

Recommendations on Identifiable, Annotation, and Complex Data Type Usage

Structural Changes:

1. Move all properties currently in AnnotatedIdentifiable to Identifiable
 - a. The separation of the content Identifiable from the current properties in AnnotatedIdentifiable is a carryover from DDI-Lifecycle where classes were Maintainable, Versionable, or Identifiable and Identifiable inherited its agency and version number from its parent Versionable. This situation no longer exists.
 - b. There is no use case where a class is not independently Versionable and therefore would benefit from the availability of the properties regarding versioning
 - c. The two properties isUniversallyUnique and isPersistent refer to the id property in Identifiable and are relevant wherever the current Identifiable properties are used
2. Change Annotation from Identifiable to a ComplexDataType
 - a. The annotation of a metadata class implies a one-to-one relationship between the object and its annotation. By definition it is not reusable.
 - b. Retaining Annotation as a ComplexDataType allows for the use of this structure to provide annotation for a related/external object such as an external data file (See ExternalMaterial)
3. Retain the extension base type AnnotatedIdentifiable using the extension base Identifiable but move hasAnnotation from a Relation to a Property with the cardinality 0..1
 - a. It is desirable to retain a means of easily identifying classes which can be annotated
 - b. It reduces the number of identifiers required for management of a DDI4 instance

Guidance for Usage of Identifiable, AnnotatedIdentifiable, and Complex Data Types:

A first order class (not complex data type or enumeration) is always Identifiable to allow it to be bound into syntaxes. If something could be a complex data type as opposed to a class then the following considerations apply:

- The following cases are first order classes:
 - The class whose content may serve as a Member of a Collection (realizes Member)
 - The class whose content may be related to multiple classes
 - The class has an independent existence
- A class should be AnnotatedIdentifiable if IN ADDITION the following is true:
 - The class contains intellectual content that needs to be discoverable in its own right
 - Example: A CodeList may need annotation as would the Categories that are being coded. These are concepts with intellectual content. However the “Code”

which links a Value to a Category only has intellectual meaning within the context of the CodeList and should be simply Identifiable.