

Data Documentation Initiative (DDI) Technical Specifications

Part I:

Overview

**(Version 3.0)
For Public Review**

February 2007

3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

Editing Team: Wendy Thomas, Arofan Gregory, I-Lin Kuo, Joachim Wackerow,
Chris Nelson

For public review

© DDI Alliance 2007

<http://www.ddialliance.org>

44 **Contents**

45

46 I. Introduction4

47 II. DDI Version 3.0 Conformance5

48 III. Definition of Terms.....6

49 III. Life Cycle Metadata and Implications for the DDI 10

50 A. DDI Instances and the Life Cycle..... 11

51 B. Repurposing of Data 12

52 C. Two Uses of the DDI: “Simple” Cases versus “Complex” Cases 12

53 1. Simple Case 13

54 IV. Migration and Modularization Design 16

55 A. Migration of 2.0 Elements 16

56 B. Modular Design..... 16

57 1. Goals for Modular Design 17

58 2. Design Rules 17

59 V. Simple Instances 18

60 A. Mapping from Version 2.* to Version 3.0 18

61 B. Functional Types..... 19

62 C. Module Descriptions20

63 VI. Multiple Files.....22

64 VII. Comparison22

65 VIII. Grouping.....22

66 A. Overall Structure23

67 B. Classes24

68 1. Instance24

69 2. Group.....24

70 3. StudyUnit24

71 4. Study Conceptual Classes.....24

72 5. Comparison25

73 C. Examples.....25

74 1. Informal Group.....25

75 2. Formal Group.....26

76 3. Nested Formal Groups26

77 4. Mixed Groups27

78 IX. Survey Instruments.....28

79 X. HTML Tagging.....29

80 XI. Uniform Notes & Citations29

81 A. Reusable Classes29

82 B. Notes30

83 C. Other Material30

84 XII. Alignment with Other Standards.....30

85

86 I. Introduction

87

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

98

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

105

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

113

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

119

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

124

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

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

133
134 While the changes in DDI Version 3.0 are ambitious in scope, one of the major
135 design goals is to avoid making migration from Version 2.* any more arduous
136 than necessary. The simple use of DDI for archival purposes is not radically
137 different between versions, and mappings of all currently-used fields will be
138 provided, as will some simple free tools for helping users.

139
140 Some of the biggest changes are the result of advances in XML technology.
141 Because the use of W3C XML Schema (XSD) has become mainstream, the DDI
142 DTD will no longer be the canonical expression of the standard. Instead, it will be
143 a sister-product of the Schema, which – while it also describes XML instances –
144 will express more of the validation parameters than are possible with a DTD.

145
146 The use of XML namespaces is another typical XML practice which DDI Version
147 3.0 will introduce. This allows the now-expanded vocabulary to be modularized,
148 making it more manageable and maintainable over the long run.

149
150 Another change found in Version 3.0 is its ability to directly contain data, if
151 desired. This change is introduced to facilitate the use of modern web-services
152 technologies, and also to permit the use of XML as an archival format.

153
154 It should be stated that DDI Version 3.0 intends to increase the degree to which
155 the metadata it contains is sufficient to support computer processing – that is, it
156 will go beyond being “human readable”, and move toward the goal of being
157 “machine-actionable”. This is a long-term goal, and will not be taken too far in the
158 early 3.* versions, but it is very much in keeping with the overall use of XML-
159 based technologies now current, such as Web services. One aspect of this is the
160 capability within Version 3.0 for DDI instances to describe the subset of fields
161 which are actually used, allowing applications to automatically determine the
162 extent of interoperability and useful of these DDI instances. This is known as a
163 “DDI profile”.

164

165 **II. DDI Version 3.0 Conformance**

166
167 Most of the DDI Version 3.0 Technical Specifications package is non-normative –
168 that is, it provides the information needed to understand and use the
169 specifications, but is not a set of rules which must be followed. The sections of
170 these specifications which are normative include the XML schemas and
171 documentation found in Part IV, and the sections of Part II: High-Level
172 Documentation which address Identification, Versioning, and Maintenance; the
173 structuring of DDI-specific URNs, the creation of conformant extensions to the
174 DDI, and the use of controlled vocabularies.

175

176 It is the intent of DDI in future versions to support fully machine-actionable
177 interoperability between different applications across the statistical lifecycle, and
178 between archives. To meet this goal, strict conformance criteria must be
179 established. This version of the DDI Technical Specifications is not yet at that
180 point, but does provide the basis on which such interoperability can be achieved.

