Identifiable objects: standard contents

A report to the Modeling Group from the Technical Committee – 2014-07-23

This document brings together the consensus from the Technical Committee plus other interested modelers and is based on discussions during TC meetings in June/July 2014. The agenda topics included for these meetings included: Identification (separate paper), Name/Label/Description, what objects are referenced, and Administrative metadata. Minutes from these meetings have been posted to the modeler’s site.

This document contains information on the following:

* Definition of what objects are referenced (i.e. have DDI Identifiers)
* Guidelines for determining the role of an object as a property or a relationship
* Standard objects for sub-sets of identifiable objects
* Administrative metadata for identifiable objects

# Guidelines for what objects are referenced

Consensus was to develop a set of general guidelines at the start since we don’t know how we’re grouping things yet in the new model. The idea is that we will tighten up the guidelines as we go and encourage modelers to be diligent about capturing the reasoning behind the decisions they make on this as they go.

If it makes sense as an object on its own, not just as a property of its ownership

* If it’s something that is managed over time
* If there are other items that make a reference to it

Modelers should document the reasoning behind other factors that influence decisions to make an object reference-able

# Guidelines for determining role as Property or Relationship

|  |  |  |
| --- | --- | --- |
| Object Type | Definition | Example |
| All objects | All UML objects within the DDI Object Library | Concept, StructuredString, xs:string, etc. |
| Properties | Objects whose content makes sense only in its relationship to its parent object. Properties may be simple primitives, extended primitives, or complex objects | Simple primitive: xs:string, xs:Boolean  Extended primitive: value (an xs:string with an xml:space designation)  Complex: StructuredString, CodeValueType |
| Identified objects (listed in “Relationship” in Drupal)  [Identified objects] | Objects which are managed over time or are items which must be referenced by another object | Concept, Category, Measure, Individual, etc. |
| Referenced objects | Objects which are not managed but which another item must make reference to | A code within a codelist has no individually managed existence, however, the individual code needs to be referenced by a category level statistic in order to relate the value of the statistic to the specific code within the code list |
| Managed objects | Objects that have meaning outside of their use and are managed over time (versioned) | Concept, Universe, Individual, Machine, Organization |

# Standard objects for sub-sets of identifiable objects

See “Names and Labels.pptx” for background information provided by Dan Gillman

## Background in DDI-L

In creating DDI 3.0, we inherited the idea of Name, Label, and Description from the DDI-C “var”. In DDI-C a variable had a required name attribute which generally contained a unique name used by a statistical package, a label used by the statistical package, and a text which allowed for further expansion of the label. Depending upon the software used to create the DDI-C instance and the conventions of the agency creating the document, name could be a duplication of the id and label content could end up either in the label or text depending upon its length, and text might contain a long label or a further description of the variable. Name was carried over in an Identifiable Type as a “local” form of identifier.

In DDI 3.1 a consistent set of objects was applied to objects that were intended to be managed within a registry. This included a standardized XxxName, Label, and Description where Xxx was the object being named (i.e. VariableName). The original Name in the Identifiable Type was changed to a UserID designating it specifically as a local identifier.

In DDI 3.2 this standard set of objects was applied consistently to all managed objects (maintainables and versionables within schemes).

During the usage of these terms in DDI the description of their usage has varied and there has been inconsistency in their applied usage. In addition, there have been changes in the ISO/IEC 11179 description of the use of Name, Label, and Description as they were included in DDI 3.1. For this reason, we asked Dan Gillman to provide a clear conceptual basis for further work in this area. (See “Names and Labels.pptx”)

## Discussion

Name and Label are synonyms and the DDI usage guidelines have been imprecise and variable over time. A Label is a representation of an object by a sign which denotes it. A Label can be further specialized as an identifier, locator, or designation. A DDI Description has been used a general description of intent, usage, or anything else we want to say about the object, as well as a definition (and sometimes both at once).

Labels

* Identifier
  + Definition: label intended as a means to dereference the labelled object
* Locator
  + Definition: identifier with known dereferencing scheme
* Designation - Label for concepts
  + Definition: representation of a concept by a sign which denotes it
  + Since a concept is a kind of object, then a designation is a kind of label
  + Similar usage as words in natural language
    - Denote meanings
    - Used in communication
    - If similar to Natural Language words, then can convey some meaning
  + Has kinds: term, appellation, code, and symbol
  + Term
    - Definition: linguistic designation for general concept
  + Appellation
    - Definition: linguistic designation for individual concept
  + Code
    - Definition: non-linguistic designation, but denoted by alpha-numeric string
  + Symbol
    - Any other designation

Definitions

* Definition – natural language statement of the meaning of a concept
* Review definitions above
* Each starts with a previously defined concept then provides differentia
* These are *intensional* definitions
* An *extensional* definition delineates kinds:
  + Example – Human teeth are incisors, canines, bicuspids, and molars
* Difference between terms and definitions
  + Terms are not necessarily nor usually statements
  + Terms do not necessarily convey meaning, or at least not precisely
  + Example –
    - *Unemployed,* as used by US Bureau of Labor Statistics
    - Unemployed has English meaning, i.e., not employed
    - Not the same as BLS meaning

## Conclusions

DDI requires the use of labels that serve as locators and designations as well as means of providing a definition and in some cases a description.

