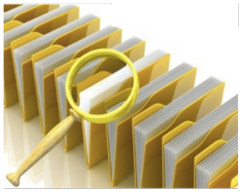
Drupal Conventions

Current Teams

Sprints

DDI 4 Drupal site


DDI 4 Documents

Explain DDI 4 to me

Project Management

Content Backlog

Past Teams

Links

The actual drupal system: http://lion.ddialliance.org/

Issues list for the technical system: https://github.com/ddialliance/lion/issues

Content Capture

The content capture is organized around packages, and will consist of those with domain knowledge and should include at least one person with modeling expertise. There should be a designated individual (the 'package leader') who is responsible for making sure that the discussion and structure of the objects and their relationships are captured in the Drupal content management system. Guidance on input is summarized on the entry form and a fuller description is below.

The package leader will also be responsible for ensuring that comments for changes to the initial description are reviewed and where these cannot be resolved, that a meeting is convened of those who have contributed, plus other interested parties to come to a resolution.
Once reviewed, the status on the objects should be flagged as Content Review/Completed and passed to the Modeling Team for Review.

Technical Committee Documentation leaders Wendy Thomas and Jon Johnson can be called upon if needed to help resolve any outstanding issues.

It would be helpful for a link to either the list of package leaders or an email contact link

Modeling Group

The role of the Modeling Group is to evaluate the Reviewed Content and suggest changes and amendments. Each package should have a nominated 'package leader' who will be responsible for ensuring that the package is reviewed and collating comments and where necessary liaising with the Content package leader to iterate the package back through the Content Group, (Modeling Review Rejected) or to be passed to the Technical Committee by changing the status to Modeling Review / Complete.

Technical Committee Modeling leaders Arofan Gregory and Achim?/Dan?/Johan? can be called upon if needed to help resolve any outstanding issues.

It would be helpful for a link to either the list of package leaders or an email contact link

Members:

- Arofan Gregory
- Dan Smith
- Joachim Wackerow
- Johan Fihn
- Larry Hoyle
- Brigitte Mathiak
- Thomas Bosch

Packages

Contributors and Package Leaders Needs updating and modeling group leaders assigned

Package	Package Leader	Contributors	Modeling Group Leader
Core	Technical Committee		
Foundational	Wendy Thomas	MV,WZM,JL,JJ,BR,SMc,DG,GD, OR	
Simple Instrument	Barry Radler	WZM,GD,	
Simple Data Description	Ørnolf Risnes		
Simple Codebook			
Agent	Johan Fihn?	WT,HL,JJ	
Discovery	Jay Greenfield?		

Status Codes

1. proposal in progress
2. content review in progress
3. content review partially complete
4. content review complete
5. model review in progress
6. model review partially complete
7. model review complete
8. released
9. rejected

Comment: Stage and Status have been collapsed into this one list.

Drupal Content Guidance

Field	Description	Example
Object Name	This is the name of the class in the model, and should follow the naming rules for elements, capitalized first letter. Object names should be unique. CamelCase	ProcessingEvent, Concept

Is Abstract	An abstract class is one which exists only to provide common properties and relationships to a group of related sub-classes, which inherit all properties and attributes of the parent abstract class. The sub-classes are extensions of the abstract class, but the abstract class itself only ever has existence through one of its sub-classes.	ControlConstruct is abstract, because it would be manifested as an If Then Else, a Question Construct, a Statement Item, etc.
Extends	This field should contain the name of an object which functions as a parent object, from which the extending class will inherit all properties and relationships. An extending object may modify its parent object by adding properties and relationships, altering cardinalities on inherited properties and relationships, etc. The children of abstract objects will always extend their parent abstract object, but non-abstract objects may also act as the parent objects of extending ones.	The IfThenElse object extends the abstract ControlConstruct object
Version	This is the version number of the object, using a two-part version number. The first number designates the "version" of the DDI model, and the second is a counter for each reviewable iteration of the object definition. Currently, the model version is "0". *sExample:* 0.1 [The first reviewable iteration of the model for the October 2013 Dagstuhl Sprint]	
Contributors	This field should be a list of all the people who have materially contributed to the definition of the object, so that people continuing the work later will know who to ask for explanations, etc. when questions arise.	
Definition	This is a **non-circular** definition of the object, which may include references to other objects in the model, but should also permit users to understand how to distinguish such an object in reality. This field is not a description, but a definition, and if taken from another source, the citation of that source should be provided.	An IfThenElse object describes the conditional logic in the flow of a questionnaire or other data collection instrument, where if a stated condition is met, one path is followed through the flow, and if the stated condition is met, another path is taken.
Example	This field should contain one or more examples which illustrate what the real-world might be. If possible, a link or citation to the actual objects should be provided.	An example of the Survey Instrument is the print form administered by the Flinders University of South Australia Centre for Aging Studies, to collect data for Wave 6 of the Australian Longitudinal Study on Aging – see www.alsa.wavesix.survey.pdf