181 **III. Definition of Terms**

182
183 The following section defines a few of the important terms for understanding this
184 document. Many of these terms have a variety of meanings, so they are defined
185 here in the narrow sense in which they are used throughout the DDI Version 3.0
186 Model. Please note that this list is being compiled as the work is being done – the
187 list of terms is in no particular order, although it will ultimately be replaced with a
188 more organized glossary.

189

190 *Raw Data*

191

- 192 • Literally what is collected in its collected form
- 193 • No recodes or constructed variables have been created, data have
194 simply been “captured”

195 *Microdata*

196

- 197 • Individual level data (whether that individual is a person, place or
198 thing)
- 199 • Variables providing identification of the record or its relationship to
200 another record may have been added
- 201 • Recodes or constructed variables may have been added
- 202 • Raw data elements may have been omitted

203

204 *Aggregate Data*

205

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

212 *Study*

213

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

219 *Concept*

220 • A definable idea or characteristic of the unit of analysis

221 *Representation*

222 • What is created when you measure a concept

223 *Variable*

224 • A specific expression of the representation of a concept

225

226 *Single Conceptual Model*

227

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

231

232 *Technical Implementation*

233

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

238

239 *File*

240

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

242

243

244

245 *Instance*

246

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

250

251 *Lower in the Model*

252

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

262

263 *Design Rules*

264

- 265
266
267
268
269
270
271
- 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.

272

273 *Functional Type*

274

- 275
276
277
- 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.

278

279 *Class*

280

- 281
282
283
284
285
286
287
- 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.

288

289

290

291 *Instance*

292

- 293
294
295
296
297
298
- 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.

299

300

301 *Reusable Classes*

302

- 303
304
- Reusable classes are those that can and do show up in multiple modules; Other Material, Universe, Citation, Notes, etc.

305

306 *Module*

307

- 308
309
- 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

310 the upper levels of DDI Version 2.0 (Document Description, Study
311 Description, File Description, Data Description, and Other Materials).

312

313 *Basic vs. Specialized Modules*

314

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

325

326 *Study Unit*

327

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

336

337 *Data Set*

338

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

341

342 *Human Readable*

343

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

348

349 *Machine Actionable*

350

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

354

355 *Persistent vs. Dynamic Information/Material*

356

357

358

359

360

361

362

363

364

365

366

367

368

369

Local Overrides

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

- 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.

- 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.

388

III. Life Cycle Metadata and Implications for the DDI

389

390

391

392

393

394

395

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.

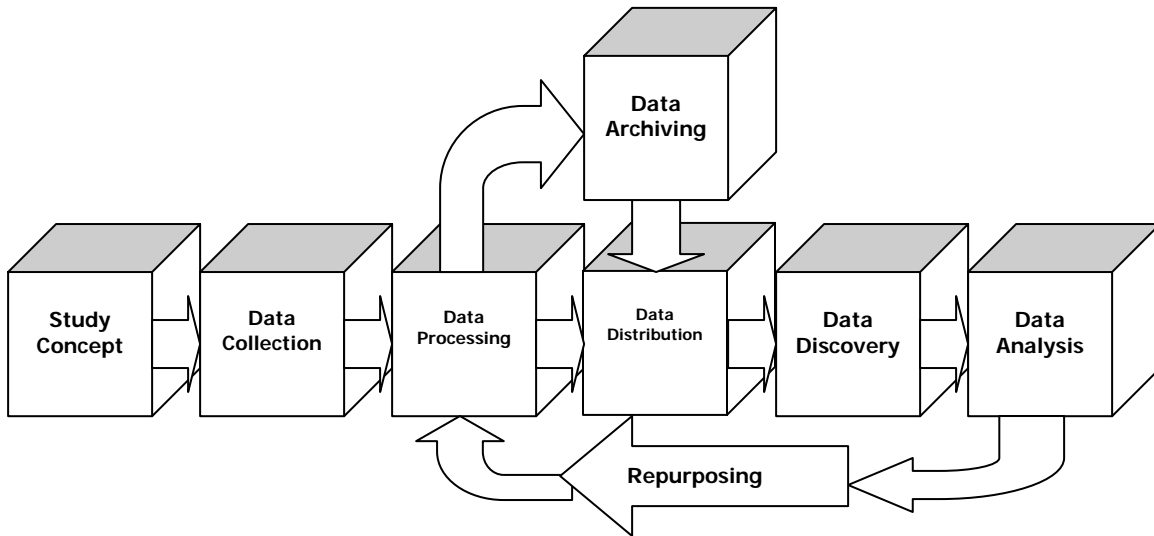


Figure: Combined Life Cycle Model

396
397
398

399 **A. DDI Instances and the Life Cycle**

400

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

408

409 For DDI Version 3.0, it is conceivable that the conceptual design of a study would
410 be marked up in DDI, and that as the study went through the life cycle, the DDI
411 instance documenting it would be updated in a sequence of versions: typically,
412 one for each stage of the life cycle. This requirement leads to the modular nature
413 of the DDI Version 3.0 design: as metadata are added, additional modules are
414 added, keeping the development of a DDI instance over time more
415 comprehensible, and making it easier to find and process the various parts of the
416 metadata which are of interest.

