Overview of the DDI Version 3.0 Conceptual Model

This is an early working draft 4 created by the Structural Reform Group

I. Introduction

This document provides an overview of the Data Documentation Initiative (DDI) Version 3.0 Conceptual Model. Unlike preceding versions, the DDI standard will consist of two parts – the conceptual model, and the XML Schemas and DTDs which are derived from it. This is a common approach to the standardization of XML vocabularies, and one which provides many benefits to users: the vocabulary itself becomes more consistent and comprehensible, and the conceptual model can prove a valuable asset to developers of applications which need to support the standard, as many tools now allow for XML binding directly from a model expressed in the Universal Modeling Language (UML) or its derivatives.

DDI Version 3.0 represents a major change from preceding versions in another fashion: the scope has increased. Historically, DDI was focused on data archiving, and while this still remains a major focus, in Version 3.0 all aspects of the data life cycle will now be supported. Thus, as a data collection process proceeds, from conception to reuse, the growing set of metadata describing this activity can be collected and expressed in DDI.

This shift in scope has many repercussions in the overall design of the DDI. It means that instances will be larger, to accommodate the expanded set of metadata. It also means that the simple case, where a single data file is described, no longer universally applies. Data from “studies” may be found in several files in a more flexible fashion than in preceding versions of the DDI. (The distinction between the “simple” case which parallels the existing use of DDI and the more “complex” cases, involving several studies, is detailed below.)

Supporting the life cycle also has other impacts: the relationships between a study and those on which it is based may also need to be recorded, and thus, groups of studies need to be described, such as a series of longitudinal studies. A natural result of this change is the ability to express comparability of studies, particularly those which are designed to be compared.

The metadata describing the life cycle is not complete without capturing information about the survey itself in a richer form than an image of a paper collection instrument. Many systems today allow for the re-use of questions, and thus instrument metadata are a necessary part of life cycle support.

Some other changes will be seen in the DDI Version 3.0 as well: a subset of HTML tagging will optionally be supported in some of the fields where longer, human-readable text is found. Also, the handling of reusable classes, such as notes and citations has been made more uniform, increasing both the consistency of the structure and the flexibility of references to external and internal materials. The importance of other metadata standards is also

recognized in this design, with the stated intent of alignment or use of several other initiatives' products.

While the changes in DDI Version 3.0 are ambitious in scope, one of the major design goals is to avoid making migration from Version 2.* any more arduous than necessary. The simple use of DDI for archival purposes is not radically different between versions, and mappings of all currently-used fields will be provided, as will some simple free tools for helping users.

Some of the biggest changes are the result of advances in XML technology. Because the use of W3C XML Schema (XSD) has become mainstream, the DDI DTD will no longer be the canonical expression of the standard. Instead, it will be a sister-product of the Schema, which – while it also describes XML instances – will express more of the validation parameters than are possible with a DTD.

The use of XML namespaces is another typical XML practice which DDI Version 3.0 will introduce. This allows the now-expanded vocabulary to be modularized, making it more manageable and maintainable over the long run.

It should be stated that DDI Version 3.0 intends to increase the degree to which the metadata it contains is sufficient to support computer processing – that is, it will go beyond being "human readable", and move toward the goal of being "machine-actionable". This is a long-term goal, and will not be taken too far in the early 3.* versions, but it is very much in keeping with the overall use of XML-based technologies now current, such as Web services.

II. Definition of Terms

The following section defines a few of the important terms for understanding this document. Many of these terms have a variety of meanings, so they are defined here in the narrow sense in which they are used throughout the DDI Version 3.0 Model. Please note that this list is being compiled as the work is being done – the list of terms is in no particular order, although it will ultimately be replaced with a more organized glossary.

Raw Data

- Literally what is collected in its collected form
- No recodes or constructed variables have been created, data have simply been "captured"

Microdata

- Individual level data (whether that individual is a person, place or thing)

- Variables providing identification of the record or its relationship to another record may have been added
- Recodes or constructed variables may have been added
- Raw data elements may have been omitted

Aggregate Data

- Data that are the result of summarizing raw or microdata to reflect data for a larger group
- Data are commonly aggregated from individual data records to a summarized geographic area
- Data can also be aggregated by class, such as all females or all males

Study

- A collection of data files resulting from the intentional collection of data through solicitation, observation, or gathering from secondary sources for a purpose described in the Study Concept; using the data collection instruments and methodology described in the Collection Process; and expressed as data files with logical structures relating back to the data collection instrument

Concept

- A definable idea or characteristic of the unit of analysis

Representation

- What is created when you measure a concept

Variable

- A specific expression of the representation of a concept

Single Conceptual Model

- The DDI Version 3.0 is based on a single conceptual model that describes what the DDI covers and how it organizes that information intellectually.

Technical Implementation

- The DDI Conceptual Model can be expressed through a number of Technical Implementations. Technical Implementations include but are not limited to XML DTDs and XML Schema. A typical example would be a database representation of the DDI documentation.

File

- A single computer file. A data set can be made up of multiple files.

Instance

- The term instance is a technical term meaning “XML instance” as defined in the XML specification. It is the complete XML document with all of its information.

Lower in the Model

- The model can be thought of as a multi-branched hierarchy. Some modules are siblings with or without specific ordinal relationships, some are parents of other modules, and there is similar relational order within the modules. As you move from describing the broad concept of the study, through the data collection process, to the description of the logical structure of the data, and finally to the physical location of a specific data item in a specific record, you are moving lower in the model. “Lower in the model” implies inheritance from information “higher in the model”.

Design Rules

- In addition to the Conceptual Model, development of the DDI will be bound by a set of design rules. Design rules govern naming conventions, inheritance structures, and reference direction as well as a range of other design parameters. The purpose of design rules is to provide consistency in the structure. Design rules can change, but change should be the result of a deliberate decision rather than accident.

Functional Type

- The idea of “functional type” has been introduced as a means of characterizing the function and/or use of specific types of metadata within the DDI Version 3.0 modules.

Class

- Classes are the most important concept in object-oriented software development, and in UML as well. Classes hold operations and attributes and are related to other classes via association or inheritance relations. A class has a few properties of its own, such as name, stereotype and visibility, but the more important aspect is its relation to other classes. One can think of classes as those things in the model which will become elements in the XML serialization.

Wrapper

- This is the term used for the construct in the model which will correspond with the top-level element in the XML document instance. The version 3.0 DDI describes instances which can contain a wide variety of metadata - there is a single element which is always used to contain whatever the DDI instance holds. This includes some administrative information about the XML document.

Reusable Classes

- Reusable classes are those that can and do show up in multiple modules; Other Material, Universe, Citation, Notes, etc.

Module

- A module is a collection of one or more classes. The DDI Version 3.0 is a modular structure (made up of modules). Modules are similar to the upper levels of DDI Version 2.0 (Document Description, Study Description, File Description, Data Description, and Other Materials).

