e.g., www.xml.org) will not avert this problem. Not only will there be competing organizations developing and registering their own SDVs (e.g., OAGIS Canonical Model (www.openapplication.org), the ebXML Core Components (www.ebxml.org/specs), the IEEE SUO (suo.ieee.org), to say nothing of all the unrelated *xxx*MLs), but many organizations won't know about, or even think to search out, registries or SDVs – they will naturally assume an insular view of the problem they are trying to solve and develop their own vocabularies for their own use. In addition, most newcomers to the field of standards have no idea how difficult it is to achieve consensus on meaning (let alone the set of terms) of the elements in a vocabulary. (An observation originating from almost 20 years in the field of data standards development.)

The conclusion from this is that there will be no authoritative agent with the global scope of control to enforce constraints on the naming and use of data elements and vocabularies. Vocabulary and registry development efforts must recognize and accept that independent vocabulary "namespaces" (i.e., scopes or domains) will arise in the world – which is why Myth #3 is a myth: uniquely-named data elements will not exist.

> *Note: The use of URIs* **does** *ensure the uniqueness of designations across the World Wide Web, but has no effect on the purpose of unique names: to convey a conventional (widely accepted) meaning.*

"Rules of engagement" are necessary for meeting and dealing with the semantics and vocabularies of the different namespaces. The first rule of engagement should be a semantic "Prime Directive": *Each "Namespace" has a perfectly valid and legitimate right to, and ownership of, their own semantics and vocabulary*. In other words, no standards organization has the right or ability to force or dictate how another organization chooses to use or name their data.

Natural Language Use

In natural languages, the meaning of words and sentences are conventions coupled with innate linguistic capabilities that have arisen over course of human evolution. Among the linguistic capabilities is the use of extra-linguistic perceptions to interpret the correct meaning of uttered sounds (i.e., words, sentences); these extra-linguistic perceptions are typically referred to as the *context* of an utterance. It is because of contextual differences that the same word is able to convey difference meanings.

Although the innate linguistic capabilities are (presumably) the same across the human species, the *conventions* vary regionally and by usage community. It is by *convention* that words acquire meaning: the constant reinforcement of the meaning of a sound/word through repeated use in a language community is what gives a word its meaning. (And it is why there are different languages, and why languages evolve over time.) It is a mistake to assert that a dictionary defines a word in the sense of specifying its meaning – it doesn't! A dictionary documents the *conventional meaning* of a word.

There have been no arguments made and no evidence presented in the literature that the (human) assignment or (human) interpretation of the "meaning" of data elements/terms in an XML document or schema will be any different than the use of individual words in natural language. Quite the contrary: the fact that technical discussion of issues often devolves into philosophical arguments provides strong evidence that they are the same. Thus, the differences in context and in the conventions of different usage communities are the reasons that Myth #2 is a myth: humans will not assign/interpret data element semantics consistently. (This is also the reason that the EDI standard requires volumes of Federal Regulations to use/interpret them consistently.) This phenomenon is amplified by the fact that language communities will arise independently and without knowledge of one another.

Schema evolution

Schemas are typically used as not only a data validation mechanism, but also as a mechanism – when coupled with natural language prose definitions - for specifying the semantics of the data. As the size, scope, and number of purposes (i.e., the objectives, missions, requirements) of a community increases, the schema must evolve in response: the size, scope, and number of purposes served by the schema will also increase. There are a number of competing and conflicting forces in schema evolution that are not easily reconciled, managed, or even recognized.

The first is that the evolutionary forces that prompt changes in business and business computing environments will induce schemas to change frequently to adapt to the new environment or face becoming obsolete. Schemas are typically designed as if they are a static declaration that can be objectively completed, as if there were some criteria by which one could say that the schema is "done". This is a mistake because the performance of a schema *always* degrades over time because of changes in the usage environment. (This observation applies, of course, to "dis-embodied" schemas that are not bound to a particular application.)

The second is that as the scope of the schema grows (i.e., the number of things it "has to do" or "be able to say"), the schema will have to accommodate a very very large number of primitive concepts in order to remain semantically precise (i.e., unambiguous). To do this schema will evolve in one of two directions:

- It will grow in size yielding a very very large number of element types and "interpretation flags"; or
- It will become very "abstract", "neutral", or "generalized" to embrace the wide variety of semantics.

The first option is impractical for the very reasons that enterprise schemas (or global data dictionaries) are impractical. Not only is a huge monolithic schema difficult to understand and manage, but it is also far too cumbersome to adaptively respond to changes in the usage environment. This is one of the primary reasons that Myth #1 is a myth: it is not possible to uniquely identify, name, and define all (or even enough) data elements necessary for effective information exchange for a large, many-purposed community.

The second option is problematic in a fuzzier way: an abstract, neutral schema is by necessity more ambiguous, thus defeating the need for semantic precision. Abstract terms with general definitions readily and validly admit varied and often incompatible uses. This is one of the primary reasons that Myth #2 is a myth.

Recommendations

The problems described above are not insurmountable, but certainly challenge "standard data vocabulary for application interoperability" proponents to answer some hard questions or rethink their direction. The following are some thoughts and recommendations for the SDVs and registry development projects to consider as they define the problem they are solving and design solutions for it.

Islands, bridges, and role of the Data Steward

The first principle that the SDV/development projects should accept and adopt is that there will be multiple independent and autonomous "namespaces" that service different communities, and that it is not possible to bring them all under the same managerial umbrella. Each community has the perfect right to define their own semantic requirements and solve them in whatever manner they see most fit. Therefore, an important component of the project work should be devoted to the analysis and design of how "bridges" between information community "islands" can be designed and built. The bridge may be a simple equivalency mapping between data elements, or a complex trading partner agreement. The "rules of engagement" mentioned above include how bridges are constructed.

