Expert Committee Meeting

October 24-26, 2004
Ann Arbor, MI

Present:
Mark Diggory, Pascal Heus, Arofan Gregory, Sanda Ionescu, Peter Joftis, I-Lin Kuo, Ken Miller, Jostein Ryssevik, Wendy Thomas, Mary Vardigan, Achim Wackerow, Oliver Watteler

Sunday, October 24

Chair: Jostein Ryssevik

Jostein opened the meeting by thanking everyone for attending. The SRG working group has accomplished a great deal by telephone, but having a face-to-face meeting is important and appropriate at this time in the development of the data model. This will be an internal working session that will permit Arofan to go away and do the first draft of the technical implementation.

Basic outline of the three days:

Sunday:

1. Review the metadata models that we might learn from and that overlap the DDI; discover which might have approaches that we can benefit from.
2. Review comment log disposition and formal procedures for comments.

Monday:

Focus shifts to the DDI model itself - the model as it exists now and how we will attempt to change it.

Tuesday:

Decision Day. Based on the conclusions of the two previous days, we will focus in on the areas that are important to start formalizing the model.

Wendy has created an SRG timetable, to which Jostein has added a red line indicating where we are now. We are delayed on some issues. This meeting will allow us to catch up.

We have created a new conceptual model that steps away from the archival model and moves us in the direction of a lifecycle model, focusing on other parts of the complete lifecycle of a dataset from conception to archiving and use. We need to see how parts of the existing DDI fit into this model. Wendy has created preliminary sketches based on this underlying lifecycle. We need to focus on areas in which the existing model doesn't function well, determine where there is a need to change logic, and try to remove inconsistencies in the model. We are laying the foundation for a modular DDI. Arofan wants to have a first version available for review by the end of December. Then we can start involving other working groups.

Arofan described a process model for reviews. There are a couple of different ways of doing reviews and making progress. Sourceforge is good for doing development, but is not an acceptable way to do public reviews. We will have a schema and will use sourceforge to track its development. We will also run a formal disposition log. We will freeze a schema version in sourceforge with line numbers in PDF so that we are all talking about the same thing. Using a comment log, once a comment has been received and disposed of, the comment raised cannot be raised

again. We need to make it obvious what people need to download to review and how to make comments. The disposition column of the comment log will be filled in by the SRG or some subset of the SRG. Disposition classes will be defined.

There will be a person or an editorial team that compiles the log. Everybody uses a template for submitting comments, and then the log is turned into a master PDF that is version controlled. We need to encourage the submission of solutions for complaints - that is, people must suggest a solution to the problem they are describing. If there are two conflicting comments, we can go to the two suggesters for further discussion.

The comments log is in Word format right now. Arofan will be the editor, and Mary Vardigan will collect comments and update the Web site. The SRG group disposes of comments, but we can perhaps rotate responsibility within the group.

Sourceforge has some maintenance issues now coming through. Bug fixes are to be disposed of quickly. Other proposals for changes to 2.0 will be voted on. All SRG members need to sign up for sourceforge. The process is that Tom, as Expert Committee Chair, will review initial submissions and then migrate them in sourceforge to Technical Review status. Sourceforge permits us to provide comments attached to the issue. The trackers were built to be public, but we are using them only for the SRG. We will have a separate mechanism outside of sourceforge to do the voting.

Other metadata models were discussed. There were 16 separate standards considered (see corresponding table). During the discussion, some items were moved to a "parking lot" for later deliberation. These items were later prioritized A, B, or C, depending on level of urgency.

Parking Lot:

1. Fixity (checksums, encryption, etc.) - C
2. Processing history of digital object - C
3. Identifiers (URI format-bound). Regarding ids, need unique ids across set of instances to combine. - B+
4. Atomic microdata; what does hierarchy look like for our data? What is DDI "item"? Hierarchy re ISADG - DONE
5. Linking (METS, etc.). xlink is a barrier to adopting; METS not something we could copy - B
6. Embeddable metadata; metadata embedded in data objects? Data embedded in metadata; real-time updating by tools - C
7. Multiple doctypes approach (comparative data, etc.) - B
8. Data typings - A
9. Agencies and roles - B
10. Reusable MD "core" but not problematic - A
11. Multi-lingual (I-Lin's proposal module) - B
12. External codes - DONE
13. "Tabulation" module? Relationship to cube module? - C
14. Administration" modules (or as part of archive module?) - B
15. Add module to describe sampling? - A
16. DDI Lite? Profiles for communities? - C
17. Standard formal expression of computations - C
18. Comparative data and how it fits in design - B
19. Physical storage (look at CWM models) - A
20. Standard data format at some point? - C
21. How to deal with documentation? Dublin Core? Controlled vocabulary in terms of type - A
22. Lists. Mechanism for reuse. - C

23. Maintaining controlled vocabulary lists. Other standards allow you to incorporate other schemes and validate. UBL built mechanism for referencing external code lists. - A
24. External classifications in XML. Neuchatel group? CLASET? (CLASET is in the public domain and SDMX is working on a subset of this large standard). EU funded project statistical offices. - C
25. Namespace design rules - A
26. Versioning - A
27. Migrating issues from V 2.0 to V 3.0 - B+

Monday, October 25, 2004

Chair: Arofan Gregory

Pascal presented information on a new initiative at the World Bank that integrates DDI. This involves a Development Data Platform (DDP), which is a next generation data management information system for macroeconomic databases plus survey data. There is a new International Household Survey Network (ISHN) being established and the DDI is being adopted as the metadata standard for exchange.

For DDP microdata, there is a survey repository catalog that uses DDI section 2, with 1300 surveys and 400 datasets. Survey analysis is performed through Nesstar based on about 30 surveys.

In terms of availability, the initiative will go up on the project Intranet this week. It will be available to selected external partners by the end of 2004 and to the general public by mid-2005.

IHSN comes out of the Millennium Development Goals, and is intended to track major indicators. The recent Marrakech Action Plan for Statistics laid out six goals, one of which was the IHSN. The IHSN will collect better data, make more data available, and improve comparability. The project leaders first met in June 2004, then in October in Geneva. Potential IHSN participants include international banks, UN agencies, WHO, UNICEF, etc. There will be 50-100 participating countries.

Work is being done now on development of survey data dissemination tools and guidelines. There will be a Data Dissemination Toolkit, with simple tools for producers to archive and disseminate metadata based on DDI. The goal will be to make documentation available in PDF, data in Nesstar NSDStat file format with a free NSDStat reader. The main components of the Toolkit will be a metadata editor (an enhanced version of the Nesstar Publisher that will allow documenting hierarchical files together); a CD-ROM builder that will record on CD-ROMs the output of the Publisher; and a set of guidelines on metadata creation.

The project will further enhance the Nesstar Publisher by adding tools for quality control, subsetting, recoding, and anonymization. There will be custom branding and licensing agreements to the Publisher and CD Builder (CDs have been successful in these countries and agencies). The schedule is for betatesting in early 2005 and release in April 2005. This means that 50-100 countries will produce DDI markup. The WHO will use it for World Fertility Surveys.

The data repository will be searchable by country, year, type of survey, and topics. Search results will include an overview of the survey(s) (DDI study description, Section 2.0), the survey instrument, and links to data, and related reports and other publications. The survey metadata will be retrievable in several different formats, including DDI and HTML. It will be possible to analyze data from a single survey, or across multiple surveys for which subsets of data would have been previously harmonized.

For the DDI community, the main significance of this project will be the broad adoption of the DDI standard by several large international organizations, as well as an important number of data collectors/producers, as noted above. The project will also help bridge the present gap between data users and producers through their links with the data disseminator. The new visibility of these data will also be an incentive for producers to strive to improve data quality.

The SRG began discussing the projected data model.

The goal of the modular structure is to be able to encapsulate, swap, and add information. Modules can be updated individually, without affecting the whole structure.

The wrapper for all modules should be message and protocol neutral. It should provide functional information about the content, and should list all contents, providing a structural map of what pieces are included and how they are related.