Basic vs. Specialized Modules

- A basic module should be able to be expressed in the widest possible number of Technical Implementations. A specialized module allows specific applications or specialized functions to dictate its features. Specialized modules can be easily identified and ignored by systems that were not designed to handle them. For example, a basic Physical Data Structure model would be able to describe fixed format and delimited format file structures. A specialized Physical Data Structure could describe the call functions for a proprietary data structure. This allows the users of proprietary software to create a specialized module that will work directly with their software.

Study Unit

- This is the full unit level of metadata captured by DDI at the study level. A study can contain one or more study units. A study containing a single study unit is a simple case and is reflected by the structure of DDI Version 2.0. A study with multiple study units is considered complex and can be described using the Version 3.0 Group module to define the relationship between the study units within a complex study. See the flowchart for determining whether the study contains a single study unit (simple) or multiple study units (complex).

Data Set

- A data set is the data files described by the Logical Data Structure. The data can be stored in one or more data files.

Human Readable

- These refer to sets of information, such as an abstract, that is intended to be read by the user. While it may be searchable by a computer (matching words or strings) it is not intended to provide a consistently structured set of instructions to a computer program.

Machine Actionable

- This refers to information that is relayed in a consistently structured manner that can be used by programmers to instruct their systems in navigating a DDI XML instance.

Persistent vs. Dynamic Information/Material

- As a study proceeds through its life cycle certain pieces of information are persistent and others are dynamic. Persistent information doesn't change once it is "published". For example, the date some particular data was collected is not going to change further down the life cycle nor is the identification of the collecting agency or the identifier of the study. However, if Archive A and Archive B both hold copies of this study, the local holdings information and access information will likely change. This information is dynamic. This is not to say the metadata will not be enhanced in various ways through its life cycle, just that some information is expected to remain stable and some is expected to change.

Local Overrides

- In the DDI Version 3.0 grouping structure, it is possible to specify metadata in the top portion of a hierarchy – that is, for any group – and have it be shared by all members of that group. Local overrides provide a mechanism where a member of the group which does not share a specific piece of metadata may state the correct value, to be used in place of the shared one.

- An example of this would be as follows: if all the StudyUnits in a group used a single set of survey questions for determining gender ("Please specify your gender", with answers "Male" and "Female") except for a single StudyUnit in the group, which asked "Are you male?" with answers of "Yes", "No," and "Unsure", then the question could be

included in the appropriate module at the Group level. For the one StudyUnit which did not use the first question, the alternative would be provided for that StudyUnit as a local override – that is, as a replacement for what was inherited from the metadata at the Group level.

III. Life Cycle Metadata and Implications for the DDI

While the original DDI took its model from the codebook, it was clear early on that many were expanding that concept to mean something much broader and perhaps more complex than a traditional hardcopy codebook. With Version 3.0, we now have the capability to document the rich complexity of social science data across its life course.

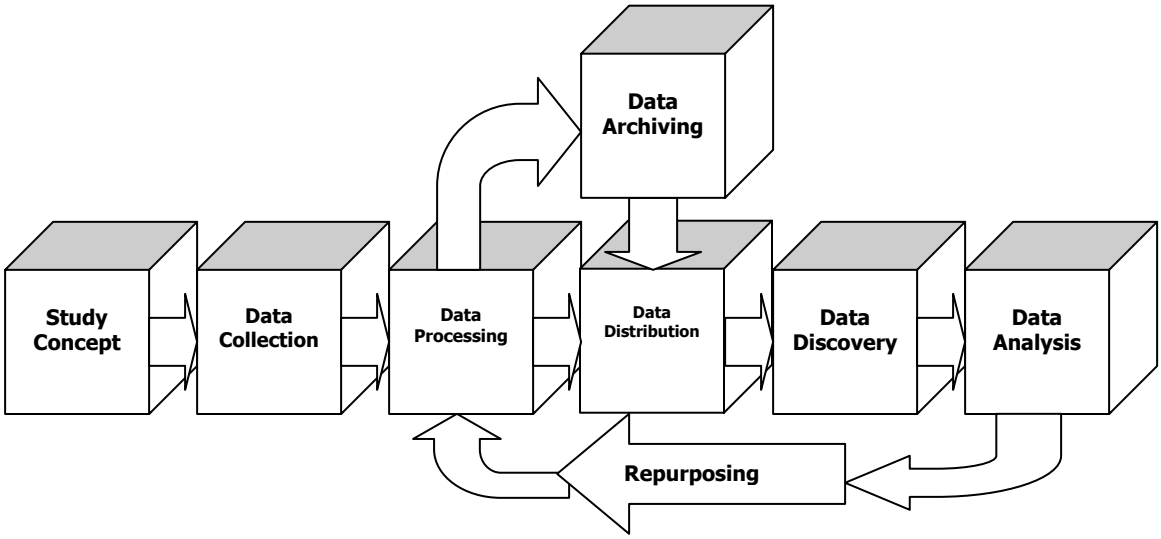


Figure: Combined Life Cycle Model

A. DDI Instances and the Life Cycle

Historically, there has been no concept of a DDI instance existing as a study was designed, administered, and then archived. As we see in the figure above, there are now several steps to the life cycle which could be documented using DDI instances. These instances could be versions of one another: that is, if they are documenting the same study or group of studies, and they are created by the same agency, then they would represent the development of a set of metadata over time.

For DDI Version 3.0, it is conceivable that the conceptual design of a study would be marked up in DDI, and that as the study went through the life cycle, the DDI instance documenting it would be updated in a sequence of versions: typically, one for each stage of the life cycle. This requirement leads to the modular nature

of the DDI Version 3.0 design: as metadata are added, additional modules are added, keeping the development of a DDI instance over time more comprehensible, and making it easier to find and process the various parts of the metadata which are of interest.

The nature of these changes will largely be additive – that is, the additional metadata relating to a particular stage in the life cycle will be added to an existing DDI instance to create a new version. Versioning changes are not limited to this type of additive change, however, as the instance must document the real-world metadata.

Note that what is versioned in DDI Version 3.0 is the set of metadata about a particular study or group of studies, from a particular agency. It is *not* the DDI XML instance. There is no assumption that DDI instances will be maintained: they can – and often are – used as a transient mechanism for the exchange of metadata which is persisted inside some other application or system.

B. Repurposing of Data

The Combined Life Cycle Model incorporates either direct dissemination to users or dissemination through data archives and recognizes that data can be reprocessed at later points in its life cycle, creating an iterative process. Typically, this occurs when existing data are re-used as part of an unanticipated, later study. This means that the life cycle is no longer linear but has become circular. In this model, *Repurposing* follows *Data Analysis* and therefore can't feed back in time. One way to address this is that each circular path is described by a new DDI instance.

