PROV as the underlying data model for brain imaging

1. NIDM Component Layer Cake
   - NIDM Dataset Descriptor
   - NIDM Experiment
     - OpenfMRI, NAT
   - NIDM Workflow
     - FSL, NITRC, SPM
   - NIDM Results
     - Image Processing, Statistical Model

2. PROV-DM → PROV-O → PROV-DM
   - Python Toolbox RDF I/O Support
   - Round-trip invalid in many cases
   - Example:
     wasEndedBy(ex:end1; ex:a1, ex:e1, -,
                  2014-06-23T12:28:54.685000+01:00, [prov:type="a"])  
     wasEndedBy(ex:end1; ex:a1, ex:e1, -,
                  2012-12-03T21:08:16.686000+00:00, [prov:type="b"])  
   - After roundtrip:
     wasEndedBy(ex:end1; ex:a1, ex:e1, -,
     wasEndedBy(ex:end1; ex:a1, ex:e1, -,
                2012-12-03T21:08:16.686Z, [prov:type = "a", prov:type = "b"]  

3. Qualified Patterns are Complex
   "When the qualified form is expressed, including the equivalent unqualified form can facilitate PROV-O consumption, and is thus encouraged."
   - Inconsistent interpretation of above statement
   - Need equivalence between representations
   - PROV is great for Workflows, but
     - Queries are complex without reasoners
     - Example: Using property paths to relate the chain of activities between source and resulting entities

4. Toolchain support
   - Libraries in other languages (JavaScript, R, MATLAB/Octave for domain scientists)
   - Scalable interactive/visualization tools and/or services.
   - Querying PROV documents
     - SPARQL - too complex for domain scientists
     - Lack of query tools
   - More clarity on how to extend PROV to domains - we have been feeling our way:
     - Generate and/or map to vocabularies
     - Extend PROV types and idioms

http://nidm.nidash.org