K-12 Education Metadata and the NIEM

Case Study: Extending the National Information Exchange Model to Include K-12 Education

Dan McCreary
President
Dan McCreary & Associates
Minneapolis, MN
http://www.danmccreary.com

ABSTRACT

This case study examines the extension of the National Information Exchange Model NIEM [1] to include K-12 education metadata. NIEM's compliance with ISO/IEC 11179 [2] metadata standards was found to be critical for cost-effective system interoperability. This study indicates that extending the NIEM can be compatible with newer RDF and OWL metadata standards. We discuss how this strategy will dramatically lower data integration costs and make longitudinal data analysis more cost-effective. We make recommendations for state education agencies, federal policy makers, and metadata standards organizations. The conclusion discusses the possible impacts of recent innovations in collaborative metadata standards efforts.

Keywords: metadata, K-12 Education, NIEM, NCLB, data dictionary, data warehouse, ISO/IEC 11179, OWL, ontology

DISCLAMER

The opinions expressed in this paper are solely that of the author. These opinions do not necessarily reflect the opinions of the Minnesota Department of Education, the US Department of Education, the Wisconsin Department of Public Instruction, or the US Department of Homeland Security.

INTRODUCTION

The ability for states to maintain statistics on student achievement has motivated policy makers to propose legislation to use these statistics to hold states accountable for complying with civil rights objectives. In November 2005 the US Department of Education awarded statewide grants of $52.8 million (USD) to be used over three years to create longitudinal data standards for statewide student testing [3]. A significant portion of these funds is being used to set up distinct statewide data dictionaries for data warehouse projects. The states of Minnesota, Wisconsin, and Michigan formed a collaborative agreement to develop standardized metadata based on federal standards. This paper describes the business drivers of this project, the approach taken, the constraints of the communicating data to researchers across states, across time and across testing standards.

LEAVE BLANK THE LAST 2.5 cm (1") OF THE LEFT COLUMN ON THE FIRST PAGE FOR THE COPYRIGHT NOTICE.

Business Requirements

The business requirements for this project were to build a shared metadata registry to be used by K-12 assessment data warehouse projects across three states (Minnesota, Wisconsin and Michigan) as well as researchers at the Wisconsin Center for Education Research. Unlike many federal K-12 standards published in unstructured documents, our goal was to build a machine-readable data dictionary backed by a structured data-element approval process. Although the primary business driver for this system was a multi-state data warehouse for student assessment results, other information technology projects would also benefit from machine-readable metadata definitions. These projects include the creation of XML-based exchange documents, web-services, and a migration to service-oriented architectures and enterprise service bus architectures.

The State of Minnesota government enterprise architecture data standards require the use of XML and ISO/IEC 11179 metadata registry structures due to the widespread adoption of these standards by federal agencies and the desire to minimize the number of metadata standards in use within the State of Minnesota.

Existing Federal Metadata Standards for K-12

The US Department of Education has created several standards for the coding and transmission of K-12 educational data. These standards include the National Center for Educational Statistics (NCES) [4], the Education Data Network (EDEN) [5] and the Common Core Data (CCD) [6] standards. In addition, a collaboration of K-12 school districts and computer system vendors have created a nationwide, single-namespace XML-based inter-operability standard called the School Interoperability Framework (SIF)[7]. Initially designed to automate data flows for new student registration within school district computer systems, SIF has since been extended to include other data elements. Unfortunately SIF does not comply with other federal metadata guidelines such as ISO/IEC 11179 structures.

Federal XML Usage Guidelines

Although there are several sources of federal K-12 education metadata, there are no machine-readable federal-level K-12 metadata standards that are consistent with XML and ISO/IEC metadata standards or other federal metadata registry guidelines [8]. In 2002 the Federal XML working group published these standards however, adoption of XML standards is lacking in many K-12 agencies due to a lack of funding. Federal metadata standards include 1) ISO/IEC 11179 metadata registry standards 2) XML data element naming standards 3) best practices. These best practices include the extensive use of XML Schemas, the use of multiple namespaces, three-part data element names for properties, the use of upper camel case data element names, and the use of formal representation terms for all data element properties.

The US Department of Justice (DOJ) and the Global Justice XML Data Model (GJXDM) [9] have championed the effective use of these federal standards and best practices. The mission of GJXDM was to connect over 10,000 law enforcement, courts, prosecutors, jail, and prison and probation systems. Accomplishing this required a comprehensive data model with efficient tools for finding and extracting sub-sets of the entire system. Tools for managing these subschema data elements were developed under a DOJ contract by the Georgia Technical Research Institute (GTRI)[10]. Subsequent DOJ grants have funded extensive national training for government agencies and software vendors. This training has resulted in widespread adoption by federal, state, and local agencies as well as vendors of criminal justice systems. Tools that integrate these standards are also becoming available. Many of the tools developed by GTRI are now being adopted in a federal-level standard called the National Information Exchange Model.