How do we treat multilingual instances? Will there be a module for each language? One solution for covering such instances would be to identify which elements can contain translated text, and enable a language string to define language used as well as other characteristics, like:

<Title> <text original="true" xml:lang="EN"> <text xml:lang="FR"> </Title>

It is generally agreed that having individual modules for translations is not a good idea. However, the solution mentioned above would permit building individual files, if they appeared necessary.

As far as the basic design of the model is concerned, a choice may be made between more simplicity, which implies tighter control, or more flexibility, which involves greater complexity. It is generally agreed that we will strive for a model that's as simple as possible, but not simpler.

The possibility of having a "fonds"-type structure, involving hierarchical levels in addition to modularity, was discussed. The "fonds" model, currently used to describe collections of records, involves the following levels - fonds, subfonds, series, subseries, file, item. What would we include at each level? Are they all necessary?

One advantage of such a structure is that information inserted at one level would apply to everything below, and need not be repeated. One disadvantage is that a fixed-level structure appears too rigid for our purposes. Recursive generic grouping also reduces simplicity.

In connection with this hierarchical levels structure, the question also arises as to what we would describe at the lowest level. A fundamental difference between microdata and aggregate data is that the lowest level of description in microdata is the variable, while in aggregate data it is a single cell in a cube. There will be therefore two different descriptive structures at the lowest level, depending on the type of data.

For the overall model, a simpler hierarchical structure could be applied, in which there need not be too many levels, but the same principle will apply, according to which upper-level modules point to the ones below, and not the other way round. DDI instances (or "products") will be situated at a lower level, while additional information, related materials, and administrative-type documentation will be situated above, and will apply to all the lower-level modules they point to. At the same time, information at the lower level overrides information from above, when the two are different (this principle would allow versioning by module, without necessarily affecting every other module).

How are the lower-level modules -- the DDI instances themselves -- to be documented?

It was noted that variable and dimension are basically the same thing - representations of concepts. But they differ in defining values. Continuous variables are measures, while discrete variables are dimensions.

What terms are we using and what do they describe?

We noted some definitions:

Survey: Designed to gather information from a population of interest in a specific geography at a specific time.
Dataset: Data structure resulting from one or several survey. It may have one or more physical storages. Datasets are at the core of DDI.
Study: One or more surveys or cubes.
Microdata: Not all microdata are survey data. Microdata result from a survey or another process.

Measure: Measure is metadata that applies at the cell level and not to the structure of a cube. We can have different cubes that only differ in terms of measures. What is the relationship between measure and variable? Several input variables can create a single measure. The unit of measure in SDMX is an attribute at the series level. Measure can be a variable or a function of a variable. Measure is different from the aggregation mechanism.

We want to provide for variables that can be both continuous and discrete at the same time - both behaviors. If it is a continuous age variable, one may want to create age groups using categories so we could add one or more recoding structures that software could use.

We want to distinguish between:

- Documenting the instrument
- Documenting the relationship between the instrument and the dataset

Questions: They may be simple or multiple response; they may also have flowchecks, or elements that control the flow of questions. What is missing right now in the DDI is the concept of the flow through the questionnaire, which is hard to capture. We should look at IQML or TADEQ or other survey production packages, which may help with this.

Variables may be of several types, including raw, recode, imputation flag, weight, id, classification, computed, derived, key variables. A question goes directly to a raw variable, but the path is more complicated for a recode.

"Many to manyness" depends on variable type.

In multilingual surveys, do we want to link to different variable? It depends on how they are set up. Links should point from variable to question, not the other direction.

Where do we put descriptions of variables that are not based on questions? There should be something for the data-generating database queries that is equivalent to questionnaires for survey data; this would be process-derived data.

Questions and variables reference concepts. What do we point to as sources for secondary analysis?

The questions side of the development of DDI 3.0 belongs to the Instrument Documentation Working Group, so we need to follow up with them on these issues.

Tuesday, October 26

Chair: Wendy Thomas

Chris Nelson will send us the DTD for IQML, which we looked at only briefly. We need to recruit new members for the Instrument Documentation group.

It is important to revisit the issue of physical storage in the model because people are not happy with the physical storage information currently in the current DDI. We should have a flexible interface to the physical stores in various formats. Do we have to take into account all the different formats? A URI isn't sufficient. What we have now is one model fits all. The warehouse model is one way to model physical stores. That model could be extendable to incorporate proprietary packages. We could have a generic layer in DDI, and the user could describe specific the data source in plug-in module. We will "parking lot" this issue.

