DDI4 Prototype Definition: Functionality, Information Coverage, Deliverables

and Timeline

# Overview

This document records the work done in the October 16-20 2017 sprint in the Prototype Definition group. The contents of this document were presented at the beginning of the sprint the following week.

This document is intended to form the basis of an agreed scope, schedule, and set of deliverables for the upcoming release in June 2018 (called “the Prototype” as of this writing, although the name may subsequently be changed). It is understood that some of what is described here may change, but this should be done according to a managed process, so that all parties are aware that a decision to make a change has taken place. It is hoped that this document will be agreed by the DDI Alliance management, and can serve as the basis for a more detailed project plan moving forward.

The document describes the use cases to be supported, the portions of the model which will be required to support these cases, a schedule for the modelling and production work – including quality assurance and packaging – and a reasonably detailed list of deliverables.

## Use Cases

While we currently have no specific plans to distribute use cases for the prototype, we need to know what functionality we consider in and out of scope, and agree on where these lines ae to be drawn. To facilitate this, the group elaborated the following list of use cases.

It has been noted that some of these might be usefully fleshed out to use as marketing materials, training materials, etc. This group did not address those options, as they were not considered to be part of the prototype deliverable set. That discussion could be further pursued.

Key for Use Cases and Content Scope:

+ denotes that if the item is not included in the prototype, this is not a “show stopper”

\* denotes that the item is high priority for the prototype

* Data Capture\*
  + Simple questionnaire\*
  + Simple protocol \*
  + Obtaining register data (official data)\*
  + Internet source (e.g. web scraping) - requires workflow and protocol for information captured about the web scraping event. Transformation path from source to dataset
  + Streaming data source (e.g. smart meter data, IOT) - what is limited due to requirement of qualitative
* Publish Common Data Elements\*
* Describe a dataset\*
  + Streaming data source (e.g. smart meter data, IOT) - what is limited due to requirement of qualitative
  + Describe slices of a dataset
* Describing register data (e.g. RAIRD) - Register Utiliser Tool (Swedish Research Council)\*
* Statistical Classification\*
* Data Transformation Processing\*
  + Transformation of data from point of capture to resultant dataset\*
  + Verification processing during archive ingest\*
  + Confidentialization\*
  + Imputation\*
* Describe a simple study\*
  + Including discovery and dissemination (ability to populate the DISCO RDF vocabularty currently published by the DDI Alliance)\*
  + Ability to create Intentional or *ad hoc* grouping of studies aka DDI 3.2\*
* Archiving a study (description, provenance, ownership)\*

# Content Scope

This section addresses those portions of the model – either existing or near complete – which will be needed to support the use cases listed above.

### Study Level Information

(This should support both 2.5 and 3.2 DDI versions from a content perspective. Current content is focused on Codebook functionality.)

* Citation\*
* Coverage\*
* Agent description (Individuals, Organizations, etc.)\*

### Data Description

* Unit record, multiple records per case\*
  + Rectangular, fixed or delimited file\*
  + Hierarchical files\*
* Event Records or Spell data (needs InstanceVariable datatype)+
* Aggregate data, mapping to SDMX\*
* CSVW compatibility - reframe not imported into model directly\*
  + Core features, see the CSVW use cases, identify the ones we can do.
  + Not clear if possible for direction of data columns, list in cell
  + See Jira DDI4Data-1
* Codebook (Nesstar, CESSDA list)\*
* Basic descriptives, summary statistics, category statistics\*
* Variable cascade\*
* Weights, (instance variables should have a list of references to possible weight variables)\*
* Streaming data / data sharding / slices+

### Data Collection Information\*

* Mode (Web,PAPI, CAPI etc)\*
* Coverage (topical, temporal, spatial)\*
* Other…

### Data Capture\*

* Basic instruments with response domains and flow\*
* Extend description of <https://ddi-alliance.atlassian.net/wiki/spaces/DDI4/pages/491683/Data+Capture+Team> \*
  + Survey\*
  + Bio-medical assay\*
  + Multi-mode\*
* Response domains\*
* Splits / joins\*

### Discovery\*

* Can we do full DISCO? Produce mapping and consider gaps.\*

### Data Processing\*

* Data transformation - describe derived variable, data cleaning, workflow\*
* Can we support the 3.2 functionality of textual description/computer code, etc.\*

### Content Compatibility with previous DDI

* Cross check with 2.5 and 3.2
  + Archive specific lifecycle information needed? Needs to be done carefully in coordination with creating an execution layer for process+
  + Geography (will be done anyway in the timeframe, relationship to ISO should documented on the item level
* Mapping of DDI4 items to 2.5 and 3.2