Adoption of advanced GDXDM tools by the US Department of Homeland Security

One result of the widely-adopted GJXDM structures and exchanged document creation tools was the acknowledgment by the new Department of Homeland Security (DHS) that many semantic standards for exchanging data between agencies had already been developed. It would be an inefficient use of taxpayer dollars to duplicate these efforts. A more cost effective approach is adoption, extension, and generalization of these standards so they are not specific to criminal justice. Creating a new standard would only cause confusion about which standards to follow. Led by semantic web advocate Michael Daconta [11], efforts were made to centralize metadata definitions for multiple federal agencies and discussion began about including domain-specific concepts from other areas such as the US Department of Health and the US Department of Education. These new efforts were given the name National Information Exchange Model to separate it from a justice-only data model.

STRUCTURE OF THE NIEM METADATA

The NIEM follows many of the guidelines set forth in ISO/IEC metadata registry specifications that date back to 1995. General in nature, the ISO/IEC 11179 specification includes a wide range of metadata issues. In practice it has been a guidepost for agencies that are attempting to create metadata registries. Unfortunately, the standard is very abstract and many of the structures optional. Only UML structures are suggested by these standards. Unlike OMG's XMI and CWM specifications, there are no XML binding to the ISO/IEC 11179 specifications. This required implementers to clearly document their interpretations of the standard. Federal agencies frequently document their interpretations to the specification but there remain wide variances.

Subclassing NEIM Upper-Ontology Conceptual Data Elements

At a high-level, the NIEM is structured as a hierarchy of conceptual data elements called "Types". The word "Types" is used because complex data "types" are created in XML Schemas. These correspond to what ISO/IEC 11179 specifications define as Data Element Concepts. These type structures fit into an inheritance hierarchy with the root data element called "Super". This corresponds to the OWL [12] root concept called "Thing".

The first level of the NIEM contains four central abstract concepts that tend to be used in most modeling systems:

Activity: A generic moment-interval container for data associated with an event that occurs at a specific point in time or time interval.

Document: Any data or information about any collection of data or information, regardless of format, which has definable boundaries and is so designated for one or more purposes.

Organization: Any unit consisting of people and processes established to perform some functions.

Person: An instance of a human being.

Many NIEM concepts are subclasses or related to these structures. These concepts form the bases for the NIEM. We found that education data fits into this structure logically. **Schools** and **SchoolDistrict** are subclasses of **Organization**. **Students** and **Teacher** are subclasses of **Person**. **Enrollment** and **TeacherLicense** are subclasses of **Activity**.

Each NIEM "Type" can have one or more "properties" associated with it. Each property has a representation term associated with it. Just like in object-oriented programming, properties of a superclass are automatically inherited by all the subclasses of that superclass. For example there is only one data element to store a person's birth date. This property is called **PersonBirthDate**. Both **Student** and **Teacher** are subclasses of **Person** so the need to store student and teacher birth dates in not necessary. This subclass hierarchy is a standard way to reuse metadata

structures and is critical to keeping the size of a metadata registry manageable.

NIEM Classification Scheme

One of the challenges with the NEIM is that it does not (as of this writing) come with any of the advanced sub-schema generation tools included with the GJXDM. To alleviate this, an early version of the NIEM (version 0.1) came with data elements classified into three areas:

Universal Data Elements – Core date types such as Activity, Document, Organization and Person.

Core Data Elements – Data elements used less frequently such as Airplane.

Domain Specific Data Elements – Data elements that are unique to a single federal agency such as the department of Education.

We imported and sub-classed the data elements in the Universal set and created education-specific extensions that could be placed in a domain specific area.

CHALLENGES

Although the NIEM was not designed to be used with K-12 data, we did not see this as a major obstacle. Most of our challenges had to do with other aspects of the NIEM.

The Need for Semantically Precise Data Element Definitions

Our fist challenge using the NIEM was the problem with finding semantically precise data element definitions. Writing semantically precise data element definitions is a complex process. Definitions need to be abstract enough to be reusable but precise enough to give guidance. ISO/IEC 11179 standards give careful guidance in this area [14]. ISO/IEC guidelines also require each data element to have a non-circular definition. Many NIEM data element definitions are circular giving users little guidance in their semantic intent.

The problem of precise data element definitions is not unique to the NIEM and the GJXDM. Many metadata registries are set up and managed by stakeholders that have little formal training on the fundamentals of writing unambiguous definitions to promote interoperability between systems. This is an area that needs emphasis by metadata project managers, business analysis, data architects and programmers.

The problem of semantically precise data definitions is most apparent when deciding what high-level data elements to subclass. The struggle to decide if something is a subclass of Activity or Document frequently requires analysis of the dynamic nature of a structure.