We viewed *Repurposing* as being a secondary use of the data from a study. While multiple products could be planned for in the original conceptualization, collection, and processing of the data, *Repurposing* reflected a new conceptual framework. For example, this might be a streamlined instructional data set, a specific sampling and restructuring of the data, or combining data from multiple sources to create a new data set (either physically or virtually). The implications of this view include the need for defining the relationships between data products conceived of during the conception process (such as the multiple products of the United States Decennial Census) as well as the ability to define both primary and secondary data sources within the *Data Collection* phase.

The movement to a modular design for the model has been developing over time and is not a radical change in direction as much as it is recognition of the emerging consensus. It is needed to provide the flexibility for dealing with specialized data files and data sets as well as the variety of technical environments within which we currently work or are in the process of developing.

C. Two Uses of the DDI: "Simple" Cases versus "Complex" Cases

One aspect of DDI Version 3.0 which follows from the support of the whole life cycle is the introduction of groups of studies as the subject for metadata documentation. Longitudinal studies are a good example of this. A longitudinal study is a study repeated at specific points in time, and thus represents a group of related studies. These need to be documented as a group – a longitudinal study involves repurposing of many aspects of the initial study, and also needs to document the relationship that exists between each of its component studies.

This and similar cases were not supported by design in the original DDI, which was intended to document individual studies, and only supported their description from an archival perspective; that is, after the fact.

To avoid making all uses of DDI complex as a result of this requirement, there are two proposed uses of the standard in Version 3.0. These are termed the "simple" and "complex" cases. The "simple" case is intended to represent a usage of the DDI similar to what was done in early versions: to document a single study. The simple case is modular, and does support the stages of the full life cycle, but it does not involve groups of studies.

The "complex" case involves groups of studies which are being compared, or a series or collection of studies which are related in some way. It is important to know which case a potential use of the DDI involves, because the 'complex" case uses features of the DDI which are potentially more difficult to understand and implement. These features are the grouping and comparison features. This design intends to allow those who need to document the "simple" case to avoid having to understand or support the full complexity of DDI Version 3.0.

1. Simple Case

A simple case is a study with a single conceptual model, with a single integrated instrument of one or more parts, that is administered at one or more occasions resulting in a data set with a persistent logical structure. This logical structure may be represented by one or more physical structures that are linked to each other with predefined keys. A single physical structure may be represented by one or more physical instances whose record layout matches the physical structure but may contain differing sets of records.

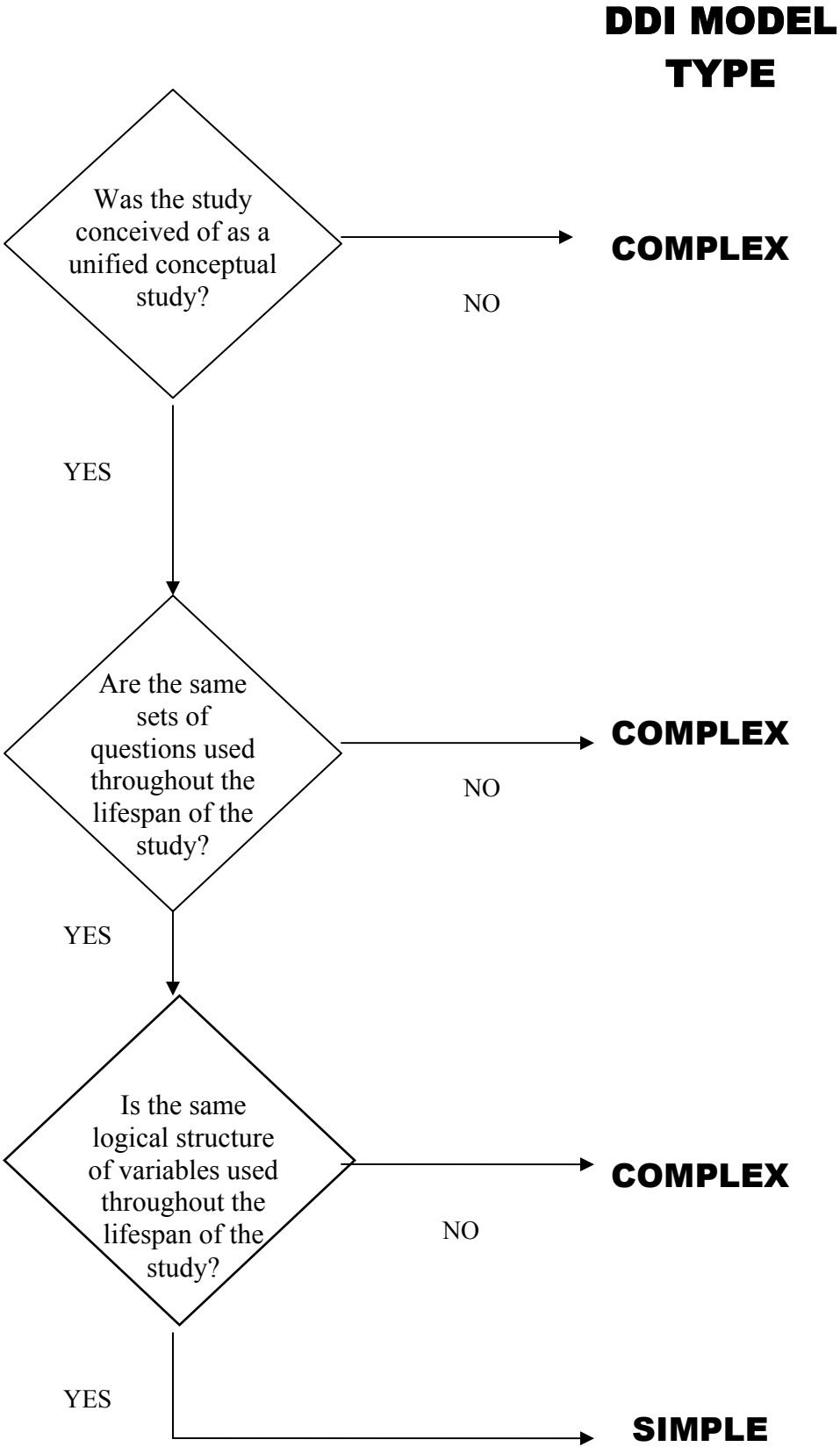
The key criteria are:

- Single conceptual model
- Single instrument made up of one or more parts (ex. employer survey, worker survey)
- Single logical data structure

If either the instrument content (questions) or the logical data structure (variables) change over the lifetime of the study, then it becomes a complex instance requiring the use of a grouping module to define the relationships between the data sets.

In the case that the creator of the XML does not choose to use any grouping module (if these modules are not supported by local systems), then a second XML instance must be created and any information on the relationship between the two instances will be restricted to human actionable sections of the metadata. Machine actionable relational information will be lost.

