WORKING DRAFT

DDI VERSION 3.0 CONCEPTUAL MODEL

STRUCTURAL REFORM GROUP

10 JUNE 2004

BACKGROUND OF THE CONCEPTUAL MODEL

The conceptual model of the DDI provides the common structure that the technical implementations refer to and describe through various approaches including DTDs and schema. The conceptual model encompasses the logic of the structure covering why and how parts relate to each other. The technical implementations explain how the conceptual model works within a specific technology.

Originally, the DDI took its model from the codebook which assumed a finalized clean version of the data. It was clear early on that many were expanding that concept to mean something much broader and perhaps more complex than a traditional hard copy codebook. There was discussion about using the DDI to capture information during the creation process, but the focus was still on the final data set. Additionally, the "codebook" approach suffered from the lack of a clear idea of what a codebook encompassed. In fact, even a brief glance at the "codebooks" represented in the ICPSR collection shows that the idea of a "codebook" has never been consistently defined in either content or structure, and that the reality of codebook construction reflects infinite variety.

As development of the DDI has progressed we have begun to move towards looking at the DDI as encompassing the development life cycle of a data set. Version 3.0 reflects this change in scope and no longer makes the assumption of a finalized version of the data. In order to support this broader scope, a modular structure is required. This allows the addition of modules or sub-modules as the data is developed, encoded, preserved, disseminated, and analyzed.

In order for the DDI to provide a structure that supports both programming and archival activities, we need to have a well structured and well understood model. We need this in order to provide consistent application of the standard as well as resolve questions regarding the application of the structure to a specific instance of an XML document.

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.

GOALS FOR MODULAR DESIGN

- To capture information on the creation of specific data within a production/technical environment in a way that it can be accurately transferred as the data travels through and outside of that environment
- To organize the modules so that they accurately record information about data and the data creations 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 (use example of a physical store that is described in a specialized module)

DESIGN OF MODEL

In many ways the original codebook concept captured a specific sub-set of information about the data at a specific point in time, a static picture. In the Version 3.0 model we wish to capture a more dynamic structure, recognizing what information continues through and what changes as the data moves through time. In short, we wish to capture the dynamic nature of the metadata. To do so we need to understand its life cycle.

The metadata life cycle as described by Ann Green and Jean-Pierre Kent in chapter 2.2 of MetaNet Work Package 1: Methodology and Tools,[1] begins with a production phase model (input, throughput, output) and then expands this to include a conceptual phase prior to the input phase, and a repurposing and preservation phase after the output phase. The conceptual phase concentrates on the design issues that take place before a process starts. This includes the design of the entire process model.

> "While output data are dependent on input data,, and can only be produced after input data have been collected, input metadata are inferred from output metadata. One cannot start thinking about what data to collect and how before one precisely knows what the end product will be and how it will be achieved." (Green & Kent, p.33)

The repurposing phase provides for the documentation of secondary analysis resulting from the original data output. The preservation phase focuses on the processes involved in long-term preservation of the data and metadata. This includes format changes to remove system or software dependencies, cleaning, integration and harmonization, and the development of related materials.

---

[1] http://www.epros.ed.ac.uk/metanet/deliverables/D4/IST_1999_29093_D4.pdf

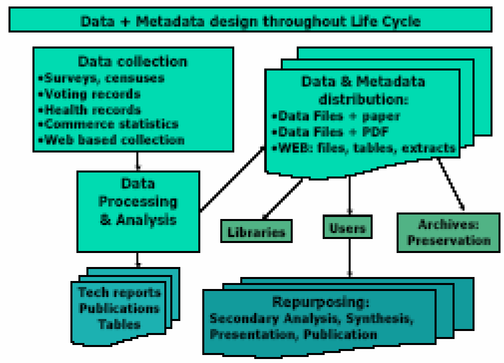


Figure 1: The Data/Metadata life cycle

Figure 2 shows an initial working model provided by I-lin Kuo from a programming or data application perspective. I-lin Kuo's model (below) and the Green/Kent life cycle model have parallel features but different areas of detail and emphasis. The *Survey Design* could be seen as analogues to the concept phase described above. *Data Processing* encompasses "Data Processing & Analysis" and "Tech reports" while *Data Distribution* is similar in each. *Data Delivery* may cover parts of "Libraries", "Users", "Archives: Preservation", and "Repurposing", while *Data Analysis* would most likely be contained within the "Repurposing" phase of the Green/Kent model. By adding the module of *Data Discovery*, Kuo bring in the issue of how the information within the structured metadata and data are being used.