## Out of scope for June

* Qualitative
* Complex questionnaires
  + going off somewhere else
  + back end office integration
  + Question Grids
* Execution layer
  + Elaborate a better description of this not running a universal computing engine
* Access restrictions on metadata
* Lifecycle orchestration

## Open questions

* Do we need an example for each use case? Does the lack of an example result in removal of the use case from the prototype?
* Assessment of status of existing content
  + What has not been done?
  + Approval process - clarify criteria, process, effect of a particular status
  + Estimate of time to complete each
  + Prioritization of effort
* Production process and technical outputs (bindings, documentation, etc.)
* Internal review process - content and implementation perspective
* When do we freeze the production process?
* When do we freeze the content? End of March
* When does modeling stop? End of January
* How do test cases fit into all of this - on going validation?

# Timeline

In the past, there has not always been an agreed timeline which has been approved at the management level. What is offered below reflects an attempt to assign realistic time frames to the various production steps we will need to perform in order to produce and package a prototype of reasonable quality within the scope as described in preceding sections.

The “owner” column indicates the team which will exercise control over all changes of the model, and all subsequent artefacts as they are produced. It is the case that upstream changes will be needed, but the management of the artefact set will reside with the team that is indicated here as the owner.

This schedule takes into account the dependencies between deliverables, and the likely availability of people to perform the work over the holiday season. It is recognized that we may need to recruit some additional resources for specific teams to perform their work in a timely manner.

|  |  |  |  |
| --- | --- | --- | --- |
| **Owner** | **Month** | **Work activities** | **Deadlines** |
| Business Modelers | November | Identified parts of the model that are included in the prototype | End of EDDI: End business modeling in Drupal |
| Modeling Team | December/ January | Review of content, revision for bugs, identification of documentation/example requirements | End of January: Freeze structural content in Drupal |
| Documentation / Processing Team | February | Work on documentation and binding content and output |  |
| Documentation / Processing Team | March | Work on documentation and binding content and output | NADDI (bgn or end): Freeze binding and documentation |
| TC owns model, Documentation owns examples | April | TC does model integrity review | Examples done by the end April |
| TC owns | May | TC continues model review and reviews documentation and example | Finalize by end of IASSIST |
| TC owns | June | Final preparation for release for review |  |

# Deliverables

DDI Moving Forward presents a shift in the overall deliverables: while DDI Codebook and DDI Lifecycle are delivered as XML Schema and documentation packages, the Moving Forward production line will also include some other deliverables. In consideration of what the Moving Forward package should contain, the following was identified:

* The Model – although this is at the heart of the documentation, we would also like to have a processable version of the model in XMI (an XML format for describing models). This will be in canonical XMI, a flavor of that format supported across many UML tools. There will only be an XML file for the entire model being published – there is no sub-set for each View, etc. as for some other deliverables.
* XML Schemas – DDI Moving Forward will provide XML descriptions of both the entire breadth of the standard (all objects in the library), and subset XML Schemas for each identified View (groupings of objects to support specific functions). These will be organized such that the XML Schemas are a coordinated set – an element created using a View schema would be valid according to the complete schema, etc.
* RDF-S/OWL Vocabulary for RDF – Our second binding to be delivered is one which would support the expression of DDI metadata as RDF. The delivered vocabulary would be monolithic, describing the entire model, and not have the supporting Views used for the XML. There will be a vocabulary for the entire library, and a set of View-level vocabularies. These would be expressed using OWL. Additionally, we may want to publish SHEX validations for each vocabulary, to allow schema-style validation of graphs created according to any of the published OWL vocabularies.
* Documentation - Documentation of the model is provided in two formats: as a clickable HTML presentation, and as a PDF document. Documentation would be organized such that each separate View would be documented in an abbreviated form of the overall documentation, which would cover the entire library. There will be both high-level documentation, explaining the whole of the specification and its supporting parts, and detail-level documentation covering each class, property, etc.
* Examples (including those specific to each binding) are important. These should be organized functionally (eg, by View) so as to make the model as approachable as possible for implementers. This presumes the existence of good examples which can be reused across the examples for different bindings.
* Mappings: As annexes to the main library specification documentation, we will provide object- and property-level mappings from DDI Moving Forward to DDI Codebook (v. 2.5) and DDI Lifecycle (v. 3.2). Also, we will include mappings against GSIM v. 1.1. While these mappings are currently in the documentation for each object/property (where applicable), this would provide a view of the entire library and how it maps as a whole.

Identified functional views include:

* Data Description (primary)
* Data capture (primary)
* Codebook (primary)
* Agent (secondary)
* Classification (secondary)

Primary views are those which are most important to the prototype release. Secondary view are those views which are useful, but are included because they contain objects also used in primary views.