The following flowchart illustrates the process of determining whether a given subject of documentation should be considered a “simple” case or a “complex” case.


DDI MODEL TYPE
Was the study conceived of as a unified conceptual study?
NO
COMPLEX
YES
Are the same sets of questions used throughout the lifespan of the study?
NO
COMPLEX
YES
Is the same logical structure of variables used throughout the lifespan of the study?
NO
COMPLEX
YES
SIMPLE

IV. Migration and Modularization Design

A. Migration of 2.0 Elements

All elements and attributes in 2.0 are currently represented in 3.0. Due to options for applying a small number of elements in 2.0, some hand editing or review of contents may be required to accurately migrate them to 3.0. The greatest change will be separating information currently in section 4.0 into questionnaire, logical descriptions of variables and related items, and physical storage locations. Software will be developed by DDI to facilitate this migration.

Because DDI was originally intended to support what is now termed the “simple” case, that aspect of the migration from Version 2.* to 3.0 should be more fully automatable. Thus, if you have single-document DDI instances, these should migrate in a fairly straightforward fashion to “simple” DDI Version 3.0 instances. In cases where DDI Version 2.* has been used to document more than a single study, the migration may become more complex, as a set of study documentation (Version 3.0 instances) will need to be created from the single source file.

The biggest change to DDI instances in Version 3.0 will be the explicit and required use of XML namespaces. It is intended that each module described below will exist in its own namespace, and these will be reflected in one of the allowed ways in the XML files themselves. Use of XML namespaces is both necessary to allow DDI to use other standard structures, and for easy maintenance of the DDI standard XML DTDs and Schemas.

XML namespaces use a prefix to identify the module from which an element description is taken. Thus, if the data collection module has its own XML namespace, it could be given the prefix “datac”. A “var” tag would look like:

```
<datac:var>…</datac:var>
```

In DDI Version 2.0, there was a single, implicit namespace. Now, each module will have a namespace, and they will be made explicit.

B. Modular Design

The design of the DDI Version 3.0 allows greater flexibility in combining various modules within a single wrapper to describe a single data file, a related group of

data files, or a related group of studies. It also allows software developers or users to select which modules of information they can handle and to ignore modules outside of their capabilities.

1. Goals for Modular Design

- To organize the modules so that they accurately record information about data and the data creation process AND contain the information on structures and relationships necessary for data discovery, extraction and manipulation
- To have basic modules that will work in all technical implementations (specialized modules may not work in all technical implementations)
- To provide specialized modules for special types of data or storage formats so that all elements in the DDI are used in a consistent way
- To organize the elements within modules so that if your system cannot handle a specific module the other modules will still work. (An example of this would be an application which is not designed to process Instrument metadata – because the Instrument metadata is in its own module, the application knows to ignore that part of the DDI instance.)

2. Design Rules

The design goals above give rise to a set of rules that guide the creation of the model:

- Persistent sections should be separate from dynamic information
- Information modules should follow through the various life cycle paths
- Information used for discovery should be in non-specialized modules
- Separation of dynamic materials and non-dynamic materials: What parts change when a data file moves from one “home” to another, or changes something like its physical storage structure? Theoretically those pieces should be modules that can be “swapped” out.
- Information discovery perspective: What information is needed at different levels of discovery/extraction/manipulation and what search engines would be accessing the information at each level? It is beneficial to keep information used by non-social science data specific discovery systems together and/or uniformly accessible.

V. Simple Instances

A. Mapping from Version 2. to Version 3.0*

In the "simple" case, there will be a set of modules which correspond roughly to the DDI Version 2.* sections. The mapping for these is as follows:

Version 2	Description	Version 3
1.0	Document Description: Citation of the XML Instance / Content Citation of the Source documents	Wrapper / Archive
2.0	Study Description	
2.1-2.2, 2.4-2.5	Study Description, Citation, Universe, Other Materials, Note	Concept
2.3	Methodology	Data Collection Process
3.0	File Description	Physical Data Structure / Physical Data Instance
4.0	Data Description	
4.1, 4.2, 4.4	Variable Groups, nCube Groups, nCubes	Logical Data Structure
4.2	Variables: 1) Question 2) Location 3) Summary Statistics 4) Everything else	1) Data Collection Process 2) Physical Data Structure 3) Physical Instance 4) Logical Data Structure
5.0	Other Material	Other material class of the relevant module

Notes:

1.0 The Archive module will hold all the information specific to the archive including holdings information and file locations. The wrapper and its various classes (Other Materials, Notes, Universe, and Citation) will hold the remaining material.

2.0 The materials currently in the Study Description are split between the Concept Module and the Data Collection Process Module roughly along the lines indicated in the table.

3.0 The Physical Data Structure Module contains the detailed record structure information and location information while the Physical Instance Module contains information on the gross file structure.

4.0 Most of the material in Data Description will move to the Logical Data Structure Module with the exception of the first three items listed under

Variable. Question information will become part of the Instrument section of the Data Collection Process, Location becomes part of the Physical Data Structure (similar to the current location map section), and summary statistics will move to the Physical Instance module.

B. Functional Types

The modules themselves are organized into “functional types”, which is a short-hand functional description, characterizing their contents. Note that there is a distinction between human-readable and machine-actionable metadata, which has become an important distinction in the types of metadata documented within the DDI.

Functional types are **not** structural in nature – all they do is describe the function of a particular set of metadata. They are presented here only for the purposes of clarifying the use of modules (which *are* structural) in the table below.

Functional Type 1:

Collection relationships of Functional Type 1 are provided by the archive or holder of the metadata. These include archive-specific information and non-technical grouping information such as common topic or common producer, organization, funding source or principal investigator. Grouping at this level doesn't carry technical implications for how the data are processed, but may affect the upper level discovery search process within the archive.

Functional Type 2:

Collection relationships of Functional Type 2 have technical implications for how to handle the data within the group. The Group Matrix identifiers are used in describing grouping of Functional Type 2 in order to provide specific information to programming applications. Examples of this are time-series, longitudinal studies, repeated surveys, etc.

Functional Type 3:

Collection relationships of Functional Type 3 allow for grouping multiple data collection actions and the use of multiple data collection instruments during a defined data collection activity. For example, data may be collected from a group of students, their parents, and their school within a specific time period using three separate collection tools.

Functional Type 4:

Collection relationships of Functional Type 4 relate one or more Logical Data Products created from the same data collection process. For example, a microdata file and an aggregate summary file, or a Household file, Family file, and Person file. Each logical data product can be represented by one or more physical storage structures and one or more physical instances of the full set of records or specific subsets of records.

C. Module Descriptions

The following table describes the various modules used in the "simple" case, as well as the Grouping modules. It also lists out a set of reusable classes which are common components of many modules.

Note that all of the modules listed below are used in the simple case, with the exception of Informal Group and Formal Group. The modules themselves are not what adds complexity to the "complex" case – it is the interaction of modules at different levels within a structural "grouping" hierarchy which makes the processing of DDI instances more complex.