Record, relational, multidimensional, and xml are the data types in the Common Warehouse Model. The traditional data in the DDI world is records data. Let's pick up on what OMG has already done with the CWM.

If one has metadata in DDI, one is always at the mercy of proprietary formats. There may be an advantage to having our own format, but we don't want to reinvent a format and structure. Strategically, we should concentrate on metadata right now, which is our strength. We should solidify that and then there may be impetus to move to a

common data format, perhaps for a Version 4.0. Triple-S has succeeded because it includes a way of encoding the data and has become an exchange format.

We now need to take Wendy's modules, drill down, and rearrange them according to the group's new conceptual model. In terms of modeling software, Wendy used Poseidon, and that is the default suggestion. I-Lin has used it to do basic things. Mark has been using a module from Eclipse and will send around links to this. We can start with Poseidon and look into using Eclipse to do class diagrams. People should use whatever tool they feel most comfortable with.

The namespace for DDI is now icpsr.umich.edu. This is a 3.0 issue. There are rules about how one publishes URIs for namespaces.

The group went back to examine the Parking Lot issues in greater detail:

1. Typing in schema. PCDATA is flexible but encourages misuse. How do we get information for each type on a field-by-field basis? We can start with what's in 2.0 already, and then just figure out what we are adding in 3.0. Wendy will go through the current elements and determine those that require valid values and recommend for comment. The goal is to disallow abuse. We should type as strongly as we can get away with; relaxing changes are easier because they are noninvalidating. Thus, we should err on the side of overinclusion. We also need to get clearer and more explicit assumptions about what needs to be human- vs. machine-readable. Date is a good example: The archives need this to be both human-readable and machine-readable. As of 3.0, do we want a DTD equivalent to a schema? There is a transition issue. Should we ask this of a broader group? When making decisions on elements vs. attributes, there are implications if we are building a DTD or a schema. We should move ahead with good schema design and not limit ourselves by a DTD. That said, we may want to turn out a low-effort DTD. The canonical version of the DDI, however, will be a schema. It's possible that the "Lite" versions we envision may be DTD-based. So we will in effect be reversing what we do now with 2.0, which is that we have a canonical DTD with corresponding Schema.
2. Reusable metadata core. Citation elements are a good example of this concept. They can be pulled from the core into specific fields. This is a good design for DDI. More than reusable types, it is namespace based.
3. Add a sampling module? This could contain stratification and PSU variables. Pascal will supply an elements list. Some of this will allow applications to take sampling into account. The current DTD has the descriptive part of sampling, but there is a technical part that isn't in the specification now. We may be able to compute error estimates with well-specified sampling fields that provide information to program against. This may be part of the methodology module and not a separate module in itself.
4. Physical modules. Arofan will get a copy of the Common Warehouse Model (CWM) and start from there.
5. How to deal with documentation? What does documentation mean in this context? In 2.0, there are fields for external related material. Should we type this in terms of kinds of material? At what level do we include this? As you move through the lifecycle, there are certain documents that are relevant. It makes sense to have a generic referencing mechanism but to apply local semantics based on the module. We should create a type that is Dublin Core information with a controlled vocabulary if necessary and present a URI for this resource. This would be an external reference type, URI and Dublin Core. These two are declared as existing in the core namespace module. In the archive module, we would declare the element in the instance Related Publications. "A:RelPub" contents will be of the type declared or reference an ID. We would lay out in the schema the recognized enumeration and allow other types to be used also - that is, an enumeration plus string. This would permit local extensions that wouldn't be machine-actionable. Wendy (with Mary and Ken) will analyze the types of other materials we know about and get back to the group with a list.
6. Maintaining controlled vocabularies (how extensive, how manage, etc.) UBL uses external codes of two types: (1) Codelists, or lists of enumerated values like currency code, country codes, etc.; and (2) Keywords. The idea is to point out to another scheme, point to the agency that maintains it, and give it a URL. This allows for binding a schema description of values and allows people to add to the document. Right now the URI we