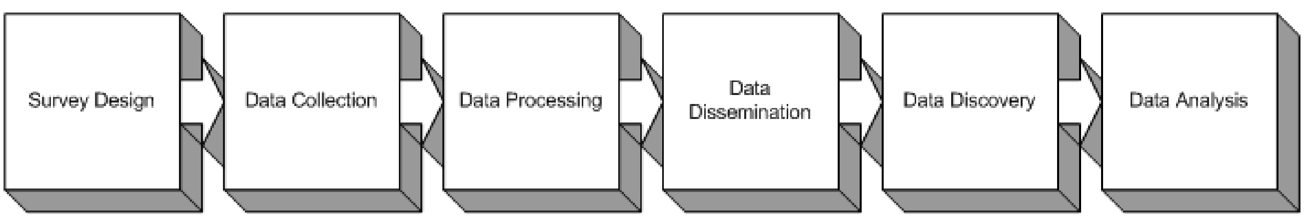


Figure 2: I-lin Kuo Model

In Kuo's model the first two boxes focus on specific aspects of the creation of data and documentation, but miss some important details in terms of modeling the DDI, primarily the preservation aspects of archiving and the repurposing of the data discussed in the Green/Kent model. By combining features of the two life cycle models we get a fuller picture.

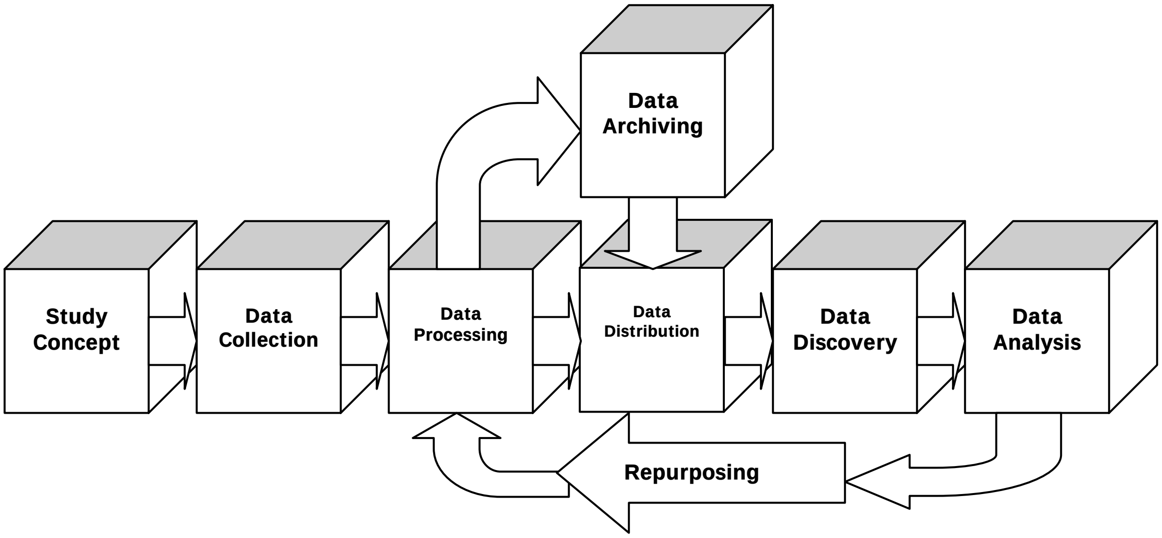


Figure 3: Combined Life Cycle Model

The Combined Life Cycle Model incorporates either direct dissemination to users or through data archives and recognizes that data can be reprocessed at a later point in its life cycle, creating an iterative process. However, it is no longer a linear but a circular model. From the concept of time 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 becomes a new 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 BASIC VERSION 3.0 CONCEPTUAL MODEL

The model of metadata below focuses on what is done with the data, in other words the data's life cycle, as opposed to how it is used. Like the model above it contains both a conceptual and data collection module. Rather than a distribution module, further differentiation is given to the description of the output data, dividing it into logical structure and physical structure modules. A combination of the modules to this point (the metadata) and the data are then prepared for discovery and distribution by an entity (here described in the Archiving module) which may further enhance the data or metadata, provide internal identifiers and processing information, and impose rules of access and distribution. In a broad sense it could encompass the "Libraries", "Users", and "Archives: Preservation" phases of the Green/Kent model. It does not address *Data Discovery* or

Data Analysis directly, but instead sees these as processes to be supported by the contents of the modules included in the model below.

