# Test Cases

Technical test cases in binding syntax (XML and RDF) for core items of Functional Views (Data Description and Data Capture). The test cases of W3C specifications are the role model for this. A good example is the suite of [test cases for CSV on the Web](https://w3c.github.io/csvw/tests/).

Develop rules for testing interoperability between the two major bindings – XML and OWL/RDF. (See “The ability to roundtrip between bindings”

Data use cases could be reliably done using the CSVW test case model

Data Input: data file

Output: XML instance

(optional: DDI4 Metadata Input? E.g. for metadata driven processing)

For small examples, it could be hand-written and a tool used to utilise the schema definition.

Any tools supporting transformation to the standard could use this to “validate” output.

Does this imply that there is only ONE dialect of DDI4. Another way would be to have a tool which normalizes DDI instances. Both approaches seem to be quite ambitious.

What are we testing?

* Evaluating the model against requirements like features of other specifications (DDI Lifecycle, DDI Codebook, SDMX, CSVW).
	+ Levels
		- Capturing same information content
		- Capturing functionality (interaction of classes)
* Simple examples as description, XML instance (4, 3, 2), RDF instance
	+ Only simple examples can be written by hand
	+ A tool would be required for the generation of more complicated examples
		- Not clear how this tool would work?
	+ SPSS command setup and DDI 4 instance
* Testing the bindings
	+ Conformance of XML instances generated of tools against published XMI instances

Against what can be tested?

Generic formal description in YAML of JSON?

October 16 am

Eric’s presentation of <http://build.fhir.org> as an example of a nice presentation format (user interface) for examples. UML diagrams and hierarchical display of model driven off of same FHIR internal schema model

Demonstration of clickable gsim

Dan Smith demonstrating DDI3 ddimodel-web cogsdata.org

COGS (reference?) <http://ddialliance.github.io/ddimodel-web> <https://github.com/ddialliance/ddimodel-web>

Uses graphvis for diagrams (positioning)

**Minutes from Monday Oct 16 am meeting**

Test suite and two interoperable implementations

Which implementations pass which tests?

Start with simple examples realized in the bindings. These could later be used for tests

This is a little different from use cases which are higher level. Example: a DDI4 codebook of a known study from an archive, or at least a subset of one.

W3C process (Eric) a test suite supplements the documentation . helps to complete documentation (overlooked features)

For things like formats - not a lot of processes - parsing the data in one form into another and then roundtrip to check

Many can contribute, working group does approval, describes issues

Nit picky tests - not interesting to users - used for checking the specification

Intersection between tests and examples more interesting to users

Multiple examples that all interconnect work even better - observation on subject want then click on subject for metadata about subject.

With something like a data format: start from use case, develop test cases (annotation to distinguish nit-picky vs exemplar , a continuum)

Other environments: Obamacare(multiple groups), W3C, FHIR

Obamacare(multiple groups) - oversight group, policy group, developers

Policy to development (provided use cases), performance not testable, but informed changes in the model

Strict naming conventions helped manage coverage.

 Contributors

 Catalogers

Darren Performance issues with large numbers of triples.

Performance as a function of model changes (e.g. direction of a relationship)

Are instances identical - can there be a canonical format

Comparing to W3C on the web

We may not always have a structured output?

XML, OWL/RDF (possibly JSON/LD)

Are there two ways of representation in the XML that are isomorphic?

Test suite may need to deal with both instantiated and referenced content

Example JSON represents a set as a list. Testing may need to sort and eliminate duplicates (canonicalization)

Next steps for this group - make real examples.

Create data with the the tricky properties

FHIR: are the examples used for testing and how are they defined - the canonical form is XML

YAML representation to define - transform to JSON - use that to generate examples.

Alternative start from XML

Would there be sufficient metadata around the YAML?

One directory - multiple documents for the same example

**Questions:**

Can we parse it

Can we find referenced objects

Can we parse, reformat, reformat back, check for equality

What do differences mean? Mistake, binding differences?

W3C manifest

Should we update the ANES example from Dagstuhl 2016 and the IPCSR example from the Lawrence Sprint? How much of this can be expressed in our RDF?

One thing to look for is are there two different ways of implementing entering information - change model or testing suite?

Sept 28 XML schema last one generated.

Next meeting: start from XML of ANES and then convert to RDF

Is the XML schema right? How would this test this against what we want the schema to look like?

Do we want to develop software that traverses the PIM and then has an equality function to compare representations of classes?