The Open PHACTS Nanopublication Guidelines
The Open PHACTS RDF/Nanopublication Working Group
V1.81 26-03-2012

INTRODUCTION

Aims & Scope of this Guidelines

This document outlines the relationship between the Open PHACTS System (OPS, see http://www.openphacts.org/) and a semantic data model called nanopublications [1]. The primary aim is to define the format and organisation of nanopublications suitable for inclusion in OPS. This document also clarifies the relationship between nanopublications and "standard RDF" and the relation between nanopublications and large datasets typically found in drug discovery research. In doing so, we also introduce the concept of "prenanopublication".

Intended Audience

This document is intended for data owners who wish to understand the Open PHACTS approach to nanopublication and adopt this within their own projects and wishing to provide data to Open PHACTS in the most citeable form.

Nanopublication schema

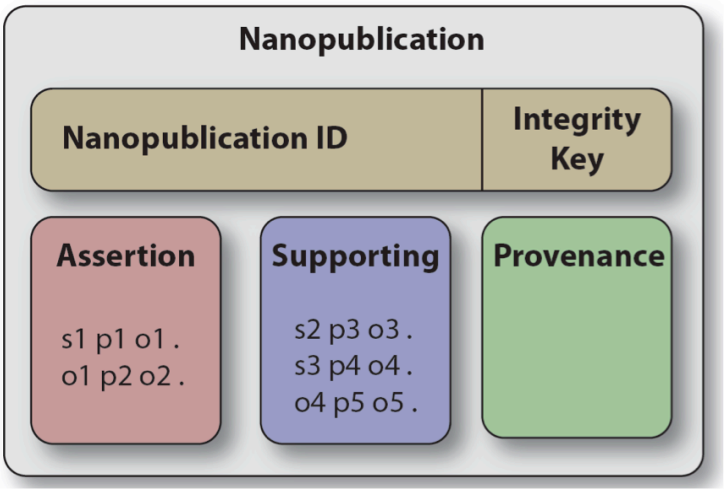


Figure 1: *Anatomy of a second-generation nanopublication implemented using RDF named graphs composed of subject (s), predicate (p) and object (o) combinations, modified from [1].*

A nanopublication is the smallest unit of publishable information: an assertion about anything that can be uniquely identified and attributed to its author. Nanopublications support fine-grained attribution to authors and institutions, with the intention of incentivising the reuse of data [3]. These assertions are organized using (a) the domain semantics drawn from community ontologies and information models, and (b) a nanopublication representation model permitting provenance, annotation, attribution and citation.

RDF and nanopublications were adopted as some of the standards to be used in the Open PHACTS Proposal as a "data model for structuring, linking and organising data…" The OPS nanopublication model is depicted in figure 1. The minimal elements being:

- **The assertion.** This contains statements that compose the scientific assertion being made by the author(s). Statistical p-values and other indicators of validity should be recorded here.
- **The provenance.** This is the authorship or origin of the assertion, how this assertion "came to be". Who made this, when did they make it, who owns the rights? Who gets the credit or the blame for this assertion?
- **The supporting information.** Here, important contextual information regarding the assertion can be added, analogous to "tags" in a Web2.0 context. The primary purpose of this element is to permit high-level filtering of large nanopublication datasets. Crucially, this section contains statements that the author is not claiming are their original invention, but are nonetheless properties that a consumer will need when searching and filtering large nanopublication sets. For example, for an assertion that protein A interacts with protein B, supporting information might include the species (e.g., human, mouse) and method where this was found (e.g., whether the data comes from laboratory experiments or *in silico* predictions).
- **The integrity key.** This ensures authenticity on behalf of the author, i.e. a consumer can be sure that it really is the ascribed author who is making the statement. Note that this is a place holder and the standards here are a work in progress and will not be described in this document.
- **The nanopublication ID.** Each nanopublication is unique and has a URI. Versioning of nanopublication is also considered but is presently under development and we do not provide guidelines/recommendations here.