417

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

423

424 Note that what is versioned in DDI Version 3.0 is the set of metadata about a
425 particular study or group of studies, from a particular agency. It is *not* the DDI
426 XML instance. There is no assumption that DDI instances will be maintained:

427 they can – and often are – used as a transient mechanism for the exchange of
428 metadata which is persisted inside some other application or system.

429

430

431 ***B. Repurposing of Data***

432

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

441

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

452

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

458

459

460 ***C. Two Uses of the DDI: “Simple” Cases versus “Complex”*** 461 ***Cases***

462

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

470

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

474

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

481

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

489

490 **1. Simple Case**

491

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

499

500

501

502 The key criteria are:

503

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

506

507

508

509

510

511

512

513

514

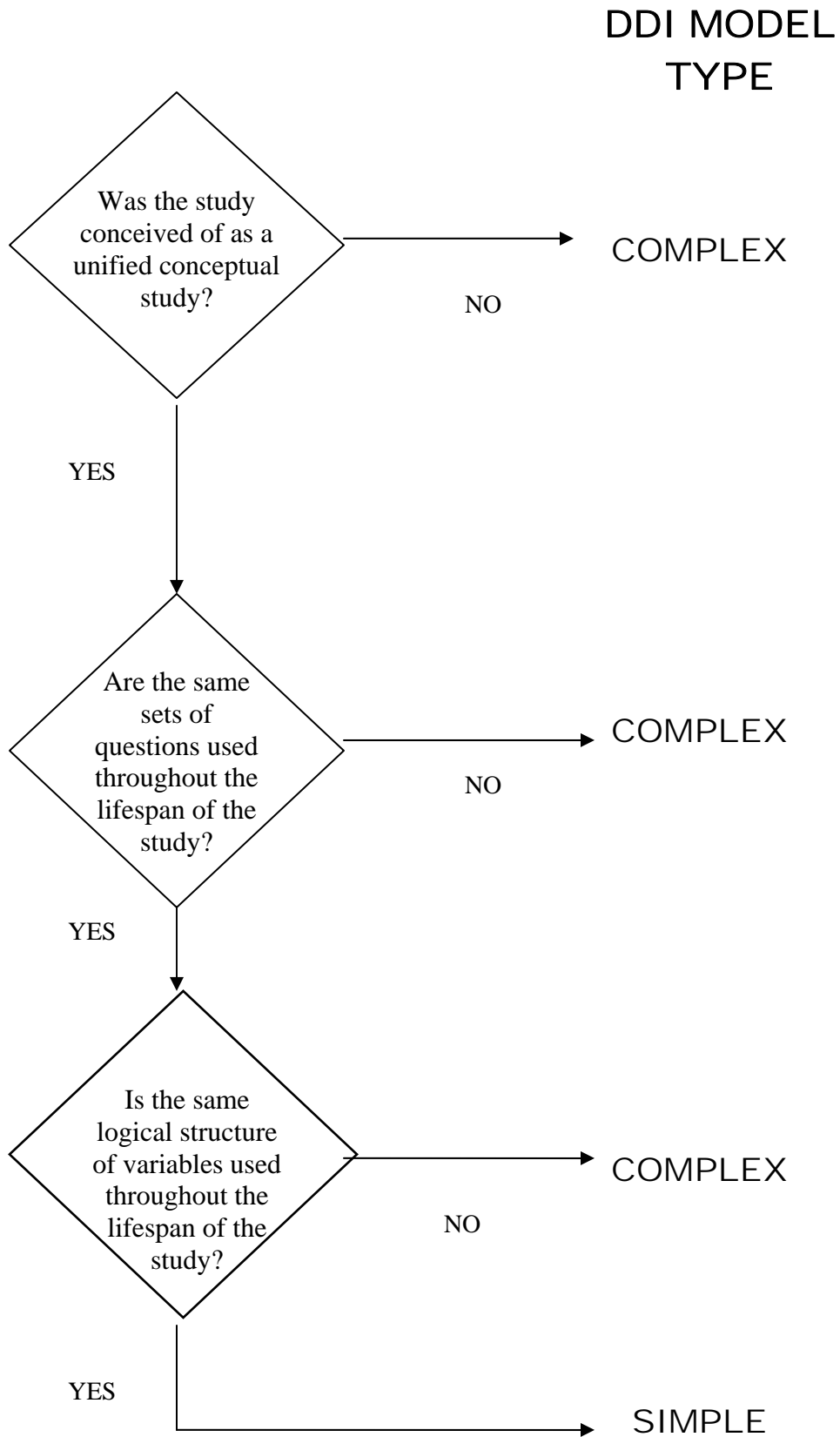
In the case that the creator of the XML does not choose to use any grouping