Locator:

* Structure: An xs:string with attributes identifying the reference scheme
* Definition: A non-translatable identifier within a known dereferencing scheme.
* Usage: This should be used when the managed object is managed within a specific system with a known dereferencing scheme.

Designation:

* Structure: An international string with the current features of a Label plus an attribute clarifying whether the content serves as a term, appellation, code, or symbol. Note that Label currently has TypeOfLabel, locationVariant, validForStartDate, validForEndDate, and maxLength.
* Definition: A linguistic sign denoting a general concept (term) or individual concept (appellation), or a non-linguistic sign (an alpha-numeric code or non-alphanumeric symbol) denoting a concept.

Definition:

* Structure: International string
* Definition: A natural language statement of the meaning of the concept. It may be intensional, starting with a previously defined concept and providing differentia, or extensional, providing delineating kinds (i.e. Human teeth are incisors, canines, bicuspids, and molars)

Description:

* Structure: Structured String
* Definition: Textual information that provides additional technical, explanatory, or usage information that informs the user in the use or interpretation of the object.

## Actions

1. Remove use of NameType and replace with a new object “Locator” (the use of a Label as a locator)
2. Change Label to “Designation”
3. Add Definition
4. Alter documentation of Description.

Determine membership of sub-groups of all managed objects. Possible sub-sets include the following and indicate which standard objects should be applied to the group:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sub-set** | **Locator** | **Designation** | **Definition** | **Description** |
| GSIM objects [Concept, Universe, Conceptual Variable, Represented Variable, Category, Classification] | X | X | X |  |
| Variable | X | X | X | X |
| Representations [Value Representation, ResponseDomain] | X | X | ? | X |
| NCube | X | X | ? | X |
| Questions [Question Item, Question Grid, Question Block] | X | X |  | X |
| Instruments | X | X |  | X |
| Processing Instructions | X | X |  | X |
| Quality Statement | X | X |  | X |
| Control Constructs | X | X |  | X |
| Interviewer Instructions | X | X |  | x |
| Geographic Structure | X |  | X | X |
| Geographic Location | X |  | X | X |
| Processing Events | X |  |  | X |
| Physical Structure | X |  |  | X |
| Record Layout | X |  |  | X |
| Agents | X |  |  |  |

Variable is a special case because of its interaction with statistical packages. However, it is defined by its component parts (concept, universe, representation, represented variable, derivation command, etc.) Consider its special need for:

1. Variable identification within a statistical system:
   1. SAS, SPSS, STATA, R name
   2. Column header in a csv or spreadsheet file for unit level data
2. Variable is an object that uses a specific “term” for display purposes

# Administrative Metadata

See AdministrativeObjects\_3\_2.xslx

This spreadsheet contains the set of Administrative metadata currently found in a DDI 3.2 maintainable and reference. The maintainable in 3.2 is an extension of versionable which itself extends identifiable. Therefore a maintainable contains all of the administrative metadata objects with the exception of the self-describing attributes “isIdentifiable” and “isVersionable”.

We made the following assumptions:

* Administrative metadata would be applied to all managed objects. Their application to the subset of Identifiable objects described as Referenced objects was not supported.
* The restriction sets (BaseIDType, DDIAgencyType, VersionType) will be removed and content of ID, Agency, and Version will be any character except ":" which is reserved as a section separator in the URN.
* Require the separate objects (ID, Agency, Version) and have URN optional (as it can be constructed from parts) - forces people to specify all three objects, also allows people to relate those separate objects to other vocabularies.
* DDI URN - Moving to the Canonical URN (urn:ddi:Agency:ID:Version)

The columns include the following information:

1. Recommendation: This is the action recommended by the Technical Committee
   1. “??” indicates that there is need for additional discussion and a recommendation has not been made
2. Notes: Discussion notes indicating the reason for the recommended action or points that need clarification.
3. Object (with sub-objects): A nested listing of the DDI 3.2 content
4. Type: The type of object
5. Cardinality: DDI 3.2 cardinality
6. Documentation: Field level documentation from the DDI 3.2