provide points at any type of file; it can be a PDF that you can't validate against. We should design a standard format for major cases. Validation is needed at authoring time. Concept is different from keyword and is fixed, not meant to have synonyms. Ncubes should have concepts, which they do. We also have in Version 2.0 controlled vocabularies hardwired into the DTD. We should distinguish between content vocabularies and qualifiers - lists of semantic distinctions that the DDI produces for its own purposes. The latter should be hardwired into the schema. The value of these codes informs the semantics for an action. For Source = archive/producer, we need to come up with a better list. Content should not be hardwired in. For keywords, we should use the approach in DDI 2.0, add a little more structure for the path, and add the ability to give concepts IDs. Enumerations should be treated differently based on whether they are structural qualifiers or content. Content should be treated like keywords, while structural qualifiers should be enumerated in the schema. We will decide others on a case-by-case basis.

7. Guide: There is a need to provide information about case/record identification, but this will actually be addressed in the complex files area. Processing instructions, which are annotations proprietary to the system processing the instance, should be included.
8. Namespace Design. This is related to visibility rules. One technique for modularizing schemas is that schema instances import other namespaces; each module could be its own namespace. This creates a natural packaging mechanism and is good for versioning. We can version namespaces. Once we publish a namespace, it never changes, and each version gets its own namespace. We should keep this to a limited set of namespaces that are functionally aligned. A version is one of the things that characterizes the namespace.
9. Versioning of instances. We need to think about real things vs. virtual things vs. transient stuff (generated on the fly). Do we give an ID to the instance of a session ID? There is another kind of situation in which new data are generated everyday on an ongoing basis. Is the DDI instance a snapshot of the metadata at a point in the lifecycle? What are we versioning? What are the reasons we might want to version instances? There are (1) studies; (2) metadata (permanent and physical storage); (3) xml instances (don't version, this is transient). We need to think also in terms of products. Each product has only one maintenance agency; having a registry in the European scenario where there are many archive products based on the same data is a good thing. Language equivalents should be viewed as multiple products.
10. Migration issues. We want to create a path to move existing instances into 3.0 using XSLT. How data typing in 3.0 affects this will depend on how instances using 2.0 were created. We need to think about creating a version of a migration stylesheet that we can then adjust for people depending on their unique usage.

Three changes to 2.0 are in the pipeline right now: (1) Bug fix of xml-lang to xml:lang; (2) nested categories proposal (reinstate nested categories because category groups do not work for every situation); (3) introducing use of XHTML 1.1 for formatting blocks of text (rather than using TEI tags).

What kind of input do we need from the subgroups to proceed? After we put out the initial 3.0 draft of the model within the next two months, the working groups can begin to help with that. Once we have this first draft, we will send it out for informal review and get groups' input. This will help them develop their substantive proposals. We can pull pieces out and provide them in context for the groups.

The schedule will be: Informal review in January; more formal internal review in February; possible meeting again in March; then another major revision of the schema to have something in good shape by the time of IASSIST in May. The DDI meeting in Edinburgh is on Sunday and Monday, May 22 and 23, 2005. If it is possible to have an interim face-to-face meeting between now and May, perhaps we could do that in late February or early March in Europe.

Mark, I-Lin, Wendy, and Arofan will now begin work on modeling. The deadline for other assignments is two weeks from now or November 8.

DDI-SRG Consideration of Other Metadata Standards

October 24, 2004

Dublin Core

Coverage	Digital/physical objects
Domain	Library
Purpose	Bibliographic reference; resource discovery tool
Representation	XML, RDF, etc.
Main Characteristics	There is no authority control; this is just the basic elements you need to have, very generic; do we want to borrow their terms? topic and coverage very broad level; not exact subset of MARC
Versioning	Simple, qualified
Modularity	no
Extensibility	Yes, Dublin Core plus qualified Dublin Core
Overlap	Yes, but at a general level, DDI more detailed
Learn/Borrow?	We could do Citation type = Dublin Core or MARC; we need more detail in determining how various documents are linked; we can provide a downwalk not a crosswalk; optional but redundant; less of a problem this way maintaining synchronicity; how will applications deal with this? we provide guidelines; controlled vocabularies around types; mappings between types; gross discovery with Dublin Core; base format of OAI is Dublin Core; we could have embedded block that is Dublin Core; generate Dublin Core from DDI instance? Dublin Core is in wrapper; top level gets harvested; once we figure out modularization of DDI, we will know where to pull Dublin Core in; have to use their definitions if we use their names