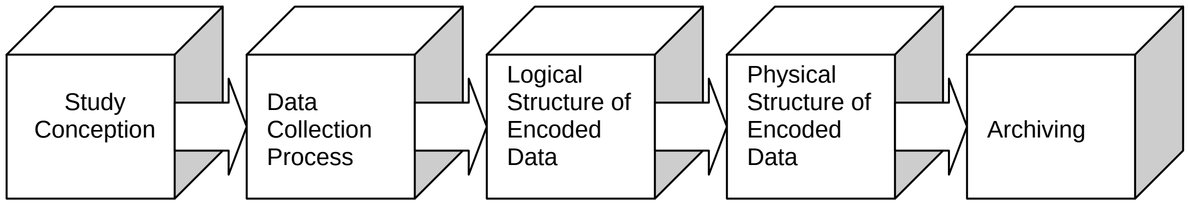


Figure 4: Modules in the Conceptual Model

Each of the basic modules can contain multiple sub-modules to reflect the specific descriptive needs of different types of processes or materials. In addition, some modules may be replicated, say if you have two physical instances of the same data in different storage formats (example: ASCII Fixed Format and STATA Data File). In addition, there clearly needs to be a "wrapper" which identifies the modules and module sub-types included in any given instance.

STRUCTURAL APPROACH FOR CONCEPTUAL MODEL

The color diagram of the DDI Version 3.0 Conceptual Model (attached), shows the inter-relationship of the modules noted in Figure 4 and the those shown in Figure 2. It separates the use of the information in the metadata and data for study level discovery, detailed discovery, and data access. It is important to differentiate various points along this range of usage because of the issues surrounding interoperability with other metadata structures, high level search engines, and systems which are closely tied to a specific technology or use. The following design rules should be used to determine what belongs in each module and whether the module should have a persistent structure or have alternative structures to deal with specialized data files, studies, or technical needs.

DESIGN RULES

- Persistent sections should be separate from dynamic information
- Information modules should follow through the various lifecycle 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 discover systems together and/or uniformly accessible.

Implications for Data Discovery

This structure has major implications for data discovery and use. First there are different levels of discovery and different approaches to discovery.

The most basic discovery approach is through the Dublin Core elements and the holding information for a specific archive. In short, what it is and where it is located. The Dublin Core information is persistent while the holding information will change. Dublin Core works for bibliographically based search systems but does not have the content detail or control of other widely used bibliographic systems. In particular it does not carry the geographic detail needed by most geographically based search systems. The use of the field COVERAGE by the Dublin Core does not following the restrictions imposed by geographic search systems and the Dublin Core does not provide options for coordinate based searching.

The next major level of discovery is in-depth identification of study contents at the level of variable or data items labels, category labels, universe statements and other information held in the logical structure material. This level of discover can answer specific questions about what data exists and if it is encoded in a manner that is useful for the individual user. It does not include, but is necessary for, the next level of use, that of data access and/or manipulation.

In order to access data the system needs to be able to identify the following pieces of information:

- Identification of the record and data items required
- Identification of the physical store and the location of the record and data items within that store (a link between the physical description and logical description)
- Logic of record selection
- Ability to process the specific physical data format
- Ability to output, manipulate and/or display the data retrieved from the physical store

Module Contents

Concept Module

- 1 – Not a repeatable base module
- Contains identifying information for XML; all Dublin Core elements
- Can be used as a stand-alone bibliographic record (not including holding information)
- Provide information on the context of the study, relationship to series or family of data sets
- Information on the purpose of the study, universe, and coverage in its broad aspects

Data Collection Process Module

- Should be able to have multiples of this…multiple questionnaires etc..
- Should this replicate or have different 'swappable' modules to reflect difference processes? Should this be a collection of sub-modules?
- Contain information on the development of the data collection tool and implementation
- Question text should go here as well as intent of questions and interviewer or recorder instruction (this allows questions to act as the source for multiple data items described in one or more logical encodings of the raw data.
- Should contain sub-modules for related resources used for processing or collecting data such as coding schemes, sampling software, etc.
- Should contain sub-modules to include or describe output materials related to the Data Collection and Processing

Logical Encoding Module

- Swappable and repeatable to reflect different types of logical data structures derived from the initial data collection
  - Raw-Microdata
    - Simple survey
    - Multiple file survey
    - Time series
    - Dynamic
    - ..
  - Aggregate
  - Others
- Describes the logical encoding of the raw collected data
- Links to questions where appropriate and includes encoding instructions

Physical Encoding Module

- 1+ swappable and repeatable
- (Need to allow for multiple stores of the same data set without replicating everything)
- Different modules for different format types
  - One dimensional
    - rectangular
    - hierarchical
  - Two dimensional
    - rows and column
  - Three dimensional
    - rows/columns and layers
  - Proprietary database
    - oracle
    - sass

    - spss
- Links to logical encoded description of data items
- Provides gross file structure description and relationship information between files and/or records

Archiving Module
- 1 holding archive information
- Sub-modules could be swappable regarding restrictions, access criteria….how much do these differ? This is one area where individuality by archive is not a problem unless they want to share this access information. There could be a generic base module with swappable sub-modules

POPULATING THE CONCEPTUAL MODEL

The attached color coded listing of elements from DDI Version 2.0 provides a preliminary suggestion of which module the current elements would fall into. Some of these elements are preceded by colored stars (*) indicating that their "home" is unclear. This may be due to a number of reasons. Sometimes the element is not well differentiated and may contain a mixture of information. At other times the element is used in more than one way and its "home" depends on how it is used in a specific instance. These will be clarified as the conceptual model takes on more structure.