module (if these modules are not supported by local systems), then a second

515 XML instance must be created and any information on the relationship between
516 the two instances will be restricted to human actionable sections of the metadata.
517 Machine actionable relational information will be lost.

518

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



523 IV. Migration and Modularization Design

524

525 **A. Migration of 2.0 Elements**

526

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

533

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

541

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

548

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

552

```
553 <datac:var>...</datac:var>
```

554

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

557

558 **B. Modular Design**

559

560 The design of the DDI Version 3.0 allows greater flexibility in combining various
561 modules within a single wrapper to describe a single data file, a related group of
562 data files, or a related group of studies. It also allows software developers or
563 users to select which modules of information they can handle and to ignore
564 modules outside of their capabilities.

565

566 **1. Goals for Modular Design**

- 567
- 568
- 569
- 570
- 571
- 572
- 573
- 574
- 575
- 576
- 577
- 578
- 579
- 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.)

580 **2. Design Rules**

581

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

584

- 585
- 586
- 587
- 588
- 589
- 590
- 591
- 592
- 593
- 594
- 595
- 596
- 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.

597

598

599 **V. Simple Instances**

600

601 **A. Mapping from Version 2.* to Version 3.0**

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

604

605

Version 2	Description	Version 3
1.0	Document Description: Citation of the XML Instance / Content Citation of the Source documents	Instance / 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

606

607

608 Notes:

609 1.0 The Archive module will hold all the information specific to the archive
 610 including holdings information and file locations. The Instance and its
 611 various classes (Other Materials, Notes, Universe, and Citation) will hold
 612 the remaining material.

613 2.0 The materials currently in the Study Description are split between the
 614 Concept Module and the Data Collection Process Module roughly along
 615 the lines indicated in the table.

616 3.0 The Physical Data Structure Module contains the detailed record structure
 617 information and location information while the Physical Instance Module
 618 contains information on the gross file structure.

619 4.0 Most of the material in Data Description will move to the Logical Data
620 Structure Module with the exception of the first three items listed under
621 Variable. Question information will become part of the Instrument section
622 of the Data Collection Process, Location becomes part of the Physical
623 Data Structure (similar to the current location map section), and summary
624 statistics will move to the Physical Instance module.
625

626 **B. Functional Types**

627
628 The modules themselves are organized into “functional types”, which is a short-
629 hand functional description, characterizing their contents. Note that there is a
630 distinction between human-readable and machine-actionable metadata, which
631 has become an important distinction in the types of metadata documented within
632 the DDI.

633
634 Functional types are **not** structural in nature – all they do is describe the function
635 of a particular set of metadata. They are presented here only for the purposes of
636 clarifying the use of modules (which *are* structural) in the table below.
637

638 *Functional Type 1:*

639
640 Collection relationships of Functional Type 1 are provided by the archive or
641 holder of the metadata. These include archive-specific information and non-
642 technical grouping information such as common topic or common producer,
643 organization, funding source or principal investigator. Grouping at this level
644 doesn't carry technical implications for how the data are processed, but may
645 affect the upper level discovery search process within the archive.
646

647 *Functional Type 2:*

648
649 Collection relationships of Functional Type 2 have technical implications for how
650 to handle the data within the group. The Group Matrix identifiers are used in
651 describing grouping of Functional Type 2 in order to provide specific information
652 to programming applications. Examples of this are time-series, longitudinal
653 studies, repeated surveys, etc.
654

655 *Functional Type 3:*

656
657 Collection relationships of Functional Type 3 allow for grouping multiple data
658 collection actions and the use of multiple data collection instruments during a
659 defined data collection activity. For example, data may be collected from a group
660 of students, their parents, and their school within a specific time period using
661 three separate collection tools.
662
663

664
665
666
667
668
669
670
671
672
673
674

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.

675 **C. Module Descriptions**

676
677
678
679
680

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.

