# URL Naming Scheme

Joachim Wackerow, 2015-02-19

## Introduction

Every file of a DDI specification needs an URL. Any published file of any specification should be available “for ever” at a specific location (URL).

The planned publication of first draft parts of DDI 4, of the RDF vocabularies Disco, XKOS, and PHDD, and of a RDF format for the controlled vocabularies raise a couple of questions.

Requirements regarding the URLs for specs:

* The URLs should be persistent
* The URLs of different specs, multiple versions and formats should be consistent across this variety

## Existing System

All locations are below http://www.ddialliance.org/Specification/.

The hierarchy of name, version, and format is represented in a path before the file name with a file type. Additional parts are version of the format and DDI-specific profile (custom use of this format).

### DDI-Codebook

http://www.ddialliance.org/Specification/DDI-Codebook/2.5/

http://www.ddialliance.org/Specification/DDI-Codebook/2.5/XMLSchema/codebook.xsd

### DDI-Lifecycle

http://www.ddialliance.org/Specification/DDI-Lifecycle/3.2/

http://www.ddialliance.org/Specification/DDI-Lifecycle/3.2/XMLSchema/instance.xsd

...

### Controlled Vocabularies

http://www.ddialliance.org/Specification/DDI-CV/

http://www.ddialliance.org/Specification/DDI-CV/AggregationMethod\_1.0.html

http://www.ddialliance.org/Specification/DDI-CV/AggregationMethod\_1.0\_Genericode1.0\_DDI-CVProfile1.0.xml

...

Genericode version 1.0 is used as XML format with a custom DDI-CV profile version 1.0.

### RDF Vocabularies Disco, XKOS, PHDD (only drafts published)

http://rdf-vocabulary.ddialliance.org/discovery.html

http://rdf-vocabulary.ddialliance.org/discovery.ttl

...

This was done along a recommendation of Richard Cyganiak, but we didn’t think on consistency across specs and formats.

## Proposed URLs

### DDI 4

DDI 4 should be better named model-driven DDI-Lifecycle or DDI-L MD because the versioning will take place on the level of the objects, functional views, and packages as well.

http://www.ddialliance.org/Specification/DDI-Lifecycle/MD/

http://www.ddialliance.org/Specification/DDI-Lifecycle/MD/UML/Conceptual\_1.0.0.xmi

http://www.ddialliance.org/Specification/DDI-Lifecycle/MD/XMLSchema/Conceptual\_1.0.xsd

<http://www.ddialliance.org/Specification/DDI-Lifecycle/MD/OWL/Conceptual_1.0.ttl>

or

http://www.ddialliance.org/Specification/DDI-Lifecycle/MD

http://www.ddialliance.org/Specification/DDI-Lifecycle/

http://www.ddialliance.org/Specification/DDI-Lifecycle/UML/FunctionalView/Conceptual\_1.0.xmi

#### Issues

1. **Major issue** (could be discussed and clarified in modelling group with comments from other groups):First, a versioning scheme needs to be developed for DDI 4, the library, the UML packages, and the functional views. The critical point is especially the relationship of parts (packages, functional views) to the whole (object library). This should be done in the light of the usage of the representation formats. There was already a discussion in London on XML namespaces regarding the functional views. Version numbers are part of the namespaces in DDI 3. A separate paper on the agreed versioning scheme seems to be important. It should lay out the reasoning and related examples. Some ideas:
	1. The packages and functional views could have the same major version number as the whole thing, i.e. 4.
	2. An updated package gets a new minor or sub-minor version number.
	3. A functional view gets a new minor number when an underlying package is updated.
		1. This could be the same minor version number as the updated package. This approach could have drawbacks because the version number has basically some meaning. The approach doesn’t work if an updated function view uses an additional package.
	4. A minor version number could indicate a possibly incompatible change.
	5. A sub-minor version number could indicate a compatible change.
2. The file type XMI is often not recognized by applications. An alternative could be Conceptual\_1.0\_xmi.xml.
3. Should OWL or RDF be used as format name for RDF/OWL?

### Controlled Vocabularies

An additional format is developed for CVS, SKOS in RDF/XML and turtle.

http://www.ddialliance.org/Specification/DDI-CV/AggregationMethod\_1.0.ttl

http://www.ddialliance.org/Specification/DDI-CV/AggregationMethod\_1.0.rdf

or

http://www.ddialliance.org/Specification/DDI-CV/AggregationMethod\_1.0\_SKOS2004-02\_DDI-CVProfile1.0.ttl

<http://www.ddialliance.org/Specification/DDI-CV/AggregationMethod_1.0_SKOS2004-02_DDI-CVProfile1.0.rdf>

or

<http://www.ddialliance.org/Specification/DDI-CV/AggregationMethod_1.0_1.0.ttl>

The first version number is for the content, the second for the used structure/format

or

<http://www.ddialliance.org/Specification/DDI-CV/Genericode/AggregationMethod_1.0.ttl>

#### Issues

1. SKOS is just one way in the RDF world for describing controlled vocabularies. SKOS itself has a version (2004-02) and could be used in different ways (here DDI-CV Profile 1.0). Should this be mentioned in the filename?

### Disco, XKOS, PHDD

The issue is that all three vocabularies can’t be categorized in relation to one of both, DDI-Codebook or DDI-Lifecycle. XKOS and PHDD can be seen as independent from them. Disco is based on both, Codebook and Lifecycle.

http://www.ddialliance.org/Specification/Discovery/1.0/OWL/Discovery.html

http://www.ddialliance.org/Specification/Discovery/1.0/OWL/Discovery.ttl

http://www.ddialliance.org/Specification/PHDD/1.0/OWL/PHDD.html

http://www.ddialliance.org/Specification/PHDD/1.0/OWL/PHDD.ttl

#### Issues

1. The URL scheme would open the way for other formats for PHDD like XML Schema: http://www.ddialliance.org/Specification/PHDD/1.0/XMLSchema/PHDD.xsd