MAJOR QUESTIONS REMAINING

The following questions will need to be answered before the completion of the conceptual model. They have been raised a number of times in discussions both inside and outside of the DDI Alliance. The include

- How do we define an instance? What constitutes a new instance?
- What modules are required and what are the basic requirements of each module?
- How do we handle versions, editions and copies of the instance as a whole as well as the individual modules?
- How do we leverage the perspective of ISO 11-179 CMR to take advantage of the rich descriptive framework of the concepts underlying the data?
- How will the topics being discussed in the Substantive Working Groups be incorporated? We need to clarify where these will fit conceptually in the life cycle.

CONCLUSION

This working draft sets out the basic conceptual model proposed for Version 3.0 of the DDI. We have tried to provide sufficient background information for others to understand our perspective and intent for the overall structure of Version 3.0. It is definitely still a work in progress and needs to be further defined, particularly in terms of the contents of the basic modules. However, we also need to confirm that the basic structure is sound and comprehensive. In order to do that, in-put is required from the Expert Users Group. We propose the following process for accomplishing the development of the conceptual model and production of the first technical implementation by the end of 2004. The timeframe is suggestive and will be adjusted to fit the needs of group, but our goal is to have the conceptual model clearly defined by early October so that work can begin on the technical implementations.

Time Period	Activity
Mid-June to Mid-July	• Get feedback on the broad model • Answer questions if needed • Finalize the basic modules
July	• Clarify contents of the basic modules • Populate the structure with the current fields from Version 2.0
August	• Get feedback on distribution of Version 2.0 items • Clarify unclear relationships
September	• Expert Committee review of final version

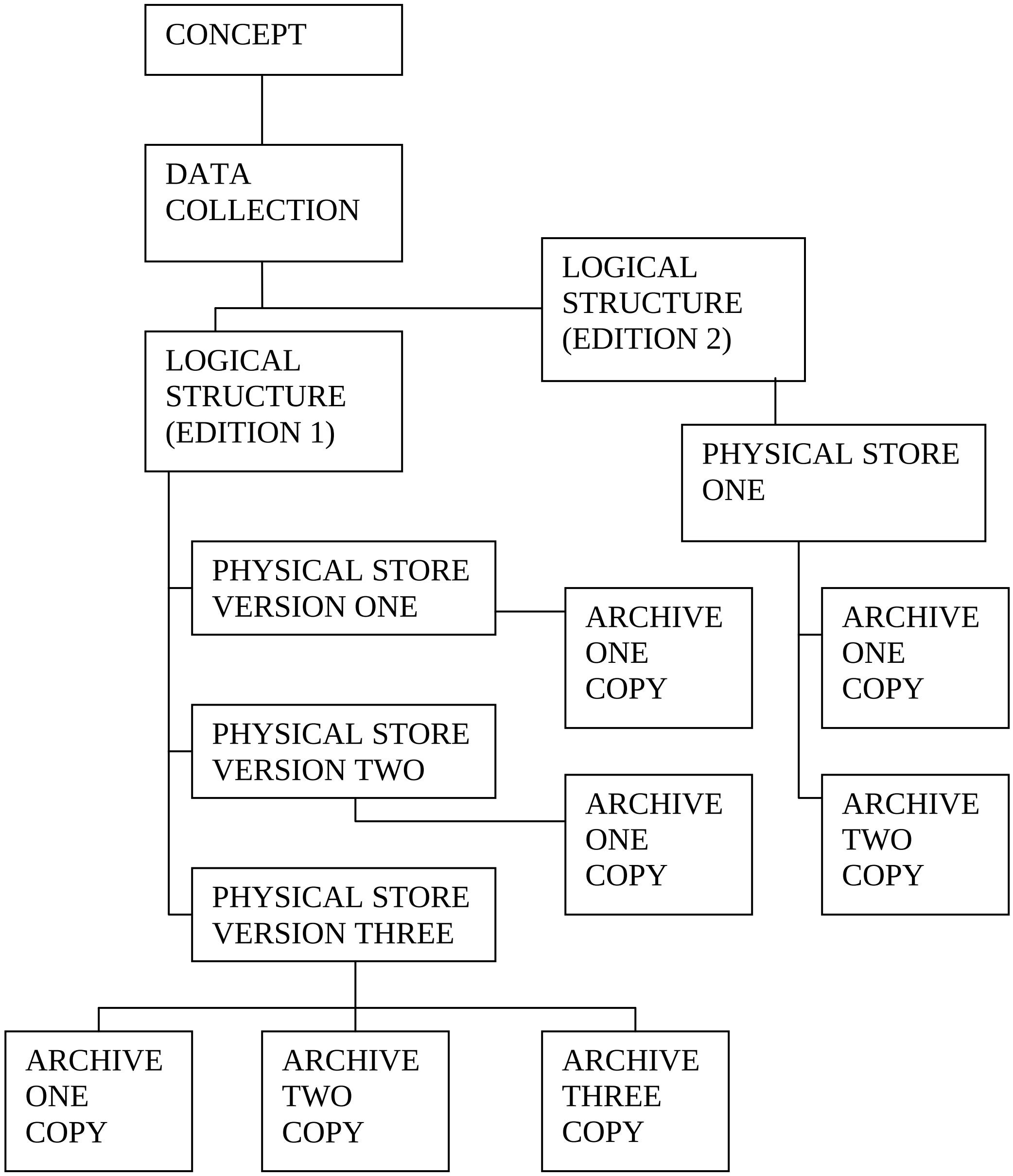
Please direct all comments and questions to the Expert Group Listserve <alliance-experts@icpsr.umich.edu>