MODULE	Description	Relationships
Wrapper	Contains top level Citation and Universe information; Provides structural map for modules included in the instance	Contains all other modules
Informal Group	Describes the collection of two or more data Functional Type 2 or 3 module sets based on topical or other non-technical basis. Imposed by the archive for purposes of internal management or local organization of materials.	
Archive	Describes all archive specific information. Originally a basic module for archive identification and access restrictions. May be extended locally to cover processing management.	
Formal Group	Collection of Functional Type 3 modules exhibiting specified	Uses technical specification matrix to identify relationship

	relationships that have technical processing implications for accessing and analyzing the data	types along 6 parameters [see Grouping, below]
Concept	Defines purpose of the data collection and resulting data products	Parent to one or more Data Collection modules
Data Collection	Describes data collection process through cleaning and data set production	Contains one or more Instrument modules Parent of one or more Logical Data Products
Instrument	Data collection instrument [currently contains Ver 2.0 question elements—needs expansion to full generic instrument description]	
Logical Data Product	Describes the logical content of the data product	Contains Variable module and nCube module if applicable to data type; describes links between 2 or more logical data products or records within a hierarchical or relational data structure
Variables	Describes the variable concept and structure; describes variable groups	LINKS from variable to question(s) from which variable was obtained
nCubes	Describes the construction of nCubes from variables; describes nCube groups	LINKS to variable(s) used for dimensions of nCube
Physical Data Product	Describes the structure of a data store in terms of a record structure (1, 2 or 3 dimensional storage structures)	LINKS from data item location information to the variable description either by pointing directly to the variable or through the nCube coordinate structure
Physical Data Instance	Describes the gross file structure; allows for subsets of full records (limited universe would be noted in Universe Class elements)	Child of Physical Data Product
REUSABLE CLASSES:		
File/Section ID	Unique identification of module including type and version	
Citation	Bibliographic citation material only (from Dublin Core)	

Universe	Universe definition (topic/time/geography)	
Other Material	References to material outside of xml instance; citation; material type identification	External Link: URI Internal Link: pointed to by appropriate element within module
Notes	Notation type and contents of note	Internal Link: pointed to by appropriate element within module

VI. Multiple Files

A major difference between Version 3.0 and 2.0 is the ability to describe multiple files within a single DDI instance if needed. Common use of this feature includes:

- One study in which the data are stored in two different physical formats -- All the information except for physical storage description can be stated once and then a separate module for each physical store is created and linked to the same logical description of the variable contents.
- One study where the description of the logical file structure and the physical file structure remains consistent, but the physical file has been separated into multiple parts to aid processing (for example: A Canadian Census summary data file split into a separate file for each province).
- A time series with multiple files derived from a common questionnaire.

Note that there is no necessary relationship between “simple” and “complex” uses of the DDI and the number of files. A ‘simple” DDI instance could describe multiple files.

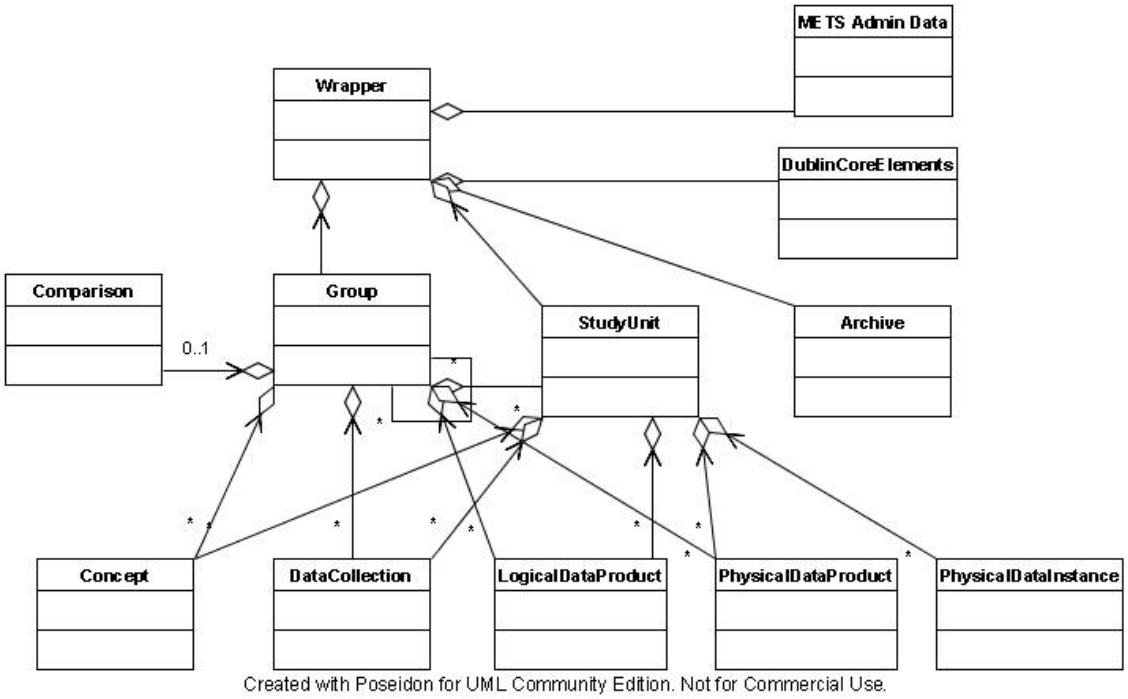
VII. Comparison

[The comparison module is not yet fully described. We have discussed the grouping and modularization aspects of Version 3.0 presented here, and it is felt that there is a sufficient degree of alignment between the thinking in the Comparative Data Group and the Structural Reform Group. This will be a topic for further work.

To date, comparisons will focus primarily on data which is designed for comparison, as opposed to ad-hoc comparisons. However, it has also been contemplated that comparisons could be conducted after-the-fact on data sets which happen to be similar, even if this might involve the creation of harmonized “virtual” data sets. The extent to which DDI Version 3.0 should support this type of comparison has yet to be determined.]

VIII. Grouping

Below is a UML class diagram showing the DDI Version 3.0 model for the grouping structure of “complex” cases. (Note that the entire model will be rendered in UML as well as XML Schema and DTDs, along with other documentation formats, but that this work is not yet complete.)



A. Overall Structure

The DDI model for Version 3.0 has extended the implicit model in Version 2.0 to allow for the grouping of an individual Item’s metadata. This grouping serves several functions:

1. To allow for informal packaging of a set of related Items’ metadata on an ad-hoc basis. For example the grouping of a collection of studies based on a common funding source.

2. To allow for informal grouping of Items for ad-hoc comparative purposes. This will be described in greater detail in the Comparison section of this document.

