**Title of session**: Design Patterns

**Day**: Tuesday

**Participants**: Klas Blomqvist, Flavio Rizzolo, Barrie, Nelson, Gary Cross-Berg, Darren Bell, Guillaume Duffes, Jay Greenfield, Dan Gillman, Arofan Gregory, Wendy Thomas

**Chair**: Arofan Gregory

**Note taker**: Wendy Thomas

**Background information**

Review of the DDI4 patterns and development of the generic principles for developing patterns. This should be a balance between achieving a clean model and allowing understandable use of the model.

Signification Pattern:

* Signifier is the thing you write down. The signified is the thing you are identifying with signifier and the combination of these is the sign.
* How far down and detailed should a pattern be realized? For example, use of realization of signification by class Identifiable.

Underlying patterns:

* Linkages between patterns and between patterns and other parts of the model take place primarily in realizations. There is limited linkage between the patterns themselves.
* The use of Pattern and Realization is a pattern in itself.

PIM and PSM equivalence with logical and physical model

Top down beauty but have to accommodate the bottom up reality

The DDI Pattern document provides a narrative description of patterns and their realizations.

Methodology Pattern

Specification procedure and code = design, process, code

Patterns can touch each other

Complexity of domain vs. complexity of model

Top down beauty but have to accommodate the bottom up reality

DDI4 went with a DDI Lifecycle means of doing patterns because it has been implemented in many places. GSIM had not been proven and so there is some disconnect in places with the GSIM. GSIM is a mix of our Process Pattern and Methodology Pattern. This could be an important and valuable input to the GSIM process as it is not as mature as the DDI work.

Question on the combination of two models. Example:

SamplingMethodology realizes the MethdologyPattern and links into Workflow

Issues for discussion:

Conceptually beautiful wiring? Is it useful? – At what point do you stop using patterns? How do recognize it when you see it? How many realizations do you have? Is this related to scope? When do we stop?

Generalized realization – when are realizations too specific? How do we recognize this sweet spot between simply the pattern and too many or too specific realizations?

What is a “methodology”? Is there a methodological architecture?

You don’t you a design pattern if it is only applicable once (actually or conceivable).

If you have different domains that would use this set of the model but then expand it out within their own domains this is worth capturing it as a pattern.

BPMN mapping to Process Model

Topics for discussion:

Modeling for beauty or function.

* Finding the balance point.
* Simple things should be simple

**Pros/Cons of**

Pro:

* Covers prior versions
* Extensible
* Makes sustainability
* Scalable and handle real world
* Leveraging existing models opens opportunities to work between domains (integrated use)
* Patterns can act as first level of conformity

How to recognize this or evaluate pros and cons:

* Can I express a simple case without all of the extra content?
* Can I express things that are more complex?
* Have you leveraged existing models?
* Guidelines for extension of other ontologies
  + Refinements as opposed to separate classes
  + Providing relation to existing classes in the specified ontology
* The quality of patterns as patterns, how they leverage other patterns
* How well the patterns support user needs
* Efficient patterns – removing or adding complexity
* Documentation and presentation designed to support a simple understandable approach for user
* How do we measure conformance to a model? Schema? RDF Vocabularies?
  + You need to look at it as set of requirements
  + Conformity is at the interface
  + The physical representation of the model is the XMI so use that to check conformance
    - Ridiculously complex
  + Can we use a rule language and schema Tron to test conformity?
* A single way to capture any one piece of metadata within a domain
* A clear description of the metadata captured by each class

Cons:

* Too much wiring
* Do we have classes / properties that are never instantiated?
* Reinventing the wheel, not leveraging existing models
* You might make the model less approachable by too much abstraction or wiring

Barrie (CDISC) We would have done it the way DDI has done it if we had to do it again.

What can be leveraged from schema.org?

* So far don’t see what’s relevant
* Dataset – lots of Dublin Core stuff

*[Links to any relevant documents*]

[Pattern Document](https://ddi-alliance.atlassian.net/wiki/download/attachments/53968978/DDI%20Patterns%20v1.0.docx?api=v2)

Schema.org

**Decisions**

*[Please note any proposals agreed]*

**Issues requiring further discussion**

*[Please note pros and cons for different positions and the reasoning for them]*

Explore FHIR’s use of schema Tron for conformance (talk to Eric)

Revisit idea of formalizing business rules to test conformance

Talk to Eric about SHEX about validation of conformance of the RDF binding using SHEX

Look at how to leverage Schema.org (ask Eric what he meant)

**Is there a need for a continuation of this discussion?**

*[Possible options are further session, plenary, or evening session]*

**Is there a need for a longer document to continue this discussion?**

*[Please note the specific subject and who is going to do this]*