Flexible Metadata Definition and MetaManager Distributed Metadata System:
A Metadata Sandbox Where Everyone Can Play

White Paper

by William Dan Terry

Proprietary - William Dan Terry, Knotworks

<CVS: $Id: FMD_MM_whitepaper_20010222.txt,v 1.1.1.1 2001/06/08 15:55:44 wterry Exp $>

(DRAFT)

22 Feb 2001

//-- Background

Currently, a number of metadata definitions and initiatives are vying for attention. The basic problem with all is that each is focused and not truly designed so that it could effectively be a complete standard for all to use. Within each domain they can provide completeness. But in a highly networked world of heterogeneous data they offer limited direct correlation or require extensive means for determining correlation. There are invariably elements of metadata in each that are missing or extraneous for any other particular use. Each effort to come up with a standard appears to bear the imprint of the focus of the group creating it. Part of the problems stems from the two disparate goals of most, if not all, metadata initiatives. One goal is to accurately and uniquely describe an object no matter what kind of object, opening the door for a deluge of elements. The other is the desire to make the metadata record compact and easily usable, requiring the weeding out of many elements that could otherwise be useful for various uses.

At one end of the spectrum, implementations will contain sparse records and still require processing of empty elements to determine that they are empty. At the other end of the spectrum, there simply will only be a cursory definition of the object. Realistically, we can’t expect an organization that captures digital renderings of invaluable texts to be happy with 10 or 20 standardized metadata elements. Nor does that organization want to manage 20 times the metadata elements than they need even if they remain empty. Any compromise between the two goals will invariably leave metadata elements out that were important to some purpose.

Just selecting what metadata elements are useful for a particular use can serve an organization well. But part of the drive for a metadata standard is the desire to be able to map across collections. If every organization selects which metadata elements they’ll use from the full standard set, the mapping between collections can easily be expected to require the determination of the mapping of 100 to 200 or more elements for every search.

Another problem with current metadata efforts is the lack of inheritance. With current schema there will be a fair amount of redundant information. Imagine the basic geographic metadata associated with every photograph of Leningrad. Then imagine the effort required to update those to include a St. Petersburg designation.

A flexible metadata definition is required to provide a metadata option not adequately covered by the current options and to provide efficient correlation across other metadata options. In concert with this is the need for a technology that can locate objects regardless of the metadata definition used to describe them.

//-- Document Notes

For clarification, I’m going to formalize some terms. An "item" is a thing that stands on its own. Don’t think of this in physical terms - a pixel can be an item even though it is part of a digital image. And what qualifies as an item in one use won’t necessarily in another. In mathematics "12" is an item. It is a positive whole number with a set of properties - even, factored by 3, etcetera. "12" as related to that pixel is not an item. It has no meaning without context - the Red value in an RGB color definition of the pixel 82 right from the left of the image and 46 down from the top of the image gives the "12" meaning. A metadata record is used to describe an item.

For the purposes of this document the term "item" will be used interchangeably with the term "object", regardless of the format or the method of presentation, i.e. word processed document, audio, video, image, painting, etcetera.

//-- Flexible Metadata Definition

The Flexible Metadata Definition (FMD) is not a monolith, rather it is comprised of a core block and a set of extension blocks. The core metadata block contains the elements that every object will need. The metadata extension blocks are based on function with a block per function. A metadata record would consist of the core block and any applicable extension blocks required to fully describe the object.

For instance, there would be an image extension, photo extension, a publisher extension, a legal extension, and an intellectual property extension block. A patent would be defined using the core block and the image (for "patent art"), text, publisher, and legal extension blocks.

To enable useful definitions of all objects there will be a large assortment of extensions. Efficiencies are derived in implementations for specific uses as they only need employ the core block and a small number of applicable extension blocks. After all, many corpora are somewhat focused due to the nature and reason of their existence. An art museum's metadata needs only slightly overlaps with those of a natural science museum.

While this may seem like just employing whatever elements are desired, the difference is that FMD defines what blocks are used, and therefore what sets of elements, within itself. This brings data management and cross collection mapping down to the mapping of a small number of standard blocks instead of hundreds of standard elements. This provides a complete federation of data elements across all implementations with a minimal of metadata metadata.