Synonyms	A list of alternate terms by which the object is typically described, including the citation of the source of such names if relevant. This field is useful when different terms are commonly used for the same object, and agreement among group members of the formal name is difficult to reach.	For the SurveyInstrument object, a synonym would be "Questionnaire".
Properties	These are the attributes of the object which take a literal value. The property is ascribed several characteristics: **Name**: The name of the property. camelCase should be initial lower caseThis needs agreeing and a good example(s) ***Description:*** This is a description of the property which allows people to understand the semantic associated with the property. For the property "local ID", the description might say "The identifier of the object within a local system, as indicated by the system property of the same object." ***Datatype:*** This field should indicate the type of the property, using one of a set of terms taken from a controlled vocabulary based on the allowed representation types for variables found in DDI Lifecycle: "String" (indicate if structured and/or international), numeric (indicate if Integer, Big Integer, Long, Short, Decimal, Float, Double, Count, Incremental) , Date, Time, DateTime, Identifier, URN, Entity. Indicate if a value is coded. Ranges are allowed where applicable (for numerics and dates). ***Cardinality:*** The allowed numbers of this property for an instance of the object, notated as 0..1, 1..1, 1..n, 0..n, etc.	
Relationships	Relationships **are directional**, and are described as part of the object which is the source (thus, if a Variable references a Question, the property is described in the description of the Variable object, and not in the description of the Question object.) In a case where the object being described is an extension of another object, inherited properties do not need to be repeated unless one or more of their characteristics is being changed, in which case the relationship must be repeated with the altered characteristics stated. Relationships are described using a set of characteristics:	

Name: The name of the relationship. For example, a Variable having a relationship with a Population will name the relationship "measures", where the Variable object is the source, and the Population object is the target. **T**he name of the relationship should be camelCase, initial lower caseThis needs agreeing and a good example(s)

Relationship Type: The type of the relationship, taken from a standard list, including:

- *Aggregation:* A relationship in which the target is a component part of the source object, but has an **existence independent** of the source object (i.e. if the parent is deleted, the child still exists). The relationship between a Variable and a Question is an aggregation.
- *Composition:* A relationship in which the target is a component part of the source object, and whose **existence depends** on the existence of the parent object. The relationship between a Person object and a Passport object is a composition. (i.e. if the parent is deleted, the child is also deleted).
- *Simple Association**:*** an otherwise described relationship
- *Extends:* Where the relationship is one that extends another object ***e.g. Inheritance*** (there is no dependency)

Target Object: The name and ID of the object within the DDI model with which acts as the target of the relationship.

Description: The description of the relationship, to be used in the documentation of the model.

Source Cardinality: The possible number of instances of the target object with which the source object may have this relationship. For example, a Person object may have many "owns" relationships to Passport objects, but a Passport will have an "owns" relationship with only one Person: when the Person is the source and the Passport the target, the source cardinality will be "1..1" and the target cardinality will be "0..n".

	Target Cardinality: The possible number of target instances with which the source may have a relationship. For example, a Person object may have many "owns" relationships to Passport objects, but a Passport will have an "owns" relationship with only one Person: when the Person is the source and the Passport the target, the source cardinality will be "1..1" and the target cardinality will be "0..n".	

Rules for Drupal consistency checking

- change status to "Content review completed" or "content review partially completed"
- use verb as prefix for relationship names: e.g. "isPart" or "hasPart" instead of just "part"
- add datatype when missing
- move text from "Description deprecated" to "Description"
- remove empty properties and relationships
- use the semantically appropriate type of relationship: association, composition, neither (simple)
- use lower camel case for associations and properties
- use upper camel case for objects
- make name, label and description into properties
- use "Name" type for Name property
- use "Label" type for Label property
- use "StructuredStringType" for Description property
- make "reference" properties (from 3.2) into associations
- use new status/stage to indicate the object has been reviewed, etc.
- Add to the log: "This has been reviewed for consistency - Toronto"