This nanopublication model is designed to be extensible i.e. as new features are required, one can create new elements in the model (see later section on extensibility). Older nanopublications may not have the newer components, but will retain compatibility. In this guideline we also introduce the concept of "prenanopublication" as a way to publish large experimental datasets in a nanopublication-friendly manner for potential use by the Open PHACTS system.

Principles of Data Provision in Open PHACTS

The Open PHACTS project aims to produce an open semantic framework for pre-competitive pharmacological data. RDF was chosen as the underlying data representation technology given its potential for data integration and interoperation. A full description of the chosen architecture of the system is available at openphacts.org. While it is not nessary to review this to understand this document, there are some key, relevant principles worth mentioning:

- The OPS data store should be thought of as a data cache rather than an as ordinary database. For any dataset, the "authoritative" RDF is produced and hosted by the original provider. The OPS system "harvests" this RDF (with systems to monitor updates etc) and loads it into a local OPS cache for automated reasoning. ***The OPS cache is optimised to support specific queries and web-services required to address drug discovery questions.***
- Consequently, ***the OPS cache is not a "general RDF" repository***. Rather, it holds only those resources required to address specific OPS use-cases.

- For non-pharmacology based use-cases, we believe the same general principles (and much of the OPS software) can be applied to create new cache-based implementations to serve different knowledge domains. Thus, ***OPS will produce not just a working system tailored towards pharmacology, but a toolkit to create similar systems*** for other problems.
- **To be included in OPS, *data needs to be in RDF, but not necessarily as nanopublications***. Although RDF is required the additional information that compose nanopublications is optional and at the discretion of the provider. As the primary purpose of nanopublications is to provide attribution, this decision should be based on the providers' views regarding citeability of the data in question.

Nanopublication Scheme Update

"Anatomy of a Nanopublication" [1] describes the basic principles for constructing individual nanopublications using **RDF Named Graphs**. Importantly, here we update the nanopublication scheme with two additional elements:

1. A nanopublication assertion may consist of more than one subject-predicate-object triple. Specifically, a nanopublication represents a single scientific assertion encoded in RDF, regardless of the number of triples required to represent that assertion.
2. A recommendation that all nanopublications use an ontology to describe the class of each named graph. As the use of named graphs in semantic data is increasing, there is a need to distinguish the use of this approach for nanopublication versus other applications. The current Nanopublication ontology is described below, although one should always refer to the most current version at http://nanopub.org/nschema.

In addition to outlining the use of the nanopublication ontology to type and connect named graphs, the example given below also highlights the use of existing vocabularies to identify predicates and entities wherever possible. It also includes a suggestion as to how the Dublin Core vocabulary can be used to mark a version number for the assertion, again discussed in a later section:

```
# The Nanopublication Schema
# A Nanopublication has an assertion, some provenance
# and some supporting information

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix np: <http://www.nanopub.org/nschema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfg: <http://www.w3.org/2004/03/trix/rdfg-1/>.

np:Nanopublication rdf:type owl:Class.
np:Provenance rdfs:subClassOf rdfg:Graph.
np:Assertion rdfs:subClassOf rdfg:Graph.
np:Supporting rdfs:subClassOf rdfg:Graph.
```

```
np:hasAssertion rdf:type owl:FunctionalProperty.
np:hasAssertion rdfs:domain np:Nanopublication.
np:hasAssertion rdfs:range np:Assertion.

np:hasProvenance rdf:type owl:FunctionalProperty.
np:hasProvenance rdfs:domain np:Nanopublication.
np:hasProvenance rdfs:range np:Provenance.

np:hasSupporting rdf:type owl:FunctionalProperty.
np:hasSupporting rdfs:domain np:Nanopublication.
np:hasSupporting rdfs:range np:Supporting.
```