For instance, a search for images of the Arc de Triumph would only need to access the core, image, and possibly architecture blocks, while cleanly ignoring the painting block of an art gallery and the photo block of a photography gallery.

Basically, the concept is trying to minimize sparse databases (sparse matrices) by using selectable function-specific blocks and inheritence. The idea is not to define a set of elements for everyone to use, but to define a set of elements in a block so that if that block were applicable to the person's needs, basically most elements in that block would be used, thereby minimizing empty fields.

For instance, the National Museum of Art would probably use less than 2/3 of the Dublin Core (10 out of the 15 - not use Publisher, Contributor, Source, Language, and Relation - others like Coverage and Rights are questionable). Certainly, databases can be designed to save storage space for empty fields in a record. But this doesn't address the logical space that needs to be processed for a query that is using all 15 fields - the database has to still check the field and see that it's empty or to filter the query accordingly. When utilizing a standard with the Dublin Core approach but that is much more extensive, the overhead associated with inapplicable fields can become quite extensive.

It's a subtle difference, but it can have grave affects on scalability.

In FMD it is possible that there will be the need for extension blocks specific to other metadata formats. These would contain only those elements that are specific to the foreign metadata format and that have no correlation to standard function-based extension blocks.

One of the beauties of this approach is its extensibility while retaining backwards compatibility. Every metadata element and functional group doesn't need to be envisioned now. While the wine industry may not be driving any metadata standards right now, at some time in the not too distant future they may appreciate being able to describe the bouquet of a wine in a database as part of their other metadata.

While this amounts to the potential for a large number of extension blocks, it would be surprising if we could really subsist on a small subset. Catalogers are already well aware of the issues involved in describing print documents alone. Now add all of the more recent media of audio, video and web pages. It's not inconceivable that there will be streaming olfactory to go along with our streaming audio and video. If it seems far-fetched, consider how much of what we have now was far-fetched 50 years ago. We now have a new field called sound design. It is the field of crafting sounds for particular uses, such as music, movies, and computer games. There will be a need for metadata for that. What's next? (Note: In April of 2001 I came across an article regarding a company based in Oakland, CA, USA which is developing streaming olfactory. The hardware one purchases is a machine to connect to a computer containing 128 scent oils which would be vaporized in various portions to emulate comment smells.)

With all of these metadata needs in each collection, the reality is that only a smaller percentage of metadata elements will

probably be leveraged for cross corpora searching. But FDM provides a coherent means for structuring the common metadata in concert with the esoteric metadata without loss of descriptive power or loss of efficiency. It is possible that with such a data architecture in concert with the system described later that a person could search on "Pictures at an Exhibition" and get listings for audio objects of Moussorgsky's music, books discussing this piece of music, and data on the painting of Baba Yaga's hut on chicken legs.

This data architecture can, in a worst case scenario, require more processing to search and access than an architecture based on a single comprehensive record. However, there are a number of problems inherent in the latter architecture approach. The single comprehensive record approach can get exceedingly large even if a particular use only requires a small percentage of the data elements. The rest of the elements must either be carried anyway to maintain compatibility with other systems using the same metadata definition, or a means of accounting for the custom differences must be built on a case by case basis. And both solutions will probably require major revisions to the system every time the metadata definition standard is enhanced.

The modular approach to the metadata definition defined herein allows a user to create a semi-customized solution, using just the standard modules that apply, to both minimize the implementation to just what is needed, while retaining compatibility with other systems using the same metadata definition. And enhancements to the metadata definition only need to be incorporated into a particular system if they apply directly to the modules in it, instead having to deal with metadata definition enhancements that are superfluous.

//-- Set Membership and Metadata Heirarchy - A Means of Implementing Inheritance

There are many situations where a corpora of metadata can have many unlike items and the items can describe a hierarchy of sets. A record for the "Solar System" can have metadata describing its position in the Milky Way. A record for the "Earth" can have metadata describing its mass and average radius. And a means to define that the Earth is a member of the Solar System set provides the means for letting the Earth "share" properties of its container, the Solar System item.