681
682
683
684
685
686

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
Instance	Contains top level Citation and Universe information; Provides structural map for modules included in the instance	Contains all other modules
Archive	Describes all archive specific information. Originally a basic module for archive identification and access restrictions. May be extended locally to cover processing management.	
Group	Collection of Functional Type 3 modules that may exhibit specified relationships that have technical processing implications for accessing and analyzing the data	Uses technical specification matrix to identify relationship types along 6 parameters [see Grouping, below]
Study Unit	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 including Question Scheme and Instrument.	
Logical Data Product	Describes the logical content of the data product	Contains Variables and NCubes
Variables	Describes the variable concept and structure; describes variable groups	LINKS from Variable to question and concepts.
NCubes	Describes the construction of nCubes from variables; describes nCube groups	LINKS to Variables used in the NCube
Physical Data Product	Describes the structure of a data store in terms of a record structure. With the use of dataset can also contain the data described by the XML instance.	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 (number of records, stummary statistics, etc.); allows for subsets of full records to be sotred without creating new physical data product description.	
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

687
688
689

690 VI. Multiple Files

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

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

702

703 Note that there is no necessary relationship between “simple” and “complex”
704 uses of the DDI and the number of files. A “simple” DDI instance could describe
705 multiple files. A “simple” DDI instance could describe multiple data files. An
706 example would be a description of a population census where there are multiple
707 data products or data products of the same type subset and stored in multiple
708 data files. A “complex” DDI instance could describe a single data file. An
709 example would be the creation of a single integrated data file from multiple
710 studies, such as the Integrated General Social Survey.

711 VII. Comparison

712 Comparison is an area in DDI 3.0 that will continue to develop. Consensus was
713 reached between the SRG and the Comparison Working Group to focus on
714 comparison of concepts, questions, and variables. Additional work will be
715 required to develop comparison of various methodologies and data collection
716 processes. Comparison in a broad sense, takes place between two or more
717 study units as either comparison-by-design or ad-hoc-comparison. DDI 3.0
718 allows for either method.

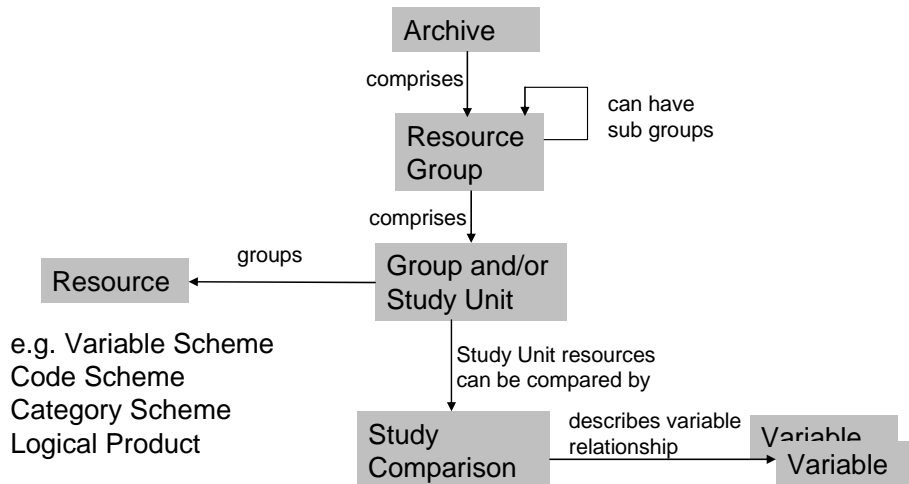
719

720 Comparison-by-design can be encoded as inheritance from a base structure
721 (concept, question, or variable), or through use of a more detailed item-by-item
722 comparison structure. Ad-hoc-comparison must be done using the comparison
723 structure. This structure provides for pair-wise comparison of individual concept,
724 question, or variable items. Think of it as creating a harmonized structure, where
725 each study unit is compared with the harmonized structure. Comparison between
726 study units works on the principle “If A=B and A=C then B=C.” The item level
727 mapping structure allows the user to define the relationship, for example
728 equivalency, parent-child, or relationship formulas.

