

Canonical Model Management- A natural evolution from XML Schema Management

The majority of organizations that are moving away from point-to-point integration quickly realize that it is essential to adopt a single Canonical Model to ensure data interoperability for data in motion. As the standard for data in motion is XML it naturally follows that the first step is to create a Canonical XSD. Many organizations are keen to accelerate this effort by adoption industry Canonical Schema like OAGi, ACORD, NIEM, FpML, ARTS, or reference models like Oracle AIA and IBM FSDM or IAA. These standards and frameworks use XML schema to create information models for data in motion.



Canonical Schemas- Information Models in XSD

- Industry Bodies
 - Insurance - ACORD
 - Banking - FPML
 - Retail - ARTS
 - Telco - SID
 - DOD - DODAF
 - Horizontal - OAGIS
 - Government - NIEM
- Oracle & IBM
 - Oracle AIA
 - IBM IIA (Insurance) FSDM (Banking)















The majority of adopters of this approach today use XML Editors and a source code repository to manage their Canonical Schema - this is the world of XML Schema Management.

Schema Management has its limitations when applied to the requirements of rolling out a Canonical Model strategy, especially for large IT shops as they try to scale across multiple lines of business and meet project needs:



Canonical Model Management NOT Schema Mgmt

	igniteXML	XML Editors
<ul style="list-style-type: none"> • Collaboration <ul style="list-style-type: none"> • Collaborative approach to building and maintaining the model 		
<ul style="list-style-type: none"> • Scalability <ul style="list-style-type: none"> • Easy adoption of model by non-experts • Automation of manual tasks e.g. trimming • Overlay metadata onto the model e.g. Mappings • Link physical models to logical models 	   	   
<ul style="list-style-type: none"> • Governance <ul style="list-style-type: none"> • Fully controlled environment for model consumers. 		

Canonical Models are dynamic and constantly changing. Typically the more they change and the larger they get, the more people are involved in their lifecycle management. Schema Management with manual editors do not support a collaborative environment for this work. Furthermore there are many scalability issues with Schema Management. For example, if an organization has adopted a standard like ACORD and has built extensions to customize the standard for their enterprise, the process of coping with updates to the ACORD standard and understanding the impact of changes quickly becomes very complicated and difficult. Likewise, these schemas are very large and complex. The runtime environment requires small subset schema for performance optimization. Trimming these large standards with manual schema editors is very time consuming, error prone, and often leads to developers inadvertently breaking the lineage between the model and the deployed message.

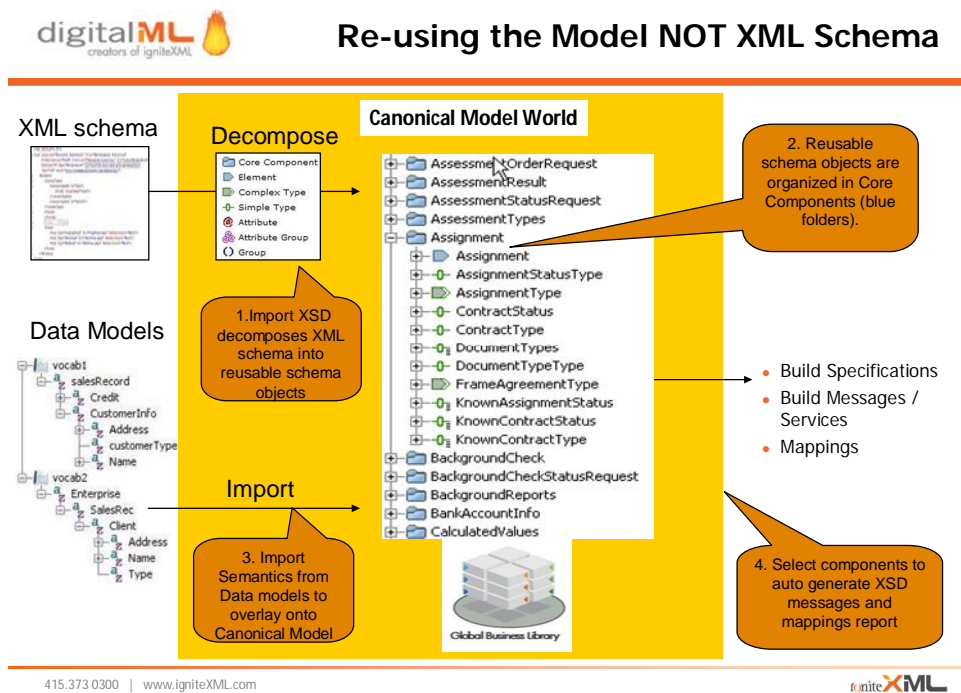
The objective of working to a Canonical Model is to achieve consistency of data in motion across the Enterprise Integration stack. If developers are able to do anything with an XML editor to your Canonical Model there is clearly no governance and therefore a high likelihood of the whole objective being missed.

For a Canonical Model strategy to be successful there is the assumption that all project teams will use and adopt the model. In reality there is often resistance to learning the model. Developers are familiar with their application vocabulary, feel it is quicker to develop point-to-point interfaces, and do not have the motivation to learn and find what they need from the Canonical Model while

business analysts, unfamiliar with complex XML schema, are forced to work through complicated mappings to understand data lineage and deliver proper development specifications. To remove this resistance, many IT shops are keen to attach metadata to the components of the model. One type of metadata is to link terms used in existing data models to the part of the Canonical that it relates to allowing developers to search their familiar vocabulary while being pointed to the approved Canonical definition. This simply is beyond the capabilities of Schema Management tools.

The eventual output from a Canonical Model is a message schema or service. These need to be mapped to the backend system. Therefore, there is a clear case for mapping backend data fields to the model once and delivering messages that specify how they map to the backend system. Again this type of functionality is impossible in the Schema Management world.

Finally many organizations are using logical modelling tools to model processes and abstract business concepts. There is a very strong need to link these logical models to the physical models held in industry schema to build a complete Canonical Model that can be understood by both business analysts and developers. This linking is also impossible in the Schema Management world.



In the Canonical Model Management world schema is decomposed into an object model where every object is individually managed in a collaborative repository. Updates, changes, versions, and impact analysis is done at an object level with a role based approval process. Model managers are able to attached metadata to individual objects to ensure data lineage, adoption, and

assist with mappings. The model is made available to developers and business analysts as reusable schema objects allowing them to easily navigate the model, find what they are looking for, build specifications, and select components to auto-generate XSD messages and mapping reports. This approach enables Canonical Model adoption, provides faster development of messages that are run-time optimized and always derived from the model, and ensures data lineage from model to schema to service.. Taking a Model Management approach ensures all the above requirements can be easily catered for to allow the success of your Canonical Model Management strategy

digitalML, the makers of igniteXML, is the enterprise Canonical Model Management company. igniteXML is used by Integration and Service Oriented Architecture (SOA) teams in Fortune 2000 and large Government organizations. Through patented technology, igniteXML has changed the ways that customers manage and ensure adoption of their Canonical Model.

North America Corporate Headquarters:

digitalML-USA Inc.
580 California Street, Suite 1600
San Francisco, CA 94104
Tel: 1.415.373.0302 Fax: 1.415.373.0307
Email: jbogard@digitalml.com and support@digitalml.com

UK Corporate Headquarters:

digitalML Ltd.
Atrium Court
The Ring
Bracknell, Berkshire, RG12 IBW
Tel: 44 1344 390542
Email: jsindall@digitalml.com and support@digitalml.com