We created a data stewardship training process that included training on how to write precise definitions. Briefly, here are some of the guidelines we included in our instruction for creating precise definitions. Clear definitions are:

Precise – The definition should use words that have precise meaning. Try to avoid words that have multiple meanings or multiple word senses.

Concise – The definition should use the shortest description possible that is still clear.

Non-Circular – The definition should attempt to avoid the term you are trying to define in the definition itself.

Distinct – The definition should differentiate a data element from other data elements.

Adding Usage and Notes Statements

We solved our problem of over specification of definitions by adding two separate text fields to our metadata registry. We tried to keep our definitions clean to promote reuse. When we have system-specific information, this was migrated to a "Usage" text block. When there were warnings about topics such as security policy associated with a data element, this information was moved into a Notes section. This allowed us to document how metadata was used in practice without corrupting the definition with information that external metadata users may not be concerned with. The general rule is if an external user is sending or receiving data and does care about the idiosyncrasies of our internal systems, the information should not be part of the definition.

DATA COLLECTION, VALIDATION AND MAPPING

Many of the stakeholders of K-12 education systems require a great deal of importing of data from external sources. These external data sources are typically school districts, charter schools, and testing vendors. Examples of this data includes: a list of enrolled students, list of licensed teachers, list of student test results and school financial information. Before this data is loaded into a data warehouse, it also must be tested for quality against a large set of business rules called "edit checks".

Legacy Data Formats for K-12 Education

In the 1970s school districts would bring a "card deck" of 80-column punch cards with school information to a regional processing center. Each card deck would be batch loaded into a mainframe and processed overnight. The next day a printout would report inconsistencies in the data and the process would be repeated till the data set was correct. This process frequently took weeks since some districts were required to drive hundreds of miles to a regional service center to have their decks processed.

Today the punch cards are gone. It is interesting to note that the 80-column fixed-width file format persists. Custom software has been written by hundreds of districts and vendors to create data in this 80 column format. Changing to a new format would require a statewide overhaul, not just at a central location but also at thousands of school districts and charter schools. One consequence of this is that many staff members are concerned about moving from 3-digit codes to 4-digit codes since it would change the layout of the data files.

Almost everyone acknowledges that XML would be a more flexible file format, but there are no statewide budgets to modernize each data submission point.

Approach 1: The Batch Upload Method

Based on these constraints, state departments are reluctant to migrate to new XML data structures. The batch upload process is still used; districts are given access to a secure web site that they can upload their data sets. Business rules can then be quickly run and errors displayed. Data submitters note the errors, modify the source data and then re-upload the entire data set.

Approach 2: Custom Validation Software

For large data sets, the upload/validate/change/re-upload process is inefficient. Attempts were made to write software that would allow organizations to remotely validate the data at their site before it was submitted. This was done by writing a custom software application that validated the data. Although this worked in many cases, it was also problematic in three ways. First, the software only ran on Microsoft Windows™. Since many K-12 organizations use the Apple MacOS™ they could not use the software without Windows emulation. Second, the installation and setup of the software required extensive technical support. Finally, business rules continually changed requiring frequent re-distribution of software. Users neglected to update their software and would validate on prior-year business rules.

Approach 3: Distribution of XML Schema and Transforms to Clients

One proposed approach was to adopt was the distribution of XML Schemas and XML transforms to validate complex data sets. Since XML valuators and XML transforms can run on multiple operating systems and be quickly updated from a central location, this was an appealing solution. The challenge is a training issue. XML Schema and transform classes were taught to staff and there are now ongoing discussions about migrating to use these techniques. The biggest challenge with this approach is that most staff members have classical "procedural" training. They know how to write Java or Visual Basic but are not adept at moving to declarative systems such as XML Schema and XML transforms. It does not help that at the federal level XML file formats are still the exception rather that the rule.

Approach 4: Record Level Transfers Using Web Services

The ideal solution involves moving away from batch-oriented process to on-demand, fine grained, computer-to-computer transfers of information with humans only intervening when exceptions occur. This is currently the approach being pursued.

On-Line Learning Use Case

An example of this type of transaction would be when a student enrolls in an online-course in a remote school district using a low-cost online Learning Management System (LMS) such as Moodle [14]. Minnesota statutes dictate that if a student takes a class out of their home district, funding for that student also moves to the district that provides the online-course. It is interesting to note that the state of Wisconsin does not allow per-class funds to be moved out of the district.

Upon enrollment this information could quickly be transmitted to a central server using a semantically precise web service. Financial records could be also updated automatically and schools could dynamically adjust their budgets and IT resources based on daily on-line enrollment.