APPENDIX:

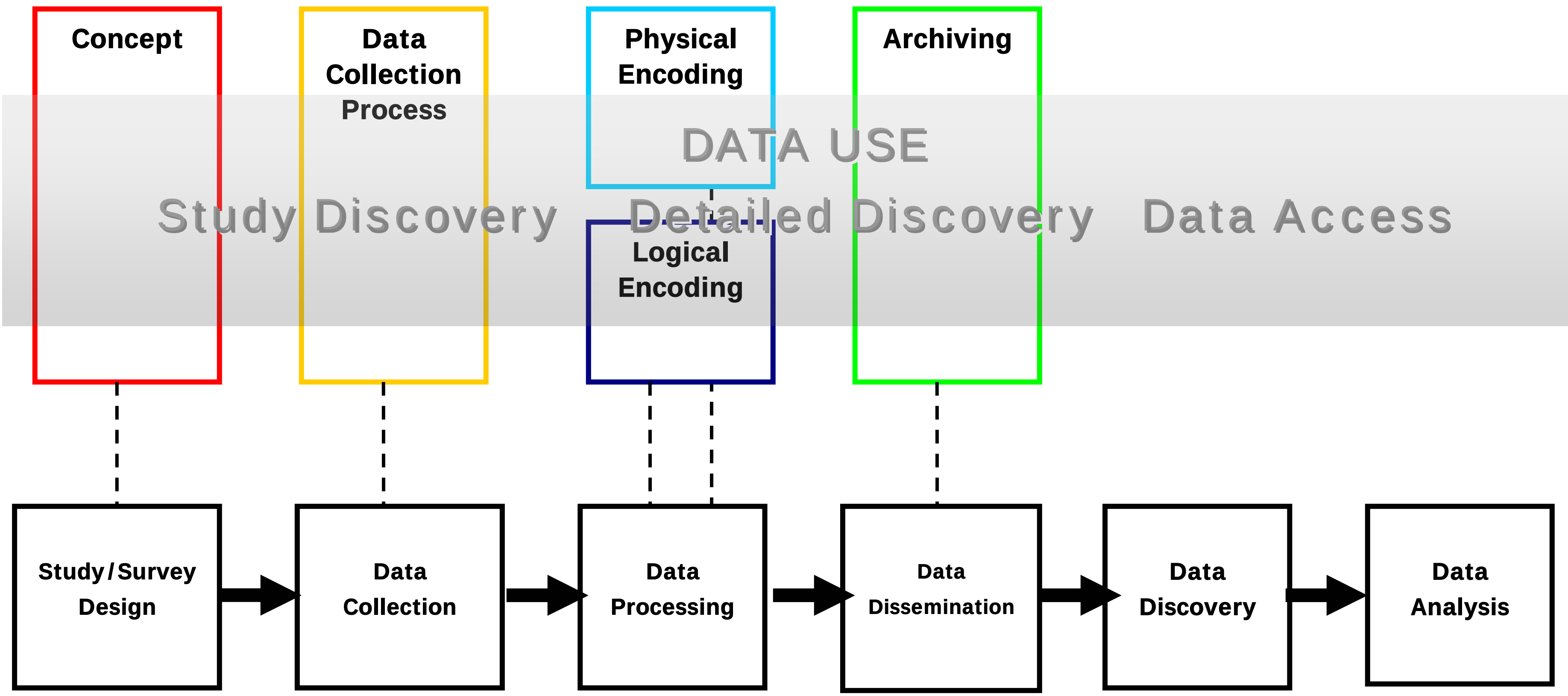
Multiple Use Modules, Versions, Editions, and Copies