So, what we have is a "member of" relationship that allows us to pass properties down the hierarchy of sets, allowing set properties to apply to members of the set.

Another example of the use of this is to consider a collection of furnature. One item defined by a metadata record may be the generic chair set. It may describe a generic chair as having four legs and providing the seating for one person. A record for the office desk chair might define the human posture oriented backrest and rollers on the legs and define itself as a member of the generic chair set.

There are two ways to interpret this coverage of metadata for an item that is the member of a set - as a union or as a distinct set of layers. For instance, a record may exist that describes a collection of digital images. It may use extension blocks 3, 8, 17, and 24 (I'm just using numbers to illustrate the concept). Records for works (the basic member of the collection versus "derivatives" such as GIF, thumbnail, etc. of the same image) in that collection may use extension blocks 2, 3, 6, and 12. And records for derivitives may use extension blocks 5 and 11. Metadata about a thumbnail could be interpretted to include metadata for the collection, the work, and the derivative. In cases where similar extension blocks occur in the heirarchy (core blocks are always considered unique to each item) there is the option to outline the heirarchy and the metadata associated with each level or to just interpret the "lower level" block as superseding the "upper level" block and consider it a single description of the thumbnail. In this case, using the union approach data in extension block 3 of the work would supersede the data in extension block 3 of the collection.

Just to clarify however, the items don't have to have any relationships at all. This is just a feature built in to FMD allowing the leveraging of heirarchies.

//-- FMD Mappings

Part of FMD is the definition of mappings to other metadata definitions. As FMD is based on assorted functional blocks, mappings will be defined that correlate the elements of other metadata formats to the appropriate combination of FMD blocks and block elements. These mappings can be used to search across metadata formats.

For example, the FMD definition for the Dublin Core mapping would be something like core.name, core.creator, topical.subject & topical.keywords, topical.description, publish.publisher, publish.contributor, core.createDate, topical.type, core.format, core.identifier, derivative.source, language.original, derivative.relation, topical.coverage, intellectualproperty.rights, using elements from the core, topical, publish, derivative, language, and intellectualproperty blocks.

//-- MetaManager Distributed Metadata System

While FMD is designed to stand on its own, it is easy to envision the creation of a distributed system architecture that organizations could use as the base of their metadata management implementations and automatically provide integrated, cooperative object location on a network across corpora and across organizations.

The MetaManager Distributed Metadata System is being based on the Dynamic Delegation Discovery System (DDDS) standard being developed by the URN IETF. The architecture is similar in concept to DNS, providing for a loosely hierarchical collection of nodes where the nodes are managed independently, generally by the organization responsible for the objects being described by the metadata. By utilizing the DDDS standard, not only is the system based on a sound design, but it enjoys the favor of an open standard, which generally has easier public buy-in than proprietary systems.

An organization that wishes to make their corpora searchable could install a MetaManager system, configure it for the metadata extension blocks that apply to their corpora, register it with a higher level node, and let it handle incoming search queries. Or an organization could configure it to belong to a closed network of nodes, as in a consortium or in a trusted-source cooperative. Or an organization could configure their MetaManager node to not communicate with any other nodes, and utilize the system purely for internal use.

Another feature that is important in the maintenance of large, or even small, corpora is the need to control read and write access of the data. This eliminates the problems of work being overwritten accidently, or intentionally, by others. Along these lines reading access to the data can also be controlled so that only approved data is available to others or is limited within a particular defined group of users.

//-- Conclusion

FMD and MetaManager are squarely aimed at serving all metadata needs efficiently and flexibly in concert with the opportunity to share information collaboratively and openly with systems built on these and even those that are not. Think of it more as a toolbox of inter-related metadata tools where everyone can pick the particular tools they need while still enjoying the benefits derived from having everyone utilizing the same toolset. FMD and MetaManager provide a metadata sandbox where everyone can play.

Proprietary - William Dan Terry, Knotworks

_____K_N_O_T_W_O_R_K_S_____

http://www.knotworks.com