MARC

Coverage	Catalog records
Domain	Library
Purpose	Bibiographic reference; resource description tool
Representation	Sparse XML
Main Characteristics	Standard catalog records at a high level of description
Versioning	
Modularity	
Extensibility	no
Overlap	More general resource discovery
Learn/Borrow?	Authority control

OAIS

Coverage	Digital objects
Domain	Archives
Purpose	Best practice for running a digital archive for a designated community; developed by the Consultative Committee on Space Data
Representation	N/A
Main Characteristics	SIP Submission Information Package; AIP Archival Information Package; DIP Dissemination Information Package; data objects interpreted using its representation information yields IO, information object; in our world an IO is produced by applying DDI
Versioning	no
Modularity	yes
Extensibility	
Overlap	
Learn/Borrow?	Use for archival portion of DDI lifecycle; Provenance, context, reference, and fixity are important preservation metadata to carry in the system; we don't have fixity currently but need to add this; capture processing history of a collection

OAI

Coverage	Digital instances
Domain	Library and archives
Purpose	Protocol for metadata harvesting; OAI-PMH; to promote interoperability; harvesting content only; messaging protocol
Representation	XML
Main Characteristics	Only six verbs: GetRecord, Identify, ListIdentifiers, ListMetadataFormats, ListRecords; ListdSets; Unique id that must correspond to URI; identifies an item in a repository; record is an item expressed in a single format; all formats have same id; do we want unique id not dependent on format? URI resolves to something format-bound, which is not typical; other metadata would be in same record; reserved list of values for the formats? no
Versioning	yes
Modularity	no
Extensibility	yes
Overlap	
Learn/Borrow?	Need DDI in schema format for OAI; means using URI formats as identifiers; we must support OAI

FONDS International Standard archival Description General ISADG

Coverage	Collections of records
Domain	Archives
Purpose	Reference/description purpose: To describe the whole of the records, regardless of form or medium, created and accumulated and used by a particular person, family, etc; collection of somebody's papers, etc.; how traditional archives approach material
Representation	own
Main Characteristics	6 elements: identity; context; content and structure; conditions of access and use; allied materials; notes; five possible levels
Versioning	no
Modularity	yes
Extensibility	no
Overlap	
Learn/Borrow?	This may be how we structure relationships in conceptual model; describes collections of materials; archives are concerned with relationships, unlike libraries which treat objects separately; perspective of lifecycle of archive; snapshot at current time; if you change something the FONDS will change; driven by boxes of stuff; living collection; add to the FONDS each time new file comes in; hierarchies; Not detailed enough at the lower levels; we have to extend it

EAD Encoded Archival Description

Coverage	Collections of records
Domain	Libraries and archives
Purpose	Same as above
Representation	XML
Main Characteristics	Library of Congress standard; ways of describing collections; interesting for building tools across archives;
Versioning	
Modularity	
Extensibility	
Overlap	
Learn/Borrow?	Public forms of documentation for sale; authority control

METS Metadata Encoding and Transmission Standard

Coverage	Digital objects
Domain	Libraries and archives
Purpose	METS represents set of digital objects; like a directory structure

Representation	XML
Main Characteristics	When render a METS instance, you can rely on that structure to navigate through content; insufficient in describing richness between nodes; section for describing behaviors like Web services that can act on it; METS is like a FONDS but without fixed levels; Tools like Fedora, dspace going METS; higher up wrapper; insert more complex DDI content into it
Versioning	
Modularity	yes
Extensibility	
Overlap	
Learn/Borrow?	Structural map; look at linking; wrapper; behaviors; modularity

