# DDI RDF-S/OWL Binding: A Survey of Past Work

Arofan Gregory, Deirdre Lungley, Dan Smith

## Introduction

We looked at three documents in light of work during the recent past n RDF, and the discussion being held during the current Dagstuhl workshop (October 16 – 20th). These included *RDF Notes* (24 November 2015, updated January 2016), *RDF Binding* (22 October 2015), and *An RDF Schema/OWL Binding for the DDI Information Model* (Richard Cyganiak’s initial specification of the original transformation made to produce our DDI output in RDF).

We have noted the issues raised where these are either being addressed currently, seem to have been dropped, etc. The goal of this exercise is to perform due diligence. In some cases we have reversed decisions made during earlier work. These cases are noted.

## *RDF Notes* Document

We have chosen to do a very direct representation of the DDI model in RDF, without flattening or simplifying constructs. The original intent seems to have been a more optimized RDF, which had fewer classes, etc., so as to be in line with other RDF vocabularies. Are we OK with this? YES

Abstract classes can be removed from the RDF representation – we have decided to include them. Are we OK with this change? (Also, see below). YES

Is there a way to organize the RDF into a reusable core? DDI already has such a thing – can this be improved, or is this question no longer relevant? NOT RELEVANT

A question was raised regarding the separation of administrative and entity-related metadata. Is this still a concern? The model groups most administrative metadata into Identifiable, DocumentInformation, and AnnotatedIdentifiable now – does this solve the problem?

There was a lot of thinking about persistent real-world entities (such as a person) and metadata about records associated with these entities. Is this topic still meaningful to us? This begs questions about versioning (versioned URIs, etc.) What do we do about this? NOTHING

Patterns: we have implemented useful patterns in the DDI model – what we have not done is found anything actionable around “micro-patterns”. Are we OK with this? YES

Named graphs: How do we advise people concerning the use of named graphs, or do we simply remain mute. Packaging of data and metadata can be an important issue to some users (NSO, for example). Do we do anything here? SEE PACKAGING ISSUE BELOW

Round-tripping (especially of identifiers) was raised here. This is a part of a broader issue which we are actively addressing now. OK - FIX IT!

Indicating equivalence between classes/properties across ontologies/vocabularies – do we do this using OWL, or in some other fashion. This issue is being addressed already in the current workshop. ON TABLE ALREADY

## *RDF Binding* Document

There were several points which seem to have already been addressed here, or are no longer relevant. We start at the “Conclusions” heading.

There is an issue about how to track the evolution of metadata/data throughout the lifecycle. The discussion here seems to have been overtaken by the existence of PROV-O, which could usefully accompany sets of DDI metadata to provide this information. What do we do with this? Document a suggested approach? Remain silent and smirk knowingly in the corner? EXPLORE PROV-O and PAV, SEE WHAT OTHER USE CASES WE HAVE BEST SUPPORTED WITHIN THE DDI MODEL ITSELF (LIKE CLASSIFICATIONS,ETC.)

One suggested use of graphs is to use them to group metadata about a persistent entity, with record-related metadata annotating that graph. This seems not to be a point of focus for us. Is it important? IGNORE

## An RDF Schema/OWL Binding for the DDI Information Model

Much of this document is now a functioning and accepted part of the current binding, and some material is not relevant except as a record of what was done. We do not focus on this type of information. However, some points were made in the document which are not just part of the binding, and deserve mention here.

The document says that triples to be packaged are put into a separate graph. The graph is named with a URI under the control of the publishing organization. We are not currently addressing issues or giving guidance around packaging and publishing. Should we again look at these issues, and craft some advice or rules about how DDI RDF is expected to be handled? HOW DO WE HANDLE PACKAGING? WHAT ABOUT REPUTATIONAL RISK (ESTIMATED VALUES FROM NSOs AS ATTRIBUTES. ETC.) OPEN ISSUE.

Because there is no formalization for abstract classes in RDF-S/OWL, a suggestion here was made to append the string “(abstract)” after the names of such classes, or alternately to use OWL unions. The first approach was seen as simpler, and sufficient. Do we need to look at this issue, given that we are now choosing to include abstract classes in our DDI ontology? BLOW IT OFF? CREATE A CROSS-BINDING SOLUTION? (DO WE ALREADY HAVE ENOUGH INFO IN THE MODEL?) LIST AS RDF BINDING ISSUE. COULD SHEX SOLVE THIS?

Mappings to other vocabularies should be done using RDF-S subClassOf, equivalentClass, subProperty, and equivalentProperty according to this document. Do we agree with this? (Still under discussion in current workshop.) PART OF BIGGER EXISTING ISSUE.

Issue: Is the ordering of values in compositions ever significant? Do we need to capture this? Does our collection pattern as in the current model solve this for us? NOT EVERY COMPOSITION IS MODELLED AS A COLLECTION. UML AGGREGATION AND COMPOSITION HAVE AN “IS ORDERED” PROPERTY. USE THAT? OPEN ISSUE.

WHAT WILL OUR WORK PROCESS FOR NEW VERSIONS OF THE RDF BINDING BE? OPEN ISSUE.