729 VIII. Grouping

730

731 Below is a UML class diagram showing the DDI Version 3.0 model for the
732 grouping structure of “complex” cases.



733
734
735

736 **A. Overall Structure**

737

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

741

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

745

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

749

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

754

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

760

761 **B. Classes**

762

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

765

766 **1. Instance**

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

769

770 **2. Group**

771 A Group is a class for both formal and informal grouping of StudyUnits or other
772 Groups. This class contains a set of required properties (time, instrument, panel,
773 geography, data sets, and language), which identify the relationship, if any,
774 between a Group's child StudyUnits or Groups. Each of these properties contains
775 a single value that identifies the nature of the relation. For groups with no formal
776 relationships, each of the properties is assigned a value of "x0" (where x varies
777 by the property). A set of tables and flow charts to determine the values of these
778 six properties can be found in "Data Documentation Initiative (DDI) Technical
779 Specifications, Part II: High-Level Documentation, Appendix Two." In addition to
780 these properties, this class also contains common metadata and information
781 about comparisons and post-hoc variables. Metadata are inherited down the
782 hierarchy of the grouping, with the ability for children to override values locally.
783 Inclusion of children to the Group may be by reference or by direct inclusion (also
784 referred to as nesting).

785

786 **3. StudyUnit**

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

791

792 **4. Study Conceptual Classes**

793

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

799

800 For example, if a StudyUnit is contained in a Group, the Data Collection class of
801 the StudyUnit inherits from the Data Collection class from the Group. The Data

802 Collection class of the Group may contain a set of basic questions. The Data
803 Collection class of the StudyUnit would inherit these questions from the Group,
804 but would also be allowed to provide additional questions. In addition to the
805 classes mentioned above, the StudyUnit also contains a Physical Data Instance.
806

807 **5. Comparison**

808 Groups can also contain the Comparison class. The Comparison class contains
809 information about the comparability of the children Groups and StudyUnits
810 contained in the parent class. This is the module where “virtual” post-hoc
811 variables and concepts could be described. Each comparison must contain a
812 reference to the concepts and variables of the Groups/StudyUnits it compares,
813 using the external key mechanism.
814

815 **C. Examples**

816
817 The following section provides samples showing the grouping of studies using
818 formal and informal Groups and a combination of both. Note that the XML
819 structures used in these examples are for demonstration purposes only, and do
820 not necessarily represent the XML in DDI version 3.0. You may wish to refer to
821 the description of grouping properties in "Data Documentation Initiative (DDI)
822 Technical Specifications, Part II: High-Level Documentation, Appendix Two" for a
823 more complete understanding of the examples given here.
824

825 **1. Informal Group**

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

```
828  
829 <Group id="A" time="T0" instrument="I0" panel="P0" geography="G0"  
830 datasets="D0">  
831   <DataCollection>  
832     <CollectionEvent>CommonCollector</CollectionEvent>  
833   </DataCollection>  
834   <StudyUnit id="1">  
835     <DataCollection>  
836       <Instrument>INST-A</Instrument>  
837     </DataCollection>  
838     <LogicalDataProduct>LDP-B</LogicalDataProduct>  
839     <PhysicalDataProduct>PDP-C</PhysicalDataProduct>  
840     <PhysicalDataInstance>PDI-Y</PhysicalDataInstance>  
841   </StudyUnit>  
842   <StudyUnit id="2">  
843     <DataCollection>  
844       <Instrument>INST-B</Instrument>  
845     </DataCollection>  
846     <LogicalDataProduct>LDP-A</LogicalDataProduct>  
847     <PhysicalDataProduct>PDP-D</PhysicalDataProduct>  
848     <PhysicalDataInstance>PDI-X</PhysicalDataInstance>  
849   </StudyUnit>  
850 </Group>
```

851

852 2. Formal Group

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

855

```
856 <Group id="A" time="T4" instrument="I3" panel="P4" geography="G3"  
857 datasets="D2">  
858   <DataCollection>All Common Collection Info</DataCollection>  
859   <LogicalDataProduct>Common Logical Data Structure</LogicalDataProduct>  
860   <PhysicalDataProduct>Common Physical Data Product</PhysicalDataProduct>  
861   <StudyUnit id="1">  
862     <Concept>  
863       <Universe>1990</Universe>  
864     </Concept>  
865     <PhysicalDataInstance>1990</PhysicalDataInstance>  
866   </StudyUnit>  
867   <StudyUnit id="2">  
868     <Concept>  
869       <Universe>1991</Universe>  
870     </Concept>  
871     <PhysicalDataInstance>1991</PhysicalDataInstance>  
872   </StudyUnit>  
873   <StudyUnit id="3">  
874     <Concept>  
875       <Universe>1992</Universe>  
876     </Concept>  
877     <PhysicalDataInstance>1992</PhysicalDataInstance>  
878   </StudyUnit>  
879 </Group>
```