Automating Semantic Mapping and Transformation

The above scenario would be simplified if every school district, school, and classroom used the same learning management system and all the computers were on a single secure network. Unfortunately, each school can choose to host their LMS at out-state Intranet Service Providers (ISPs). Class enrollment information will always be coming to state agencies in multiple formats from hundreds of remote computer systems. State K-12 agencies, school districts and education vendors need powerful tools to allow non-programmers the ability to quickly map these transactions in semantically precise ways.

One class of tool that has become cost-effective ($300 to $900 USD per named user) in recent years is visual data element mapping tools such as Altova's MapForce™. These tools allow non-programmers to visually map data from one system to another and generate a run-time program to execute this transformation in XSLT, Java or other procedural language. A screen capture of this process is shown in Figure 1.

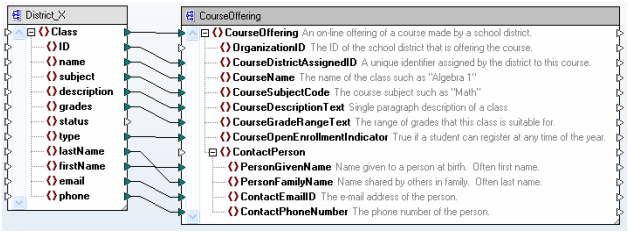

Figure 1: Mapping Course Data Using MapForce™

On the left side of Figure 1 a mapping to a simple school on-line course catalog "flat file" is shown. On the right is a sub-schema that could be generated directly from an XML transform of a metadata registry. There are two important differences between the left-hand schema and the right-right hand schema: the use of a three-part XML data element name and the use of short but precise data element definitions. The use of precise data element definitions is discussed above.

Use of Representation Terms

A helpful technique for data mapping projects has been the addition of representation terms to XML data element names. Representation terms were initially used to classify data types or how the data should be represented within a database using data types such as date, integer or string representations. Table 1 lists some of the representation terms adopted based on an analysis of the ebXML, GJXDM and NIEM metadata registries:

Amount – A monetary value with units of currency.

Boolean – A true/false value.

Code - An enumerated list of all allowable values. Each enumerated value is a string that for brevity represents a specific meaning.

Date – An ISO-8601 date in the form yyyy-mm-dd.

ID – A unique identification of a record within an identification schema.

Measure – A numeric value determined by measurement with units.

Name - A textual label used as identification of an object.

Text - Character string generally in the form of words.

Table 1: Sample Representation Terms

As these standards have evolved, developers have found it useful to use the representation term suffix of an XML data element to describe more of the semantic classification of the data, not simply its data type. For example, some representation terms are used to identify records in a data set. Although we may not be concerned if the form of the identification is a string or an integer, we do care that this information can be used to differentiate records in a record set.

This process has been very useful to data architects that are building data warehouse cubes. Data architects are interested in what data elements identify data, what data elements classify data within a data set (known as a dimensional category) and what data elements can be used as measures (items that can have sums and averages calculated). Data elements that have a suffix of **ID** are excellent candidates for avoiding duplication in a dimension. Data elements that end in **Indicator** or **Code** are used for dimensional categorization and data elements that end with **Amount** or **Count** can be used as measures. For example any data element that begins with Person or Student and ends with ID can be used to ensure that the Student dimension does not contain duplicate student information.

The addition of a representation terms as a suffix to the XML data element name makes the XML data element names slightly longer. Users find that the semantic clues and consistency that it creates, more that compensate for the additional length of the data elements.

There was also confusion about other representation terms that should be used. For example both **Percent** and **Rate** implied that dividing two numbers arrived at a number. Both **Number** and **Count** were used to count entities. Both **Value** and **Quantity** were used to store floating point numbers. We hope that standards organizations work to clarify what representation terms should be used in these areas.

METADATA REGSITRY MANAGEMENT SOFTWARE

An initial effort was made to select a software system for managing the metadata registry. An extensive list of features was created and circulated to several metadata registry vendors. None of these vendors would commit to fulfilling the terms of the RFP for under $250,000. Although a search was done for open-source metadata registry software, none was found that met our requirements. Integration with internal directory structures for data element approval was an especially difficult integration point.

A quick short-term custom solution was created to handle initial data element structures. Because the initial requirements for this system were straightforward we hoped to start simple and grow.