The same data set could also have multiple versions of the same logical structure stored in different physical formats (similar to a translation of text). Finally the same original data set could have multiple editions where information in the logical structure has been corrected, limited in the coverage of elements or records (subsets) or contain added items such as recodes or alternative geographic codes. The graphic on the following page shows a tree of data sets resulting from a single study.

In this hierarchy the study had two editions of the logical structure. Edition 1 has three versions of the data set (for example, fixed format ASCII, a comma delimited file, and a proprietary data base file). Archive 1 hold a copy of all physical stores of each edition of the study. Archive 2 has a copy of one physical store for each edition, and Archive 3 has only a single physical store of edition 1.


CONCEPT
DATA COLLECTION
LOGICAL STRUCTURE (EDITION 2)
LOGICAL STRUCTURE (EDITION 1)
PHYSICAL STORE ONE
PHYSICAL STORE VERSION ONE
ARCHIVE ONE COPY
ARCHIVE ONE COPY
PHYSICAL STORE VERSION TWO
ARCHIVE ONE COPY
ARCHIVE TWO COPY
PHYSICAL STORE VERSION THREE
ARCHIVE ONE COPY
ARCHIVE TWO COPY
ARCHIVE THREE COPY

DDI V.3.0 - Conceptual Model 24 May 2004



DDI Codebook Outline -- Version 2.0

WORKING DRAFT

* == ELEMENT IS OPTIONAL & REPEATABLE
+ == ELEMENT IS MANDATORY & REPEATABLE
? == ELEMENT IS OPTIONAL & NON-REPEATABLE
== ELEMENT IS MANDATORY & NON-REPEATABLE

CONCEPT MODULE
DATA COLLECTION MODULE
PHYSICAL ENCODING MODULE
LOGICAL ENCODING MODULE
ARCHIVING MODULE
OBTAINING DATA MODULE
* COULD BE SECTION DENOTED BY COLOR OF STAR DEPENDENT ON USE

0.0 codeBook
1.0 docDscr*
1.1 citation?
1.1.1 titlStmt
1.1.1.1 titl
1.1.1.2 subTitl*
1.1.1.3 altTitl*
1.1.1.4 parTitl*
1.1.1.5 IDNo*
1.1.2 rspStmt?
1.1.2.1 AuthEnty*
1.1.2.2 othId*
1.1.3 prodStmt?
1.1.3.1 producer*
1.1.3.2 copyright?
1.1.3.3 prodDate*
1.1.3.4 prodPlac*
1.1.3.5 software*
1.1.3.6 fundAg*
1.1.3.7 grantNo*
1.1.4 distStmt?
1.1.4.1 distrbtr*
1.1.4.2 contact*
1.1.4.3 depositr*
1.1.4.4 depDate*
1.1.4.5 distDate?
1.1.5 serStmt?
1.1.5.1 serName*
1.1.5.2 serInfo*
1.1.6 verStmt*
1.1.6.1 version?
1.1.6.2 verResp?
1.1.6.3 notes*
1.1.7 biblCit?
1.1.8 holdings*
1.1.9 notes*
* 1.2 guide?
1.3 docStatus?
1.4 docSrc*
1.4.1 titlStmt
1.4.1.1 titl
1.4.1.2 subTitl*
1.4.1.3 altTitl*
1.4.1.4 parTitl*
1.4.1.5 IDNo*
1.4.2 rspStmt?
1.4.2.1 AuthEnty*
1.4.2.2 othId*
1.4.3 prodStmt?
1.4.3.1 producer*
1.4.3.2 copyright?
1.4.3.3 prodDate*
1.4.3.4 prodPlac*
1.4.3.5 software*
1.4.3.6 fundAg*
1.4.3.7 grantNo*
1.4.4 distStmt?
1.4.4.1 distrbtr*
1.4.4.2 contact*
1.4.4.3 depositr*
1.4.4.4 depDate*
1.4.4.5 distDate?
1.4.5 serStmt?
1.4.5.1 serName*
1.4.5.2 serInfo*
1.4.6 verStmt*
1.4.6.1 version?
1.4.6.2 verResp?
1.4.6.3 notes*
1.4.7 biblCit?
1.4.8 holdings*
1.4.9 notes*
1.5 notes*