Using this nanopublication schema, one may then describe a simple gene product nanopublication as follows. In this example, the nanopublication asserts that the human PDE5 gene (UniProt 076074) belongs to a series of Gene Ontology categories (data is taken from bio2rdf):

```
@prefix :    <http://www.example.org/mynanopub/>.
@prefix ex:  <http://www.example.org/>.
@prefix np:  <http://www.nanopub.org/nschema#>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix go:  <http://purl.obolibrary.org/obo/>.
@prefix up:  <http://purl.uniprot.org/core/> .
@prefix pav: <http://swan.mindinformatics.org/ontologies/1.2/pav/>
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

{
  :nanopub1 np:hasAssertion :G1;
            np:hasProvenance :G2;
            np:hasSupporting :G3.
  :G1 a np:Assertion.
  :G2 a np:Provenance.
  :G3 a np:Supporting.
}

:G1 {
 <http://purl.uniprot.org/uniprot/O76074>
       up:classifiedWith go:GO_0000287, go:GO_0005737, go:GO_0007165,
                         go:GO_0008270, go:GO_0009187, go:GO_0030553.
}

:G2 {
  :nanopub1 pav:versionNumber "1.1"
  :nanopub1 pav:previousVersion "1.0".
  :nanopub1 dct:created "2009-09-03"^^xsd:date.
  :nanopub1 dct:creator ex:JohnSmith.
  :nanopub1 dct:rightsHolder ex:SomeOrganization.
  :nanopub1 up:citation <http://bio2rdf.org/medline:99320215>.
}
```

```
:G3 {
  :nanopub1 up:organism <http://bio2rdf.org/taxon:9606>.
}
```

Structuring the data in this way allows us to perform Sparql queries to extract information from nanopublication repositories. For instance, the query below will find all the nanopublications that have some provenance in which the creator is JohnSmith

```
select * where {
    ?nanopub <http://www.nanopub.org/nschema#hasProvenance> ?prov.
    Graph ?prov
     {?s <http://purl.org/dc/terms/creator>
<http://www.example.org/mypubs/JohnSmith>.}
}
```

Large Datasets & PreNanopublications

Singleton Nanopublications versus Big Data Nanopublications

One perspective on nanopublication involves their *de novo* creation by human beings. This can be done using semantic tools while authors are writing articles or reflecting on experimental results. In this case, nanopublications are conceptually similar to traditional publication – a scientist publicly declaring an assertion that should it be re-used in support of other scientific claims. For those wishing to generate one-off or small numbers of *de-novo* nanopublications, ref [1] coupled with the recommendations above should suffice.

However, large numbers of assertions (100s, 1000s or more) may be derived from large online databases and high-throughput experiments. Data providers may use nanopublications as a mechanism to expose individual assertions and enable citation and attribution of these data. Nanopublication consumers, such as Open PHACTS users, may generate results from analysis of these assertions. Those nanopublications that contributed to a new result can be cited via their URI, providing benefit to the authors [3]. Using nanopublications in these two contexts places very different demands on computational infrastructure and so we propose the following recommendations for nanopublications from large-scale data generating systems.

Nanopublications From Large Databases

Nanopublications can be serialised in RDF, which raises the question, how is nanopublication related to the RDF generated for large databases, such as those produced in high-throughput experiments or those in the linked data cloud? Critically, the nanopublication framework was conceived not as a replacement for good, standards-compliant RDF [2], but rather as a way to enable the semantic citation of individual assertions [3]. Nanopublication is designed to *add to* existing RDF, specifically:

Nanopublication is a layer ***on top*** of RDF encoded data to provide a standard for the ***identification of individual scientific assertions*** within a dataset and enables the ***provenance*** to be assigned to each assertion and the entire dataset itself.