1. A small XML-driven Glossary-of-Terms was created to centralized terminology.
2. A second XML Schema was created for Data Elements based on the meta-model implied in the ISO/IEC 11179 metadata registry specification with RDF/OWL structural data elements strongly considered.
3. An Apache Ant script was used to concatenate all the data elements together to form a complete Data Dictionary.
4. An XSLT transform was created to transform Data Elements into HTML.
5. As business requirements expanded, additional fields were added to the data element XML schema.
6. The Apache Ant build file was expanded to include data element validation and installation scripts for both the public web site and the private intranet site.
7. A library of reusable XSLT templates was extended to allow for rapid customization of reports.
8. The original XML Schema was redesigned to promote isolation of state-specific items such as systems dependencies, stakeholder lists, approval status, individuals, and teams.
9. User interface forms were generated directly from the data element XML schema.
10. Transforms were written to transform the XML schema to create other user-interface data elements such as pick-lists.
11. XML data elements were eventually loaded into a relational database to promote rapid searching.
12. Individual data elements were stored in XML files and could be fully version-controlled using the open-source Subversion system.

As a result of this process a XML Schema-based Model Driven Architecture (MDA) was achieved. Any changes to the XML schema that defined data element structure was immediately transformed into other development artifacts.

This promoted flexibility and allowed metadata structures to be quickly added without any rebuilding of databases.

This strategy was dependant on having a staff that had proficient XSLT and XPath skills. Without these skills the temptation to copy-structures over transform would become too difficult to overcome.

Publishing in XML, OWL and XMI Formats

The internal structures we chose for our metadata were strongly influenced by the ISO/IEC 11179 specification and RDF structures. We could validate our metadata registry by transforming the registry into XML Schema, Web Ontology Language (OWL) and XML Metadata Interchange (XMI.) formats.

The data-dictionary-to-OWL transforms used only basic OWL-Lite structures such as **`class`**, **`subclass`**, **`property`**, **`label`**, **`comment`** and **`enumeration`**. These OWL files were imported into Stanford Medical Informatics Protégé and Altova's SemanticWorks™ applications to check for consistency. Although we found this very helpful in our debugging process both systems could be improved by more precise error messages on data import.

We also stored semantic mapping to other metadata registries and created **`equivalentClass`** and **`equivalentProperty`** statements with the hope that one day semantic brokers could utilized this information. This information has yet to be tested.

We also found that our semantic mapping to other standards were frequently not an exact match. Therefore we added a match precision level (low, medium, high) property to aid future tools. Semantic mapping to foreign metadata would be expedited if each metadata registry would provide permanent versioned URLs for each data element. An excellent example of this is the permanent URL structures used by the Dublin Core standards.

XMI metadata files are imported into many UML modeling tools and allow software developers to begin by sub-classing a library of simple Java classes that have been automatically generated directly by the metadata registry or through the UML modeling tools. These structures are commonly known as Plain-Old-Java-Objects or POJOs. Customized transformations can allow a software developer to create data element selection criteria such as "All data elements that are referenced by system X and Y but not Z". These are expressed in a complex XPath expression and copied into a data element extraction template. This generates a list of POJOs (similar to a want-list) and the developer can then subclass these structures and add their own application-specific business logic.

STEWARDSHIP, LEARNING, TRUST AND VISUALIZATION

We believe that one of our most important insights in this project is the tight coupling between understanding structures and acceptance of the data stewardship role. It is well understood in the ontology community that it is usually not feasible for a data architect to have ongoing subject matter expertise in all domains. Maintenance of data element definitions must be carefully delegated to the appropriated stakeholder teams and the role of data stewardship must be accepted by each domain team.

What we found is that there are two critical aspects to this process. First, each domain must have a supportive manager that allocates sufficient time to their staff for this work. Second, the domain team must have tools to quickly understand their structures before they could contribute. We found the best way to do this was with the creation of colorful hierarchical visualization tools.

We searched for low-cost visualization tools that could import an XML node list and did not require the installation of expensive desktop client software. Although we felt that the Scaleable Vector Format (SVG) structures were ideal, SVG does not supply automatic placement of objects in a graph. In the end we found that the open source FreeMind software was ideal for displaying and browsing colorful object-hierarchies. We also created high-quality output in MindJet's MindManager™ for those users that installed the free viewer or for those users that had appropriate licenses. Both these systems did an excellent job of importing XML formats. Creating XML transforms into each of these formats could be done in a few hours.

We found that having XML transformation templates to tools to quickly "prune" the data element hierarchy to display only data elements under discussion and color-code them based on status accelerated the process of stakeholder trust building and adoption of the role of data stewardship.

Structures for Tracking Data Sources

One of the first complaints to our early system was the lack of precision mapping to the actual sources of data for a data warehouse. For example the team building our data warehouse was tasked with doing many complex extract, transform and loads (ETL). These ETLs were performed on both operational systems and mirrored copies of these systems. For example the ETL team wanted to know "In what system, table and column can I find a student's gender code?". Our initial version has a single system, table and column to store this information. It quickly became apparent that this was not adequate. There were often multiple systems that had this data stored in a variety of locations based on multiple factors. Our data source structures quickly evolved to a one-to many structure and also captured times such as master or slave information and other data that ETL programmers required. XML transforms that generated SQL scripts were also created and used by the ETL team.