The role of the data steward, then, is not as a policing agent for data element names, definition, and use within a community, but rather as an ambassador and "bridge warden" empowered to represent and explain the semantic requirements of the community he represents. He is responsible for the construction of "his half" of he bridge when negotiation and forging relationships with other "islands".

Stability versus evolution of schemas

Schemas must adapt and change with changing business needs. Therefore, the applications that use data elements within a community should be metadata-driven (i.e., actively use the schema) as far as possible. This will allow the application to adapt more quickly and easily to changes in the environment because a new, more suitable schema can be "plugged in" to the application. (The componentization and orchestration of web services will also enhance the adaptability of the application.)

This adaptability is even more important when the construction of bridges is considered. Negotiating a bridge with another community often will necessitate re-thinking and altering the local community's perspective on their schema, necessitating changes or tweaks. But the changes and tweaks are a *positive* behavior rather than negative: it is a self-corrective evolutionary action. It enables the local community to better align their applications for effective information interchange with the newly connected community, thus forging a large, better integrated community.

Although almost all schemas will need to change over time, the stability of a schema is important for many applications (and some usage communities). There are conditions under which a schema is stable over time; empirically, a schema is most stable when it:

- is small in size;
- has a small scope;
- services a small usage community (ideally just one person); and
- serves a single (or small number) of fine-grained purposes.

Schemas that reflect these properties are also the least ambiguous schemas. Ironically, abstract schemas are also fairly stable over time because they have more “semantic play” – they are semantically more “forgiving”, “looser”, or “malleable” with respect to semantics (which of course means that they’re ambiguous.)

Ideally, schemas that reflect the stability properties above can be “bridged” to a larger, more widely-scoped community schema, thus enabling more effective (i.e., less ambiguous) exchange of information because the links to the less ambiguous schemas are maintained. And although the local schemas must still be adaptable in the face of change, the use of bridges also isolates islands from the ripple effect of impacts of changes to other schemas and other bridges.

Standard Data Element Semantics

Despite everything said above, it is still possible to envision “standard data elements” that “everyone” uses the same way regardless of context, application, or schema. These standard data elements are representations of concepts that are ubiquitous throughout human endeavors. The notions of a “person” and a “person’s name”, a “point in time”, and a “location on the planet” are universal concepts (or at least universal within the sphere of human experience) and the adoption of standard data elements with conventional and widely-understood semantics is not only feasible, but likely. The ebXML Core Components are targeting the definition of ubiquitous concepts like these.

It is also possible to define “standard data elements” for small, finely-purposed domains. The Dublin Core Metadata Initiative (www.dublincore.org) has defined a set of 15 standard data elements for tagging web resources for the purpose of cataloging and search (much like a global, searchable “card catalog”). These elements have been widely adopted and used by libraries to make their collection catalog information available on the World Wide Web.

It should be recognized, though, that these concepts and finely-purposed domains are relatively few in number when compared to the infinite variety of information that humans might want to exchange between applications.

Conclusion

The purpose of this paper has not been to argue that the problems and the challenges that face the SDV/registry development projects are unsolvable. Rather, it is to suggest that the solution vision must be more expansive. Faulty assumptions must be rooted out and the problems that are thereby exposed must be explicitly and directly addressed. Despite their intuitive appeal, namespaces, SDVs, registries, and unique data element names will not solve the problem of interoperability. What is needed is the recognition that the semantics of a schema (or, more precisely, the semantics of data governed by a schema) must be explicitly bound to a known community that it serves, and that "bridges" between the communities will be an inevitable part of any comprehensive solution. By leaving the semantics locally bound to a community, the whole issue of "standard semantics" becomes moot.

References

[1] GAO, "Challenges to the Effective Adoption of the Extensible Markup Language Report to the Chairman, Committee on Governmental Affairs, U.S. Senate," General Accounting Office, Washington, D.C. GAO-02-327, April 2002.

[2] Knox, R., "GAO Report on XML Adoption Challenges Breed Confusion," Gartner, Inc., Research Note/Commentary COM-17-0045, 11 July 2002.

[3] Perry, W., "Standard Data Vocabularies Unquestionably Harmful," 2002. http://www.xml.com/lppt/a/2002/05/29/perry.html

Bio

Mr. Burkett is a senior information systems engineer specializing in the design and development of data-level application interoperability solutions with an emphasis on model-based data exchange, data transformation, data integration and data mapping technologies. He has extensive modeling experience using a variety of schema languages, and has developed methods for mapping between schema languages. Mr. Burkett has led teams that have designed interoperability software solution components, such as an XML Registry/Repository, using established requirements engineering and design engineering principles and practices. He has been an active participant and leader in the international data exchange standards development community (ISO TC184/SC4); and he has applied many World Wide Web technology standards, such as XML, XML Schema, XSLT, ebXML/OASIS, PDML, and STEP/EXPRESS. Mr. Burkett's papers on XML semantics and PDML have been presented/published at XML conferences and in refereed technical journals.

Contact Information

William C. Burkett
Product Data Integration Technologies, Inc.
444 W. Ocean Blvd Suite 1800
Long Beach, CA, 90802 USA
562-495-6500 x13
Fax:562-495-6509
wburkett@pdit.com
http://www.pdit.com