An example is shown in **Figure 2a.** Here a dataset in tabular form. Each "row" gives specific relations between entities that can be thought of as specific scientific assertions. When converted to RDF, these assertions can be represented as a collection of semantic triples. Some nodes may be "reused", partaking in multiple assertions (e.g. the URIs that represent the concepts of "human" or "mouse" could be re-used thousands of times to indicate species in each data row). The resulting RDF encodes a graph of nodes and edges, but one where reconstructing specific individual assertions can be difficult for both humans and machines. The named graph approach (**Figure 2b**) augments this RDF, demarcating the scientific assertions and the triples that compose them. Essentially, the named graphs provide a mechanism to "draw a ring around" the set of triples that denote each scientific assertion.

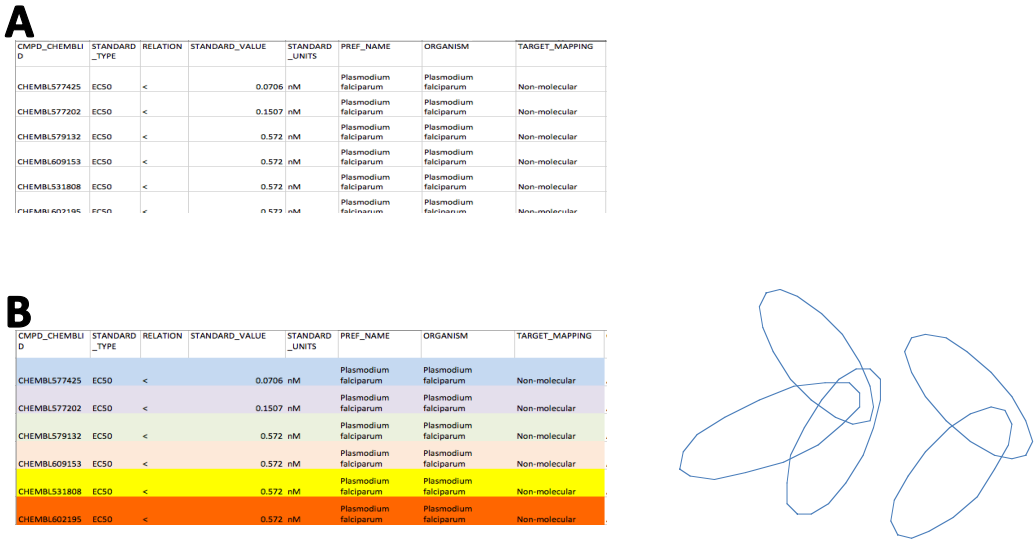


Figure 2: *Named graphs facilitate the identification of individual assertions in RDF. (A) Data in a table (top) can be converted to standard RDF forming individual assertions (bottom, each assertion is coloured separately). This RDF implies a graph structure of nodes and edges. (B) A graphical representation of the data with individual assertions 'circled' and colour coded to match the data in the table.*

Large datasets will often be a mix of assertions, provenance and other supporting data i.e. the components of a nanopublication. Thus, identification of specific assertions is a big step towards nanopublication. For instance, a database of pharmacology results may contain assertions that a drug inhibits a protein with a certain activity value. It may also provide a Medline ID as a reference but also other fields regarding that reference (title, journal, keywords etc). While the

Medline ID is useful information for the nanopublication, the other fields, though perfectly valid inclusions to the overall RDF dataset, they may be seen as ancillary to the assertion or provenance elements of the nanopublication. Therefore:

Assertions and "ancillary data" may co-exist in the same RDF. Not all data in an RDF dataset should be represented as nanopublications.

Prenanopublications

Large, machine produced datasets are now common place. Many bioinformatics and chemoinformatics databases are built by curating data from many different sources including extraction of data from publications, *in silico* models, or other bioinformatics applications. There are also datasets generated from large-scale experiments, such as high-throughput 'omics analyses and pharmacological screening. The principles of linked data encourage authors to release large datasets in RDF as part of the linked data cloud [2], but to increase citeability of these data, dataset producers could expose this data as nanopublication.

