Summary of Recommendations from the Technical Committee to the Modelers Group

Wendy Thomas, TC Chair 2014-10-22

# Identification

|  |  |
| --- | --- |
| **Proposal** | **Action** |
| All objects have an Agency Identifier, Item Identifier, and Version Identifier (referred to as A:I:V) |  |
| Structure of each section of A:I:V is a character string excluding the use of colon (:) or space |  |
| Containership does not affect identity |  |

**Reasoning:**

See Identification Paper on Modelers page.

**Outstanding Issues:**

**We require a decision between the use of Strategy 1 and Strategy 2 alone or using both. *The groups doing the transformations are responsible for determining the approach used.***

XML Serialization: When referencing an identified item to create a semantic relationship, the Agency Identifier, Item Identifier, and Item Version shall be specified to create the relationship reference. The Type of Item may be included in the reference.

RDF Serialization: When referencing an identified item to create a semantic relationship, the URI of the item will be used. To enable interoperability between the RDF and XML representations, additional practices will be needed when creating relationships in RDF to determine the identity of an item.

Strategy 1

DDI will be publishing a practice for creating concise bounded descriptions [1] (CBD) and named graphs of an identified item or set of items. This document could include an additional requirement.

When referencing an identified item via a URI that will not be included in the CBD or named graph, still include the Agency Identifier, Item Identifier, and Item Version as properties of the URI in the CBD. Additionally, the RDF type information about the referenced item may be included.

Strategy 2

“In OWL 2 a collection of properties can be assigned as a key to a class expression. This means that each named instance of the class expression is uniquely identified by the set of values which these properties attain in relation to the instance.”  - Owl 2 primer

To be additionally formal, the Agency Identifier, Item Identifier, and Item Version can be added to the class expression of the identified item with the owl HasKey to create a composite key. This would uniquely identify the named instances.

See Owl 2 keys [2]

Example:

DDIItem HasKey ( :AgencyId : ItemId :ItemVersion )

# Properties and Relationships

|  |  |
| --- | --- |
| **Proposal** | **Action** |
| Property: All objects that can be described by types in Primitive or Extended Primitive | Review and correct content |
| Relationships: All other objects | Review and correct content |

**Reasoning:**

See Report to Modelers

**Outstanding Issues:**

Creating new extended primitives – We need guidelines for candidates for new extended primitives. Content groups should just enter as a relationship but note in the description of the object that this may be a candidate for an extended primitive. Extended primitives should be created by modelers.

* A Property does not have an identifier therefore no other object can create a relationship to it
* A candidate for an identifiable object includes all objects that are designed to be managed and versioned over time OR must support the ability to create a relationship, for example an instruction in a question may need to be attached to a specific Code within a Code Representation.

# Relationship Objects (Managed Objects)

|  |  |
| --- | --- |
| **Proposal** | **Action** |
| All objects described as Relationships have Identification and Administrative Objects attached to them (See Administrative Objects section 5: Administrative Objects in Identified Objects for recommended objects based on the list of current 3.2 administrative objects.) | Create extension base add to objects |
| All objects may contain the DDI 4 equivalent of Name, Label, and Description. If used they are entered as properties following a standard name for the property with a pre-specified extended primitive type. (See section on *Name, Label, and Description* for proposed objects) | Create the extended primitives to provide the type for each and enter into Drupal as requested by Content groups. |

**Reasoning:**

See Administrative Objects Section 4: Administrative Objects in Identified Objects for recommended objects based on the list of current 3.2 administrative objects.

**Outstanding Issues:**

1. Is there a sub-set of Relationship objects that are not “managed” independently? That is, they exist as objects with identification because they are complex in structure (not a simple or extended primitive)and/or must be identifiable in order to attach or relate to another object, but do not make sense as independent objects. Example: A Code within a CodeList has a relationship to a Category and an associated Value.
2. If a sub-set is defined, in addition to the identification structure, should they
* contain administrative objects?
* allow for the use of the DDI 4 equivalent of Name, Label, and Description?
1. Do Relationships need to support the current administrative objects late bound and late bound restrictions? (Note these are the only pieces of administrative data that were not dropped from the DDI 3.2 list (See Administrative Objects Section 5: Administrative Objects in Reference Objects .)

# Name, Label, and Description

|  |  |
| --- | --- |
| **Proposal** | **Action** |
| Change DDI 3.2 Name to 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.
 | Create as an Extended Primitive – Enter usage as requested by Content Groups |
| Change DDI 3.2 Label to 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.
 | Create as an Extended Primitive – Enter usage as requested by Content Groups |
| Add Definition * Structure: International string
* Definition: A natural language statement of the meaning of the concept. It may be intentional, starting with a previously defined concept and providing differentia, or extensional, providing delineating kinds (i.e. Human teeth are incisors, canines, bicuspids, and molars)
 | Create as an Extended Primitive – Enter usage as requested by Content Groups |
| Change documentation for 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.
 | Create as an Extended Primitive – Enter usage as requested by Content Groups |

**Reasoning**

It was generally agreed that these should simply be available for all identifiable objects with the use to be determined by the content groups. Some guidance has been provided in the Report to Modelers, Standard Objects for Sub-Sets of Identifiable Objects. See “Names and Labels.pptx” provided by Dan Gillman for additional background information.

**Outstanding Issues:**

1. If a differentiation between managed and other identifiable objects, should these be available for use by objects determined to be in the “non-managed” subset.
2. If more consistency is needed in terms of availability some guidance has been provided in the Report to Modelers, Standard Objects for Sub-Sets of Identifiable Objects