XMP

Coverage	Digital metadata
Domain	
Purpose	Adobes's extensible metadata platform; platform to attach metadata records to digital objects (embed)
Representation	RDF/XML/XMP
Main Characteristics	Picture created by digital camera; xmp travels with the photo; versioning added; open source; public domain; starts with a data model based on RDF information model; storage and packaging model; set of schemas defining metadata properties; internal properties only read by software; external by humans
Versioning	
Modularity	yes
Extensibility	Yes; xmp basic schema (basic descriptive information extending Dublin Core); rights schema, etc.
Overlap	Only when it comes to general descriptive Dublin Core-like information
Learn/Borrow?	Embedded, traveling metadata; modularity based on namespaces; extensibility; building model on a simple information model like RDF; embed DDI with SAS/SPSS would be equivalent; we are far from this right now

FGDC

Coverage	Geographic space
Domain	Geography
Purpose	Describe points, lines, polygons, circles, arcs;
Representation	

Main Characteristics	
Versioning	
Modularity	
Extensibility	
Overlap	
Learn/Borrow?	Geographers search by coordinates so we added bounding box to the DDI; we also included bounding polygon with a minimum of four points; our definitions differ; when geographers say smallest piece, they mean that they can create all levels in between; broad geographic element and smallest piece of geography are not sufficient; no controlled vocabulary; in redoing DDI we need to be conscious that a geographer expects to identify a geographic entity; how they understand definitions may be different; our situation now is that we have element names from FGDC and definitions from Dublin Core; Should DDI be able to incorporate another standard? Harmonize DDI tags with FGDC? Concept of geographic time missing; get another group to take a hard look at this; proper requirement analysis needed; we don't want to move into GIS territory; There are two problems we want to solve in the DDI: (1) ability to search by geography for statistics (locate statistical data in space) and (2) put data onto a map and link to maps; link to external standards; get working group on this; deliverable from group: uses of information; pieces of information needed; relationships to other information in DDI; what are geographic models out there? FGDC compliant with ISO; Open GIS and GML are simpler; what we have now is human-readable

TEI

Coverage	Text markup
Domain	Humanities
Purpose	Text markup and formatting
Representation	XML
Main Characteristics	
Versioning	yes
Modularity	
Extensibility	
Overlap	Formatting elements used now for large text areas
Learn/Borrow?	The DDI should dispose of the TEI, but white space formatting not adequate; xhtml has similar structure and seems the best option; restrict allowed set to structure elements that require formatting; use generic class attribute to link to a stylesheet; modular; subset xhtml namespace; distinguish at the containing level

Triple-S

Coverage	Survey data
Domain	Market research
Purpose	Interchange standard for market research data
Representation	Pre-XML
Main Characteristics	Minimalistic DDI with major focus on the question/variable level, although there is study level information like title, provider, producer, etc.; doesn't support multifile studies, aggregate data, hierarchical dimensions; supports multiresponse questions better than DDI; Language differentiation simpler, but better than the DDI; 60 software packages have Triple-S support; interchange standard for market research; includes both data and fixed format ASCII file
Versioning	no
Modularity	no
Extensibility	no
Overlap	90% of elements map directly
Learn/Borrow?	Simplicity; Language differentiation mechanism; made distinction between elements that were human readable and what would appear in different languages and what wouldn't; how can we leverage vendor support they have? provide crosswalk; third party software integrates Triple S with SPSS MR; simple core with optional modules

SDMX

Coverage	Aggregate data, almost exclusively time series
Domain	International statistical agencies
Purpose	Data exchange format with the metadata needed;
Representation	XML
Main Characteristics	SDMX drew limit of metadata at how much you need to understand the data; Version 2 will have pure metadata reporting; level that is above to represent that dimensions are equal; Insists on clean cubes; describes cube and will include hierarchical code lists in next version; has three schemas for timeseries, one for nontimeseries; exchange rates by country and by time; contains concepts; has standard way of expressing time (ISO 8601); concept has to be formal with a definition; frequency and periodicity; technical specs; semantic harmonization separate; ISO11179 style definitions for concepts; expressed using key families; assumption of clean cube is big assumption; hierarchical dimensions not in yet; typically, one can link multiple cubes; embed into another data cube; when confronted with sparsely populated cubes, produces logical values; uses namespaces for modularity and extensions; mapping that lets you see how comparable this is in registry; similar to comparability approach; published lots of different cubes not really related; like to have ability to say this time dimension repeated across cubes; descriptive richness not tied to all cubes; document trends without the data being there; if identify trends without the data accumulated, up to us to go down to the 11179 level