3. To group a set of Items formally, based on common, machine-actionable parameters. These parameters include time, instrument, panel, geography and data sets. An example of this type of group is Items from a longitudinal study or other comparable by design Item.

4. To allow for the inheritance of common characteristics of studies up the metadata structure hierarchy. This allows the simplification of DDI instances, by allowing the common meta-data to be stated only once at the upper level of the grouping hierarchy. Note that this inheritance does not apply to the parameters of formally grouped Items.

B. Classes

To perform the above functions, the following classes are implemented (taken from the Modules listed above):

1. Wrapper

A Wrapper is the top-level class, which carries information about the DDI instance. This is the top level element in the instance.

2. Group

A Group is a class for both formal and informal grouping of StudyUnits or other Groups. This class contains a set of required properties (time, instrument, panel, geography, and data sets), which identify the relationship, if any, between a Group's child StudyUnits or Groups. Each of these properties contains a single value that identifies the nature of the relation. For groups with no formal relationships, each of the properties is assigned a value of "*x*0" (where *x* varies by the property). A set of tables and flow charts to determine the values of these five properties can be found in the Group Properties section of this document. In addition to these properties, this class also contains common metadata and information about comparisons and post-hoc variables. Metadata are inherited down the hierarchy of the grouping, with the ability for children to override values locally. Inclusion of children to the Group may be by reference or by direct inclusion (also referred to as nesting).

3. StudyUnit

A StudyUnit is a study with a single conceptual structure on which all the lower-level modules depend. These modules include Data Collection, Instrument, Logical Data Product, Physical Data Product, and Physical Data Instance. This corresponds to a single, "simple" instance of the DDI.

4. Study Conceptual Classes

Groups and StudyUnits both contain the cluster of modules which describe a study (or collection of studies) and its data. These classes include Concept, Data Collection, Logical Data Product, and Physical Data Product. These classes at any level, always inherit from their ancestors' classes, but can provide local overrides.

For example, if a StudyUnit is contained in a Group, the Data Collection class of the StudyUnit inherits from the Data Collection class from the Group. The Data Collection class of the Group may contain a set of basic questions. The Data Collection class of the StudyUnit would inherit these questions from the Group, but would also be allowed to provide additional questions. In addition to the classes mentioned above, the StudyUnit also contains a Physical Data Instance.

5. Comparison

Groups can also contain the Comparison class. The Comparison class contains information about the comparability of the children Groups and StudyUnits contained in the parent class. This is the module where "virtual" post-hoc variables and concepts could be described. Each comparison must contain a reference to the concepts and variables of the Groups/StudyUnits it compares, using the external key mechanism.

C. Examples

The following section provides samples showing the grouping of studies using formal and informal Groups and a combination of both. Note that the XML structures used in these examples are for demonstration purposes only, and do not necessarily represent the actual final structure. You may wish to refer to the description of grouping properties in the section below for a more complete understanding of the examples given here.

1. Informal Group

This example shows a group of StudyUnits sharing common Data Collection information - perhaps common collector – for instance, Health Canada:

```
<Group id ="A" time="T0" instrument="I0" panel="P0" geography="G0"
datasets="D0">
       <DataCollection>
              <CollectionEvent>CommonCollector</CollectionEvent>
       </DataCollection>
       <StudyUnit id="1">
              <DataCollection>
                     <Instrument>INST-A</Instrument>
              </DataCollection>
```

```
            <LogicalDataProduct>LDP-B</LogicalDataProduct>
            <PhysicalDataProduct>PDP-C</PhysicalDataProduct>
            <PhysicalDataInstance>PDI-Y</PhysicalDataInstance>
      </StudyUnit>
      <StudyUnit id="2">
            <DataCollection>
                  <Instrument>INST-B</Instrument>
            </DataCollection>
            <LogicalDataProduct>LDP-A</LogicalDataProduct>
            <PhysicalDataProduct>PDP-D</PhysicalDataProduct>
            <PhysicalDataInstance>PDI-X</PhysicalDataInstance>
      </StudyUnit>
</Group>
```

2. Formal Group

This example shows a formal group of StudyUnits sharing common properties, for instance American Housing Survey over the course of many years:

```
<Group id="A" time="T4" instrument="I3" panel="P4" geography="G3"
datasets="D2">
      <DataCollection>All Common Collection Info</DataCollection>
      <LogicalDataProduct>Common Logical Data Structure</LogicalDataProduct>
      <PhysicalDataProduct>Common Physical Data Product</PhysicalDataProduct>
      <StudyUnit id="1">
            <Concept>
                  <Universe>1990</Universe>
            </Concept>
            <PhysicalDataInstance>1990</PhysicalDataInstance>
      </StudyUnit>
      <StudyUnit id="2">
            <Concept>
                  <Universe>1991</Universe>
            </Concept>
            <PhysicalDataInstance>1991</PhysicalDataInstance>
      </StudyUnit>
      <StudyUnit id="3">
            <Concept>
                  <Universe>1992</Universe>
            </Concept>
            <PhysicalDataInstance>1992</PhysicalDataInstance>
      </StudyUnit>
</Group>
```

3. Nested Formal Groups

This example shows nested formal Groups, for instance, the Current Population Survey, which provides a sub set of topical questions on a monthly basis. The top level Group contains the basic set of questions, which apply to every month. The next level Group contains the topical questions for a given month:

```
<Group id="A" time="T2" instrument="I3" panel="P4" geography="G4"
datasets="D4">
      <DataCollection>
            <ResearchInstrument>
                  <Question id=”Q1”>Question1</Question>
                  <Question id=”Q2”>Question2</Question>
                  <Question id=”Q3”>Question3</Question>
```

```
            </ResearchInstrument>
      </DataCollection>
      <Group id="A1" time="T2" instrument="I1" panel="P4" geography="G4"
datasets="D2">
            <DataCollection>
                  <ResearchInstrument>
                        <Question id=”Q4”>Question4</Question>
                        <Question id=”Q5”>Question5</Question>
                  </ResearchInstrument>
            </DataCollection>
            <LogicalDataProduct>Jan Logical Data
Structure</LogicalDataProduct>
            <PhysicalDataProduct>Jan Physical Data
Product</PhysicalDataProduct>
            <StudyUnit id="A11">
                  <Concept>
                        <Universe>Jan1999</Universe>
                  </Concept>
                  <PhysicalDataInstance>Jan1999</PhysicalDataInstance>
            </StudyUnit>
            <StudyUnit id="A12">
                  <Concept>
                        <Universe>Jan2000</Universe>
                  </Concept>
                  <PhysicalDataInstance>Jan2000</PhysicalDataInstance>
            </StudyUnit>
            <StudyUnit id="A13">
                  <Concept>
                        <Universe>Jan2001</Universe>
                  </Concept>
                  <PhysicalDataInstance>Jan2001</PhysicalDataInstance>
            </StudyUnit>
      </Group>
      <Group id="A2" time="T2" instrument="I1" panel="P4" geography="G4"
datasets="D2">
            <DataCollection>
                  <ResearchInstrument>
                        <Question id=”Q4”>Question4</Question>
                  </ResearchInstrument>
            </DataCollection>
            <LogicalDataProduct>Feb Logical Data
Structure</LogicalDataProduct>
            <PhysicalDataProduct>Feb Physical Data
Product</PhysicalDataProduct>
            <StudyUnit id="A21">
                  <Concept>
                        <Universe>Feb1999</Universe>
                  </Concept>
                  <PhysicalDataInstance>Feb1999</PhysicalDataInstance>
            </StudyUnit>
            <StudyUnit id="A22">
                  <Concept>
                        <Universe>Feb2000</Universe>
                  </Concept>
                  <PhysicalDataInstance>Feb2000</PhysicalDataInstance>
            </StudyUnit>
            <StudyUnit id="A23">
                  <Concept>
                        <Universe>Feb2001</Universe>
                  </Concept>
                  <PhysicalDataInstance>Feb2001</PhysicalDataInstance>
            </StudyUnit>
      </Group>
```

```
</Group>
```