Metadata Structures for Impact Analysis

It was also critical for use to be able to produce impact analysis reports before data elements were depreciated or changed. We wanted to know which computer systems were impacted. We did this by adding list of all systems that reference any given data element. Although this information is very useful, it is also time consuming to

keep up-to-date as new systems come on line and older systems are decommissioned. Return on investment for this maintenance should be studied carefully.

Data Element Workflow

When we started gathering information in any new sub-domain we did an analysis of existing documentation. This was usually in the format of unstructured text documents. From these documents a glossary of domain-specific terms was created. We also analyzed the definitions for tables and columns in relational structures. From there we looked candidate data elements. Each data element in the metadata registry was assigned one of three approval status codes:

Initial-draft Implies that the data element has been found in some system or documentation and is being considered for review by a data stewardship team.

Assigned-to-review-team A data stewardship team has accepted this data element for consideration and is reviewing it.

Approved-for-publication The data element has been approved by the data stewardship for publication to all stakeholders.

One of the challenges in this process was to get formal publication sign-off from each data stewardship team. We found that some incentives were needed in this area. As cubes were being built the reports for these cubes were stored in file systems managed by individual domains. We informed each team that these reports would only be guaranteed to run in the future if they only used approved data elements. The more reports were created the more peer pressure there was to finalized each element for publication.

Data Element Relevance and Externalization Factors

We found that each sub-domain had hundreds of potential data elements that could be incorporated into the metadata registry. It is also well known that the larger a registry becomes, the more difficult it is to use. It becomes harder to find a given data element, harder to write unambiguous definitions that differentiate data elements and more difficult list and visualize the structures.

To avoid this problem a criteria list was created to determine the value of a data element in the metadata registry. Although the actual criteria is somewhat specific to each domain and can become very complex, at the core we developed what we called an **externalization factor** for each data element. The externalization factor indicates how "exposed" a data element is to the outside word and potential subscribers and consumers of this data element through web services.

Data elements that are imported or exported from many systems have the highest externalization factor and must have a high priority and resources assigned to them. Data elements that are entered by users or that appear in reports also have high externalization factors. Data elements that are used as temporary intermediate processing between only two systems have a lower externalization factor.

We found that almost all sub-domains required accurate descriptions of Organization, Student and School-Year coding. The creation of clear identifiers for these concepts was also very critical to the accurate creation of conformed dimensions used by multiple data marts.

FUTURE DEVELOPMENTS

Focus on Semantic Integration

Although much of the funding for these projects is being driven by NCLB-funded longitudinal data analysis, the problem of cost-effective and semantically precise data movement between systems is a universal problem [14]. It is not unique to NCLB or data warehouse systems. One of the main goals of this project was to focus on building precise semantics and allow everyone to benefit.

Migration to XML Web Services: Trains and Track Gauges

Most K-12 organizations that we interacted with are still using non-XML data transfers and will continue to do so unless there is substantial new funding from federal or state agencies that encourage migration to XML standards. These data exchanges are frequently done by software vendors that have written custom software for each states data submission standards. There is little or no incentive for these vendors to write new data submission software.

One of the metaphors used to explain the cost dynamics is the train and the track-gauge analogy. It is inexpensive to change a single train engine to run on a slightly different track gauge. It is very expensive to tear up all the train tracks and adjust them to a new gauge. Similarly, once data submission standards are in place and vendors write custom software, these data submissions standards are very expensive to change.

One of the ways to approach this is to build parallel data submission standards. This process would allow schools and districts to submit data in XML in parallel to the established formats. Extending the SIF standard to include multiple-namespace data structures could be one way to reach this objective.

Another strategy is for state and federal agencies to insist on XML data transfers for all data submission interfaces. Although many vendors using older software may resist this, most modern software systems integrate XML processing as part of the software development tools. As newer software is installed at districts that comply with standards such as SIF, the migration to XML and web services will be less revolutionary and should meet with lower resistance from software vendors.

Growth of Business Rules Engines

There has been a great deal of discussion about the high-cost of requiring business rules to be maintained in complex procedural languages such as COBOL, Java, C#, SQL stored procedures or Visual-Basic. A more cost effective approach has been to allow web service

transactions to enter a generic web-service enabled workflow rules engine were edit checks are stored in a business rules engine. The maintenance of these workflows and rules is frequently done using graphical tools. The addition of open-source business rules engines to application servers could dramatically lower the costs of these systems. With training, workflows and business rules can frequently be modified by non-programmers.

Like data warehouses, business rules engines are another integration point in any organization that deals with complex data. The costs of migrating to easy-to-maintain workflow and business rules are dramatically lower if semantically precise metadata definitions are used within the rules engine. Although few educational organizations are using business rule engines today to audit and analyze data submissions, we expect this to be economically feasible if an organization has precisely defined metadata.

