

## Functional Views for Business Modelers

Functional Views are sets of DDI classes organized to support a particular purpose, such as describing a data capture, describing a data set, creating a list of organizations and/or individuals, etc. Classes are instantiated as objects in instances (like XML or RDF) of the model bindings. Functional Views are implemented as a set of references to unmodified classes as described in the DDI Library (Drupal). Individual classes can be (and are) reused in multiple Functional Views. If some properties or relationships are available on a class, but are not needed for particular Functional View, they can be documented. Modelers should take care to make sure that the classes they need do not already exist in the library, and to consider the impact of any changes they make to existing classes. The names of classes and their properties and relationships should not be specific to a Functional View. And even though Drupal organizes the library into packages, all names of classes should be unique within the entire library.

Functional Views are documented as discrete parts of the DDI Model, intended to be used as self-contained modules (realized as UML packages on the model level). For common functions in DDI, the Functional View should provide enough information to support an implementation (with the RDF or XML, of course). The entire library is also documented and bound to syntax deliverables, which will include all of the classes used in any Functional View.

Functional Views are used in the generation of “convenience” XML schemas and RDF vocabularies, which contain places only for the classes which exist in the Functional View. This does not mean that there cannot be relationships with classes outside the Functional View – only that they will not be found in the XML instances or graphs generated from the Functional View. Such relationships and dependencies should be considered when deciding which classes to reference from the Functional View. Note that the versioning mechanism being currently considered would version external releases as a monolithic whole, so when DDI 4.1 is released – after the release of DDI 4.0 – every Functional View would change to be 4.1.

One class which is intended to be available within all Functional Views is named DocumentInformation and is found in the Utility package. All relationship classes are recommended for inclusion: Annotation (Agent types), Access, FundingInformation, ExternalMaterial, and Coverage (TemporalCoverage, TopicalCoverage, SpatialCoverage, BoundingBox). If one of these classes is not to be included in the Functional View, there must be a good justification. It is not needed for all types of implementations, however, and so should be optional.

Functional Views are intended to be specific to common functions or purposes, and the DDI Alliance will be releasing a number of them, as DDI itself can be used for many different functions. The initial set will include such functions as describing surveys and other types of instrumentation for data collection, description of logical and physical data sets, describing codebook metadata, describing individuals and

organizations (“Agents” in DDI terminology), documenting methodology, and classification management. This list of functions will grow over time.