4. Mixed Groups

This example shows a informal Group containing both StudyUnits and formal Groups, for instance studies funded by United States Department of Housing and Urban Development, grouped together. This group contains one StudyUnit, and a formal Group representing the American Housing Survey:

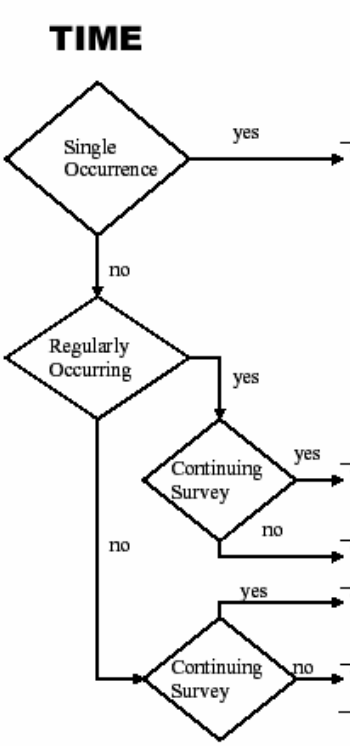
```
<Group id ="A" time="T0" instrument="I0" panel="P0" geography="G0"
datasets="D0">
       <DataCollection>
              <CollectionEvent>CommonCollector</CollectionEvent>
       </DataCollection>
       <StudyUnit id="1">
              <DataCollection>
                     <Instrument>INST-A</Instrument>
              </DataCollection>
              <LogicalDataProduct>LDP-B</LogicalDataProduct>
              <PhysicalDataProduct>PDP-C</PhysicalDataProduct>
              <PhysicalDataInstance>PDI-Y</PhysicalDataInstance>
       </StudyUnit>
       <StudyUnit id="2">
              <DataCollection>
                     <Instrument>INST-B</Instrument>
              </DataCollection>
              <LogicalDataProduct>LDP-A</LogicalDataProduct>
              <PhysicalDataProduct>PDP-D</PhysicalDataProduct>
              <PhysicalDataInstance>PDI-X</PhysicalDataInstance>
       </StudyUnit>
       <Group id="AA" time="T4" instrument="I3" panel="P4" geography="G3"
       datasets="D2">
              <DataCollection>Common Collection Info</DataCollection>
              <LogicalDataProduct>Common Logical Data
       Structure</LogicalDataProduct>
              <PhysicalDataProduct>Common Physical Data
       Product</PhysicalDataProduct>
              <StudyUnit id="AA1">
                     <Concept>
                            <Universe>1990</Universe>
                     </Concept>
                     <PhysicalDataInstance>1990</PhysicalDataInstance>
              </StudyUnit>
              <StudyUnit id="AA1">
                     <Concept>
                            <Universe>1991</Universe>
                     </Concept>
                     <PhysicalDataInstance>1991</PhysicalDataInstance>
              </StudyUnit>
              <StudyUnit id="AA1">
                     <Concept>
                            <Universe>1992</Universe>
                     </Concept>
                     <PhysicalDataInstance>1992</PhysicalDataInstance>
                     </StudyUnit>
       </Group>
</Group>
```

D. Group Properties

PARAMETER	TAG	DESCRIPTION
TIME	T0	No Formal Relationship
	T1	Single Occurrence
	T2	Multiple Occurrence: Regular Occurrence: Continuing
	T3	Multiple Occurrence: Regular Occurrence: Limited time
	T4	Multiple Occurrence: Irregular Occurrence: Continuing
	T5	Multiple Occurrence: Irregular Occurrence: Limited time
INSTRUMENT	I0	No Formal Relationship
	I1	Single
	I2	Multiple: Integrated set of 2 or more instruments used for different subgroups
	I3	Multiple: Base with Topical changes
PANEL	P0	No Formal Relationship
	P1	Single panel surveyed multiple times
	P2	Single panel surveyed once
	P3	Rolling panel (multiple interviews limited duration)
	P4	Different panel each survey
GEOGRAPHY	G0	No Formal Relationship
	G1	Single geography surveyed multiple times
	G2	Single geography surveyed once
	G3	Rolling geography (multiple interviews limited duration)
	G4	Different geography each survey
DATA SETS	D0	No Formal Relationship
	D1	Single data file from a data collection
	D2	Multiple data products from a single data collection
	D3	Integration of multiple data sets into a single integrated structure
	D4	Multiple data files each from a different data collection

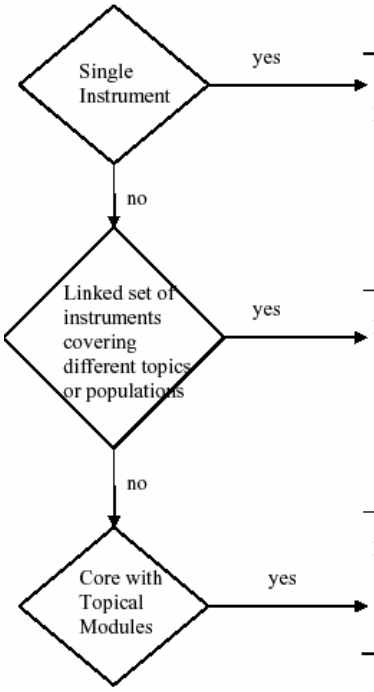
For formal Groups, each of the properties listed above is clarified in one of the following diagrams. Note that the values, T0, I0, P0, G0, and D0 denote that the Group contains no formal relationships between its children. These values are not shown in the diagrams below.