RECOMENDATIONS

As a result of this effort the following recommendations are offered by the author.

Recommendations for Data Architects

1. Organizations and applications that exchange data should be encouraged to publish their metadata in an industry standard machine-readable format to facilitate software agent interoperability.
2. Published data dictionaries should drive exchange document creation standards and published web services and metadata registry "shopping cart" tools should be accessible to non-programmers.
3. Data warehouse initiatives should attempt to reuse and integrate existing federal metadata standards.
4. Data architects and integration managers should encourage fundamentals of metadata publishing and transformation training.
5. Metadata standards should continue to be developed with the goal of building semantic integration brokers and agents.
6. Producers of data mapping software should integrate semantic equivalency statements into automated mapping systems.
7. XML integration appliance vendors should include semantic integration services to make integration easier.
8. Organizations should perform ROI analysis on semantic integration.
9. Awards should be given to organizations that publish useful and high-quality metadata.

Recommendations for Federal, State and Local Agencies

1. Federal and state agencies should follow ISO/IEC 11179 and Data Reference Model (DRM) guidelines and follow best practices such as precise definition, three-part data element names and formal representation terms for all data element properties.
2. Grants should be given to agencies that agree to publish not just their data elements, but also the semantic equivalence of their metadata to centralized metadata registries such as the NIEM.
3. Metadata projects should each include two or three staff members that are proficient with XML mapping and XML transformation.
4. As well as the use of NIEM standards, other semantically precise XML standards that are already being used by federal agencies such as ebXML and the Business XML Reporting Language (XBRL) should also be investigated for semantic precision and overall cost effectiveness.

Recommendations for Metadata Standard Bodies

Some standards bodies sometimes spend a great deal of time focused on the proper ways to describe internal metadata registry structures. However as metadata registries become popular, external interfaces become critical for interoperability. Highly visible items such as data element definitions, XML data element naming conventions, the appropriate use of upper ontologies and representation terms become critical for interoperability. This is especially true while humans are still involved in data element mapping. Metadata standards organizations need to step forward and analyze these best practices and codify them. There are the several main areas that need to be standardized:

1. A common well-defined upper ontology for items such as Activity, Document, Organization and Person based on industry actual usage.
2. A standardized XML data element naming convention such as the three-part Object-Property-Term upper camel case data element name.
3. The promotion of shopping-cart style subschema generators.
4. A list of approved representation terms and detailed documentation of when to use these terms.
5. Mapping of metadata registry structures into metadata publishing structures such as OWL.

The lack of clear standards in these areas may not be appealing to many academic researchers. However they were major obstacles in building a semantically precise metadata registry in this project.

Toward the Semantic Wiki

In addition to the lack of these standards the process of clarifying and extending metadata structures is a very cumbersome process for most standards organizations. As a browser of a metadata registry finds the data element they are looking for there are no easy-to-use tools to document how a data element is being used in practice, who is using a

given data element, what structures are similar to this data element, what extensions are proposed for a data element, and what are the known problems with a data element. Getting a new data element integrated into a metadata standard is a multi-year process.

This could all change if metadata standards bodies moved toward a more collaborative environment for publishing metadata standards. The rise of the Wikipedia has been an existence proof of how disparate contributors from around the world can collaborate to build high-quality systems. Current Wiki projects also need to implement features that can allow for better semantics and version and release publishing. This would allow carefully built "snapshots" of metadata wiki driven standards that can be commonly references.

If on-line learning education experts in China or India have insights on how courseware metadata could be structured, their ideas should have equal merit. Today some standards organizations required expensive membership to contribute their ideas. Open-documentation systems such as the Moodle-documentation Wiki are far more likely to have intelligent contributions and rapid worldwide peer review due to the nature of Wiki's structure. Metadata standards organizations should stay informed with the open-documentation standardization growth and evolution rates. The principals of economic Darwinism may apply to metadata standards organization.

Data Element Folksonomies: Metadata for your Metadata

As metadata registries grow in size, one of the key challenges is finding the correct data element that meets your business requirements. For example searching a metadata registry for the keywords "Individual**"** or "Human**"** may give the searcher no indication that the word "Person**"** is actually used to describe an instance of a homo-sapian. The lack of metadata registry structures to suggest synonyms and recommend preferred data elements has also complicated this problem.

One of the most recent developments in web searching has been the use of ah-hoc user-contributed tags to data. The use of these tags has been recently referred to as Folksonomies [17]. Folksonomies allows hard-to-find items such as images, URLs and web logs (blogs) on web sites such as Flickr.com, Technorati.com and del.icio.us. These tags allow users to quickly search large data sets using similar tags to find the items that are relevant.