2.0 stdyDscr+
2.1 citation+
2.1.1 titlStmt
2.1.1.1 titl
2.1.1.2 subTitl*
2.1.1.3 altTitl*
2.1.1.4 parTitl*
2.1.1.5 IDNo*
2.1.2 rspStmt?
2.1.2.1 AuthEnty*
2.1.2.2 othId*
2.1.3 prodStmt?
2.1.3.1 producer*
2.1.3.2 copyright?
2.1.3.3 prodDate*
2.1.3.4 prodPlac*
2.1.3.5 software*
2.1.3.6 fundAg*
2.1.3.7 grantNo*
2.1.4 distStmt?
2.1.4.1 distrbtr*
2.1.4.2 contact*
2.1.4.3 depositr*
2.1.4.4 depDate*
2.1.4.5 distDate?
2.1.5 serStmt?
2.1.5.1 serName*
2.1.5.2 serInfo*
2.1.6 verStmt*
2.1.6.1 version?
2.1.6.2 verResp?
2.1.6.3 notes*
2.1.7 biblCit?
2.1.8 holdings*
2.1.9 notes*
2.2 stdyInfo*
2.2.1 subject?
2.2.1.1 keyword*
2.2.1.2 topcClas*
2.2.2 abstract*
2.2.3 sumDscr*
2.2.3.1 timePrd*
2.2.3.2 collDate*
2.2.3.3 nation*
2.2.3.4 geogCover*
2.2.3.5 geogUnit*
2.2.3.6 geoBndBox?
2.2.3.6.1 westBL
2.2.3.6.2 eastBL
2.2.3.6.3 southBL
2.2.3.6.4 northBL
2.2.3.7 boundPoly?
2.2.3.7.1 polygon+
2.2.3.7.1.1 point+
2.2.3.7.1.1.1 gringLat
2.2.3.7.1.1.2 gringLon
* 2.2.3.8 anlyUnit*
* 2.2.3.9 universe*
* 2.2.3.10 dataKind*
2.2.4 notes*
2.3 method*
2.3.1 dataColl*
2.3.1.1 timeMeth*
2.3.1.2 dataCollector*
2.3.1.3 frequenc*
2.3.1.4 sampProc*

DDI Codebook Outline -- Version 2.0

WORKING DRAFT

* == ELEMENT IS OPTIONAL & REPEATABLE
\+ == ELEMENT IS MANDATORY & REPEATABLE
? == ELEMENT IS OPTIONAL & NON-REPEATABLE
== ELEMENT IS MANDATORY & NON-REPEATABLE

CONCEPT MODULE
DATA COLLECTION MODULE
PHYSICAL ENCODING MODULE
LOGICAL ENCODING MODULE
ARCHIVING MODULE
OBTAINING DATA MODULE
* COULD BE SECTION DENOTED BY COLOR OF STAR DEPENDENT ON USE

2.3.1.5 deviat*
2.3.1.6 collMode*
2.3.1.7 resInstru*
2.3.1.8 sources?
2.3.1.8.1 dataSrc*
2.3.1.8.2 srcOrig*
2.3.1.8.3 srcChar*
2.3.1.8.4 srcDocu*
2.3.1.8.5 sources*
2.3.1.9 collSitu*
2.3.1.10 actMin*
2.3.1.11 ConOps*
2.3.1.12 weight*
2.3.1.13 cleanOps*
2.3.2 notes*
2.3.3 anlyInfo?
2.3.3.1 respRate*
2.3.3.2 EstSmpErr*
2.3.3.3 dataAppr*
2.3.4 stdyClas?
2.4 dataAccs*
2.4.1 setAvail*
2.4.1.1 accsPlac*
2.4.1.2 origArch?
2.4.1.3 avlStatus*
2.4.1.4 collSize?
2.4.1.5 complete?
2.4.1.6 fileQnty?
2.4.1.7 notes*
2.4.2 useStmt*
2.4.2.1 confDec?
2.4.2.2 specPerm?
2.4.2.3 restrctn?
2.4.2.4 contact*
2.4.2.5 citReq?