880

881 3. Nested Formal Groups

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

886

```
887 <Group id="A" time="T2" instrument="I3" panel="P4" geography="G4"  
888 datasets="D4">  
889   <DataCollection>  
890     <ResearchInstrument>  
891       <Question id="Q1">Question1</Question>  
892       <Question id="Q2">Question2</Question>  
893       <Question id="Q3">Question3</Question>  
894     </ResearchInstrument>  
895   </DataCollection>  
896   <Group id="A1" time="T2" instrument="I1" panel="P4" geography="G4"  
897 datasets="D2">  
898     <DataCollection>  
899       <ResearchInstrument>  
900         <Question id="Q4">Question4</Question>  
901         <Question id="Q5">Question5</Question>  
902       </ResearchInstrument>  
903     </DataCollection>  
904     <LogicalDataProduct>Jan Logical Data  
905 Structure</LogicalDataProduct>
```

```

906         <PhysicalDataProduct>Jan Physical Data
907 Product</PhysicalDataProduct>
908         <StudyUnit id="A11">
909             <Concept>
910                 <Universe>Jan1999</Universe>
911             </Concept>
912             <PhysicalDataInstance>Jan1999</PhysicalDataInstance>
913         </StudyUnit>
914         <StudyUnit id="A12">
915             <Concept>
916                 <Universe>Jan2000</Universe>
917             </Concept>
918             <PhysicalDataInstance>Jan2000</PhysicalDataInstance>
919         </StudyUnit>
920         <StudyUnit id="A13">
921             <Concept>
922                 <Universe>Jan2001</Universe>
923             </Concept>
924             <PhysicalDataInstance>Jan2001</PhysicalDataInstance>
925         </StudyUnit>
926     </Group>
927     <Group id="A2" time="T2" instrument="I1" panel="P4" geography="G4"
928 datasets="D2">
929         <DataCollection>
930             <ResearchInstrument>
931                 <Question id="Q4">Question4</Question>
932             </ResearchInstrument>
933         </DataCollection>
934         <LogicalDataProduct>Feb Logical Data
935 Structure</LogicalDataProduct>
936         <PhysicalDataProduct>Feb Physical Data
937 Product</PhysicalDataProduct>
938         <StudyUnit id="A21">
939             <Concept>
940                 <Universe>Feb1999</Universe>
941             </Concept>
942             <PhysicalDataInstance>Feb1999</PhysicalDataInstance>
943         </StudyUnit>
944         <StudyUnit id="A22">
945             <Concept>
946                 <Universe>Feb2000</Universe>
947             </Concept>
948             <PhysicalDataInstance>Feb2000</PhysicalDataInstance>
949         </StudyUnit>
950         <StudyUnit id="A23">
951             <Concept>
952                 <Universe>Feb2001</Universe>
953             </Concept>
954             <PhysicalDataInstance>Feb2001</PhysicalDataInstance>
955         </StudyUnit>
956     </Group>
957 </Group>
958

```

959 **4. Mixed Groups**

960 This example shows an informal Group containing both StudyUnits and formal
961 Groups, for instance studies funded by United States Department of Housing and
962 Urban Development, grouped together. This group contains one StudyUnit, and a
963 formal Group representing the American Housing Survey:
964

```

965 <Group id="A" time="T0" instrument="I0" panel="P0" geography="G0"
966 datasets="D0">
967   <DataCollection>
968     <CollectionEvent>CommonCollector</CollectionEvent>
969   </DataCollection>
970   <StudyUnit id="1">
971     <DataCollection>
972       <Instrument>INST-A</Instrument>
973     </DataCollection>
974     <LogicalDataProduct>LDP-B</LogicalDataProduct>
975     <PhysicalDataProduct>PDP-C</PhysicalDataProduct>
976     <PhysicalDataInstance>PDI-Y</PhysicalDataInstance>
977   </StudyUnit>
978   <StudyUnit id="2">
979     <DataCollection>
980       <Instrument>INST-B</Instrument>
981     </DataCollection>
982     <LogicalDataProduct>LDP-A</LogicalDataProduct>
983     <PhysicalDataProduct>PDP-D</PhysicalDataProduct>
984     <PhysicalDataInstance>PDI-X</PhysicalDataInstance>
985   </StudyUnit>
986 <Group id="AA" time="T4" instrument="I3" panel="P4" geography="G3"
987 datasets="D2">
988   <DataCollection>Common Collection Info</DataCollection>
989   <LogicalDataProduct>Common Logical Data
990 Structure</LogicalDataProduct>
991   <PhysicalDataProduct>Common Physical Data
992 Product</PhysicalDataProduct>
993   <StudyUnit id="AA1">
994     <Concept>
995       <Universe>1990</Universe>
996     </Concept>
997     <PhysicalDataInstance>1990</PhysicalDataInstance>
998   </StudyUnit>
999   <StudyUnit id="AA1">
1000     <Concept>
1001       <Universe>1991</Universe>
1002     </Concept>
1003     <PhysicalDataInstance>1991</PhysicalDataInstance>
1004   </StudyUnit>
1005   <StudyUnit id="AA1">
1006     <Concept>
1007       <Universe>1992</Universe>
1008     </Concept>
1009     <PhysicalDataInstance>1992</PhysicalDataInstance>
1010   </StudyUnit>
1011 </Group>
1012 </Group>
1013
1014
1015