Versioning	yes
Modularity	yes
Extensibility	Yes; uses namespaces to accomplish this; early-bound namespace
Overlap	Timeseries data cubes
Learn/Borrow?	Encourage aggregate group to look at this; possible to have another level of abstraction to describe family of variables and put into DDI package; when documenting each instance, instead of repeating, just point to generic description; good approach; in general could be used to describe common information; generic dimension; way data warehouse treats cubes; make another instance as a collection of collections; base document and then a document that talks about more than one instance; standard ways of expressing time periods; FAME has good code list; timeseries database; there is a ISO standard that deals with numbering weeks, which is important when dealing with periodicity; Venetian blind version of schema; URIs for identifiers

ISO 11179

Coverage	Concepts, data elements
Domain	Statistical agencies/data producers
Purpose	Provide registry of items for use in survey questionnaires
Representation	XML
Main Characteristics	Data dictionary model; if you are doing a data vocabulary, do it like this; naming is object oriented; object property representation hierarchy; nobody uses that; naming rules not worth pain involved; guidelines for defining things; this is useful; managing concepts and data elements
Versioning	Five parts, until latest, unreleased one is blowing away part numbering
Modularity	yes
Extensibility	
Overlap	Concepts, data elements
Learn/Borrow?	Guidelines for defining things; CMR Census and StatCan use this; conceptual modeling and retrofitting them to work with metadata; becoming important in data world at large; CMR model complex; to be compliant is good politically; what type of compliance? Core concept is appealing; different value sets to the same concept; key to solving comparability issues; ISO markup means you can discover comparable items. Alignment is easy but not using their model; we could say we are compliant now; word our definitions for version 3; stay with these two levels of harmonization

MetaDater

Coverage	Survey data
Domain	European social science data archives, comparative data
Purpose	Involve PIs in documenting their data and store all metadata in a single place

Representation	UML use cases
Main Characteristics	Two software packages that interact and an underlying database; MD-PRO provider, and MD-COLL collector; tool for documenting while doing survey; then tool for archives; need conceptual model, which is being developed; overall model including all possible information on a survey; from the customer relationship level, user level, whom we give data to, variable level, etc.; end up with entity relationship model; still working on this; underlying structure like OAIS model
Versioning	
Modularity	yes
Extensibility	
Overlap	Lifecycle approach; survey done, analyzed, publication archive, dissemination; data producers not good at documenting; getting producers to fill in documentation information not ex-post, as happens now in archives; comparative data
Learn/Borrow?	Can they share their draft conceptual model? We will try to get a copy to inform our efforts; our data models need to align with each other

CWM Common Warehouse Metamodel

Coverage	Digital metadata
Domain	Business
Purpose	Major effort by Object Management Group (OMG) to enable easy interchange of warehouse and business intelligence metadata between warehouse tools, platforms, and metadata repositories.
Representation	UML
Main Characteristics	Built up using OMG tools; Meta Object Facility (MOF); XMI XML Metadata Interchange; Object Model is at bottom; a subset of UML itself; cubes, XML, record data, relational data object model; transformation on top of that along with OLAP; queries and extracts; data mining; information visualization; business nomenclature
Versioning	
Modularity	Highly modular; several layers of submodules; plug and play as you build
Extensibility	
Overlap	Yes, describing part of their package describes some of our data
Learn/Borrow?	This was cutting edge when Corba was in operation; this was developed in 1997; supported by Oracle, SAS, IBM, etc.

IQML

Coverage	Electronic questionnaires, Web surveys
Domain	European project

Purpose	Project to develop language combining survey responses and questions; self-configuring; describe survey questionnaires; especially Web based surveys
Representation	XML
Main Characteristics	Flexibility
Versioning	
Modularity	
Extensibility	
Overlap	
Learn/Borrow?	Project no longer in operation; need to talk to others about this