Although Folksonomies have their own problems, they do avoid the costs of having experts in ontology development take multiple-years to codify a subject-domain. If items are hard to find or incorrectly classified, thousands of users can quickly re-classify items using refined education metadata.

For example semantic wiki extensions to systems such as Wikipedia could allow it to be a repository for reusable learning objects that could be quickly integrated into learning management systems. This would allow instructors to quickly find reusable learning objects that apply to their course and their state assessment standards and integrate them into their curriculum.

Analogies for the Future

There are a few insightful analogies we have used for this project.

Metadata Publishing and Web Publishing: Many of the same factors that are at work publishing a highly visible web site also apply to publishing influential metadata. The institutional commitment to review, change-control and persistence of URLs that we have seen in the Dublin Core standards must be widely adopted by educational metadata publishers.

Synonyms and DNS: Semantic equivalence mapping needs to be integrated into the infrastructure of the web before semantic brokers and semantic agents are viable on a wide scale basis. The ability to look up a mapping between an IP address and a domain name today is part of the infrastructure today. We need some world-wide synonym registries with the reliability of DNS to allow semantic brokers to cut integration costs.

ARPANET/DAML and NIEM: We can't predict the future of the NIEM as a central repository of federal metadata semantics. But we know that in the past federal projects like ARPANET and DAML have led the way for standards to be created. We feel that following the NIEM or its successors will be critical for K-12 data architects.

HTML `<a>` and `owl:equivalentClass` We know that search engines have exploited document linking using the HTML anchor <a> tag to find authoritative web pages. We expect metadata search engines to use the `owl:equivalentClass` and `owl:equivalentProperty` in published metadata to create similar rankings in the future.

SUMMARY: THE FUTURE OF EDUCATION METADATA

Today taxpayers around the world pay billons of dollar of additional taxes due to the inefficiency and inflexibility of legacy K-12 data standards. States, school districts, and teachers are burdened with needless paperwork that requires manual data re-entry for re-transmitting duplicate data to meet audit and compliance requirements.

Every day teachers around the world re-create lesson plans, course content, and quizzes from scratch that could be shared if they could be cost-effectively stored and retrieved. Many of these challenges have their root in the slow adoption of K-12 education metadata standards, metadata management and precise semantics. Our hope is that the insights gained in this project aids K-12 data architects around the world to effectively use semantic web technologies and other metadata management strategies and allow everyone to spend more time with our children and less time dealing with inefficient computer systems.

ACKNOWLEDGMENTS

I would like to acknowledge the Information Technology staff at the Minnesota Department of Education including

Kathy Wagner, John Paulson, Dan Fitzgerald and Brian Hecht. At the Wisconsin Department of Instruction I would also like to acknowledge Par Jason Engle. At the Wisconsin Center for Education Research I would like to thank Rob Meyer for his contributions to this project.

REFERENCES

1.NIEM - National Information Exchange Model http://www.niem.gov

2.ISO/IEC Metadata Registry Specification http://en.wikipedia.org/wiki/ISO/IEC_11179

3. See US Department of Education Press Release dated November 18, 2005 *14 States Win $52.8 Million in Grants for Longitudinal Data Systems* from http://www.ed.gov/news/pressreleases/2005/11/11182005a.html as part of the US Department of Education Statewide Longitudinal Data Systems Grant CFDA Number: 84.372 http://www.ed.gov/programs/slds/

4.National Center for Educational Statistics data (NCES) handbook web site: http://nces.ed.gov/programs/handbook/

5.Education Data Exchange Network (EDEN) http://www.ed.gov/about/offices/list/ous/sas/pbdmi/eden/workbook.doc

6.The Common Core of Data (CCD) http://nces.ed.gov/ccd/

7.School Interoperability Framework (SIF) http://www.sifinfo.org/

8.*Federal XML Developers Guide* published in April 2002 http://xml.gov/documents/in_progress/developersguide.pdf

9. US Department of Justice Global Justice XML Data Model http://it.ojp.gov/jxdm/

10.Georgia Technical Research Institute (GTRI) http://justicexml.gtri.gatech.edu/

11. Web Ontology Language (OWL) http://www.w3.org/2004/OWL/

12.Michael C. Daconta, Leo J. Obrst, Kevin T. Smith *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*, John Wiley & Sons, 2003

13.For a description of how to write data element definitions see: http://en.wikipedia.org/wiki/Data_element_definition

14. Moodle Open-Source Learning Management system http://moodle.org/

15.Sample OWL file for K-12 Education http://education.state.mn.us/datadictionary/owl/Education.owl

16.McComb, Dave. *Semantics in Business Systems*. Morgan Kaufmann, 2003 pp 453-469.

17.*Folksonmies: Tidying up Tags* See by Guy Marieke http://www.dlib.org/dlib/january06/guy/01guy.html