```

1016 IX. Survey Instruments

1017
1018 Elements describing the questionnaire content and structure have been moved
1019 from the variable element into a sub-module of the data collection process. This
1020 allows for a more coherent and richer description of the survey instrument and
1021 the means of data collection (face-to-face interview, mail out form, phone
1022 interview, CAI, etc.

1023

1024 Response domains, questions, and instruments are defined as maintainable
1025 objects so that they and their contents can be reused. This allws organizations to
1026 stroe and reuse questions from a question bank as well as supporting the
1027 development of larger community-wide question banks.

1028

1029 By separating questions from the variable content and referencing them, studies
1030 that have resulted in multiple logical ata product creation from a single data
1031 collection process (such as Census microdata and summary statistics files) can
1032 all reference the same question description, proving a certain level of
1033 comparability between two or more logical products.

1034

1035 The survey instrument sections currently created for DDI 3.0 do not include
1036 information on the devleopment process for the questionnaire or study, but
1037 working groups have already begun to explore what is needed for adding this
1038 material at a future date.

1039 **X. HTML Tagging**

1040

1041 A large number of text areas allow for XHTML structural tags to facilitate the
1042 presentaiton of paragraphs, lists, tables, etc. All structured text fields have an
1043 attribute to denote language of the text. Because of the ubiquity of XHTML and
1044 the consequent support provided for it in most development environments, it was
1045 felt that XHTML provided a better approach to formatting than a set of DDI-
1046 specific formatting tags. Only designated elemetns allow for XHTML tags and
1047 they are generally those that are intended to be human-readable as opposed to
1048 machine-actionable, and whose content may require structure in order to convey
1049 the intended information.

1050 **XI. Uniform Notes & Citations**

1051 ***A. Reusable Classes***

1052

1053 **File/Section ID, Citation, and Universe:**

1054 Version 2.0 of the DDI allows for the description of bibliographic citations,
1055 universe descriptions, other related materials, and notes at numerous and
1056 specific places throughout its structure. Version 3.0 has pulled these out and
1057 created uniform structures for each of these classes. The reusable classes are
1058 available in each of the modules and may be linked to any element within the
1059 module. This approach increases both the consistency of the structure and the
1060 flexibility for application of references to outside materials and internal notes. A
1061 more extensive and structured type identifier is used to assist the programmer
1062 and user in sorting through the information held in each class structure.

1063

1064 The Version 3.0 citation has been divided into three parts:

1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076

- 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]
- Coverage: This is the topical, spatial, 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].

1077 **B. Notes**

1078 The primary change in the use of notes is that they are now grouped together in
1079 a class that is available in each module of the DDI. Notes can be referenced from
1080 any element, providing a level of flexibility not available in Version 2.0. In addition,
1081 a set of types is being developed to identify specific types of commonly used
1082 notes to increase capabilities for uniform processing by software systems.
1083

1084 **C. Other Material**

1085 A single uniform structure for identifying, describing and pointing to Other
1086 Materials of all types has been developed and added to the model in a similar
1087 format to Notes. This includes a single class structure available in each module
1088 and the ability to point to a other material reference from any element within the
1089 module. In nested modules, Other Material contents can be inherited down the
1090 tree and referenced from lower modules.
1091

1092 **XII. Alignment with Other Standards**

1093
1094 In developing DDI 3.0, over 30 related standards were reviewed for relevance by
1095 the SRG. After review a list of primary standards was selected for careful
1096 consideration: These included:

1097		
1098	ISO/IEC 11179	Compliance with basic structures regarding data items and
1099		their conceptual basis
1100		
1101	Dublin Core	For easy exploration by external search engines and creating
1102		compatible record extracts
1103		
1104	SDMX	Ability to map from SDMX to a DDI Ncube and back again
1105		
1106	METS	To keep in mind its basic constructs

1107

1108 Metadater Share development information to allow DDI 3.0 to serve as a
1109 transport structure for metadata into and out of this structure