1. Time



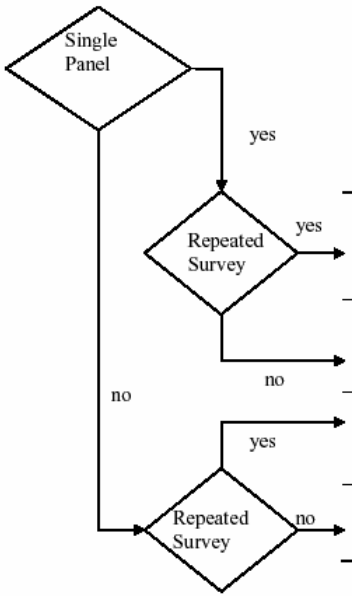
VALUE	EXPECTED INFORMATION	
	In GROUP	**Lower in Model [in Data Collection]**
T1	Single Data Date Range	
T2	Periodicity Start Date	Single Date Date Range
T3	Periodicity Date Range	Single Date Date Range
T4	Description of occurrence model Start Date	Single Date Date Range
T5	Description of occurrence model Date Range	Single Date Date Range

2. Instrument



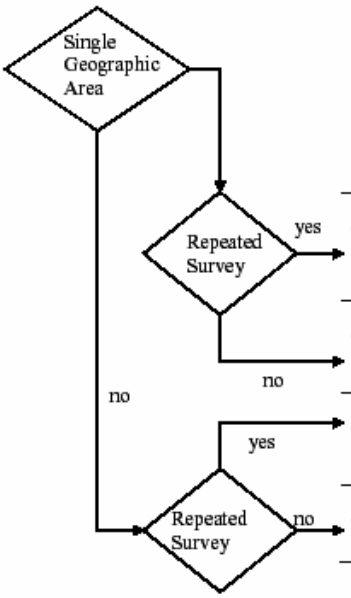
VALUE	EXPECTED INFORMATION	
	In GROUP	**Lower in Model [in Data Collection]**
I1		Instrument specific information
I2	Number of instruments Relationship of Instruments Linking mechanism	Relationship between instrument and panel(s) Instrument specific information
I3	Variance in periodicity of core and topic modules	Relationship between instrument and panel(s) Instrument specific information

3. Panel



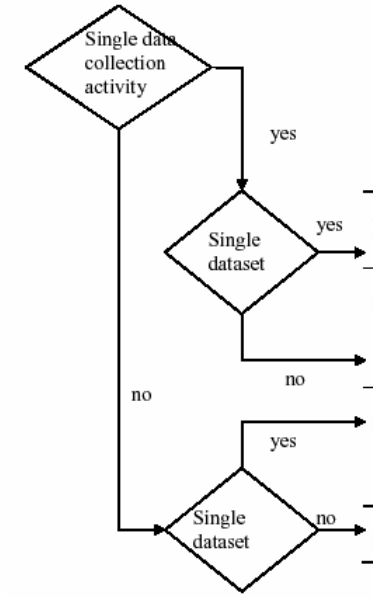
VALUE	EXPECTED INFORMATION	
	In GROUP	**Lower in Model** [in Data Collection and not covered in other attribute information]
G1	Number of repeats for panel Universe	
G2		Universe Sampling method and size
G3	Overall Universe Wave pattern Number of repeats per panel	Specific Universes Specifics in sampling method and size
G4	Overall Universe Sample selection/ differentiation	Specific Universe Specifics in sampling method and size

4. Geography



GEOGRAPHY	VALUE	EXPECTED INFORMATION	
		In GROUP	**Lower in Model** [in Data Collection and not covered in other attribute information]
Single Geographic Area → Repeated Survey → yes	G1	Number of repeats for geography Geographic Cover	
Single Geographic Area → Repeated Survey → no	G2		Geographic Cover
Single Geographic Area → no → Repeated Survey → yes	G3	Overall Geographic Cover Wave pattern Number of repeats per geography	Specific Geographic Cover
Single Geographic Area → no → Repeated Survey → no	G4	Overall Geographic Cover Geographic selection/ differentiation	Specific Geographic Cover Changes in geographic selection

5. Datasets



VALUE	EXPECTED INFORMATION	
	In GROUP	**Lower in Model** [in Data Collection or lower and not covered in other attribute information]
D1		
D2	Rational for multiple logical products Product mix details	Specific logical data set information [in logical data descriptions with links back to questionnaire]; possible concept keys [logical module}
D3	Integration basis Purpose of integration Selection process Linking process Harmonization process	Specifics of individual data collection processes
D4	Dataset to data collection activity mapping	

IX. Survey Instruments

Elements describing the questionnaire content and structure have been moved from the variable element into a sub-module of the data collection process. This allows for a more coherent and richer description of the survey instrument and the means of data collection (face-to-face interview, mail out form, phone interview, CAI, etc. The 2.0 elements concerning question content will be held in this module and a working group is determining how to capture additional information and relate it to other work being done in this area. Additionally, multiple data file descriptions can point to the same questionnaire when the data collected is used to create multiple products or the questionnaire is administered multiple times.

X. HTML Tagging

[*This section will describe the use of HTML tags in free-text areas, and how applications can detect their use or non-use for those cases where application support is a question.]*

XI. Uniform Notes & Citations

A. Reusable Classes

File/Section ID, Citation, and Universe:
Version 2.0 of the DDI allows for the description of bibliographic citations, universe descriptions, other related materials, and notes at numerous and specific places throughout its structure. Version 3.0 has pulled these out and created uniform structures for each of these classes. The reusable classes are available in each of the modules and may be linked to any element within the module. This approach increases both the consistency of the structure and the flexibility for application of references to outside materials and internal notes. A more extensive and structured type identifier is used to assist the programmer and user in sorting through the information held in each class structure.

The Version 3.0 citation has been divided into three parts:

- File/Section ID: This is the equivalent of holdings information in a citation [where something is located and how it is referenced]
- Citation: This is the bibliographic citation information that doesn't change [author, title, publisher, publication place and date]
- Universe: This is the topical, geographic, and temporal coverage of the module or item. By separating this information out, it allows for local enhancement, or the identification of items covering subsets of the overall data set [for example, a separation of an international data file into individual files for each country each with its own universe description or the separation of a hierarchical file into its component record types].

B. Notes

The primary change in the use of notes is that they are now grouped together in a class that is available in each module of the DDI. Notes can be referenced from any element, providing a level of flexibility not available in Version 2.0. In addition, a set of types is being developed to identify specific types of commonly used notes to increase capabilities for uniform processing by software systems.

C. Other Material

A single uniform structure for identifying, describing and pointing to Other Materials of all types has been developed and added to the model in a similar format to Notes. This includes a single class structure available in each module and the ability to point to a other material reference from any element within the module. In nested modules, Other Material contents can be inherited down the tree and referenced from lower modules.

XII. Alignment with Other Standards

- METS
- Dublin Core
- Metadater
- ISO 11179
- Others? (SDMX...)

[We have the chart we created at the October meeting – we will choose others from that.]