2.4.2.6 deposReq?
2.4.2.7 conditions?
2.4.2.8 disclaimer?
2.4.3 notes*
** 2.5 othrStdyMat*
** 2.5.1 relMat*
** 2.5.1.1 citation*
** 2.5.2 relStdy*
** 2.5.2.1 citation*
** 2.5.3 relPubl*
** 2.5.3.1 citation*
** 2.5.4 othRefs*
** 2.5.4.1 citation*
2.6 notes*

3.0 fileDscr*
3.1 fileTxt*
3.1.1 fileName?
3.1.2 fileCont?
3.1.3 fileStrc?
3.1.3.1 recGrp*
3.1.3.1.1 labl*
3.1.3.1.2 recDimnsn?
3.1.3.1.2.1 varQnty?
3.1.3.1.2.2 caseQnty?
3.1.3.1.2.3 logRecL?
3.1.3.2 notes*
3.1.4 dimensns?
3.1.4.1 caseQnty*
3.1.4.2 varQnty*
3.1.4.3 logRecL*
3.1.4.4 recPrCas*
3.1.4.5 recNumTot*
3.1.5 fileType?
3.1.6 format?
3.1.7 filePlac?
3.1.8 dataChck*
3.1.9 ProcStat?
3.1.10 dataMsng?
3.1.11 software*
3.1.12 verStmt?
3.1.12.1 version?
3.1.12.2 verResp?
3.1.12.3 notes*
3.2 locMap?
3.2.1 dataItem*
3.2.1.1 CubeCoord*
3.2.1.2 physLoc*
3.3 notes*

4.0 dataDscr*
4.1 varGrp*
4.1.1 labl*
4.1.2 txt*
4.1.3 concept*
4.1.4 defntn?
4.1.5 universe?
4.1.6 notes*
4.2 nCubeGrp*
4.2.1 labl*
4.2.2 txt*
4.2.3 concept*
4.2.4 defntn?
4.2.5 universe?
4.2.6 notes*
4.3 var*
4.3.1 location*
4.3.2 labl*
4.3.3 imputation?
4.3.4 security?
4.3.5 embargo?
4.3.6 respUnit?
4.3.7 anlysUnit?
4.3.8 qstn*
4.3.8.1 preQTxt*
4.3.8.2 qstnLit*
4.3.8.3 postQTxt*
4.3.8.4 forward*
4.3.8.5 backward*
4.3.8.6 ivuInstr*
4.3.9 valrng*
4.3.9.1 range*
4.3.9.2 item*
4.3.9.3 key?
4.3.9.4 notes*
4.3.10 invalrng*
4.3.10.1 range*
4.3.10.2 item*
4.3.10.3 key?
4.3.10.4 notes*
4.3.11 undocCod*
4.3.12 universe*
4.3.13 TotlResp?
4.3.14 sumStat*
4.3.15 txt*
4.3.16 stdCatgry*
4.3.17 catgryGrp*
4.3.17.1 labl*
4.3.17.2 catStat*
4.3.17.3 txt*
4.3.18 catgry*
4.3.18.1 catValu?
4.3.18.2 labl*
4.3.18.3 txt*
4.3.18.4 catStat*

DDI Codebook Outline -- Version 2.0

WORKING DRAFT

* == ELEMENT IS OPTIONAL & REPEATABLE
\+ == ELEMENT IS MANDATORY & REPEATABLE
? == ELEMENT IS OPTIONAL & NON-REPEATABLE
== ELEMENT IS MANDATORY & NON-REPEATABLE

CONCEPT MODULE
DATA COLLECTION MODULE
PHYSICAL ENCODING MODULE
LOGICAL ENCODING MODULE
ARCHIVING MODULE
OBTAINING DATA MODULE
* COULD BE SECTION DENOTED BY COLOR OF STAR DEPENDENT ON USE

4.3.18.5 mrow?
4.3.18.5.1 mi*
4.3.19 codInstr*
4.3.20 verStmt*
4.3.20.1 version?
4.3.20.2 verResp?
4.3.20.3 notes*
4.3.21 concept*
4.3.22 derivation?
4.3.22.1 drvdesc?
4.3.22.2 drvcmd?
4.3.23 varFormat?
4.3.24 geoMap*
4.3.25 notes*
4.4 nCube*
4.4.1 location*
4.4.2 labl*
4.4.3 txt*
4.4.4 universe*
4.4.5 imputation?
4.4.6 security?
4.4.7 embargo?
4.4.8 respUnit?
4.4.9 anlysUnit?
4.4.10 verStmt*
4.4.10.1 version?
4.4.10.2 verResp?
4.4.10.3 notes*
4.4.11 purpose?
4.4.12 dmns*
4.4.12.1 cohort*
4.4.12.1.1 range*
4.4.13 measure*
4.4.14 notes*
4.5 notes*

** 5.0 otherMat*
** 5.1 labl*)
** 5.2 txt?
** 5.3 notes*
** 5.4 table*
** 5.5 citation?
** 5.6 otherMat*