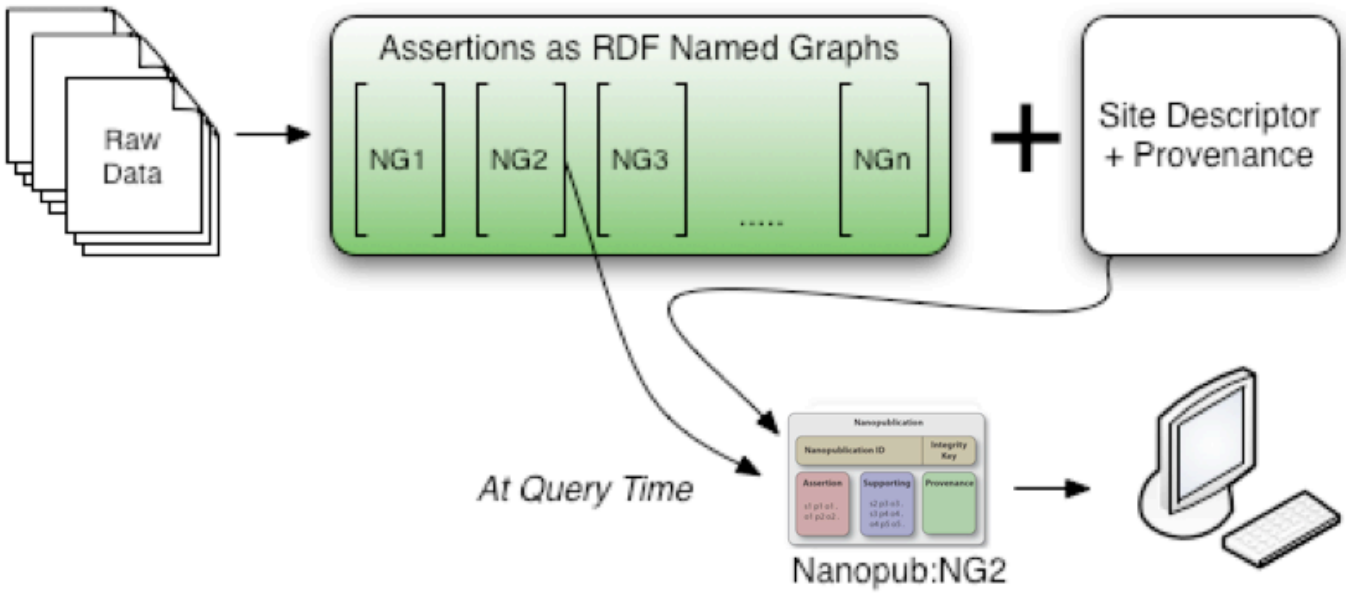
However, before proceeding along this path, producers should consider carefully what sort of information is to be released and how will it benefit data consumers. For nanopublication, some attempt should be made to interpret and summarise the data to produce scientific assertions that can be easily consumed by others. Specifically:

Nanopublications concern scientific assertions. If nanopublications are desired, some level of scientific interpretation should be performed to create actionable information that attaches additional value to specific data points.

We must consider the question: are all nanopublications created "equal"? Nanopublications from scientists declaring a single new scientific fact represent a different type of knowledge compared to 100,000 assertions generated directly from a genomics experiment. For the latter, most of theses assertions may never further scientific progress. We can distinguish this type of assertion from those produced manually via the following features:

- The assertion is simply a data point, its role in any real-world process is as yet, unknown
- There are many (hundreds-millions) of assertions each with the exact same provenance
- All the assertions have been generated in the exact same way (e.g. the same run of a next generation sequencing machine)

The Open PHACTS the system is designed to represent such large datasets in a "nanopublication-compliant" manner without bloating the dataset by recording the exact same provenance for every individual data point. This is called "prenanopublication" and it is described schematically in Figure 3. As with all nanopublications, individual assertions are identified by placing corresponding triples in named graphs. However, provenance and supporting information is only added at the level of the dataset.



Any provenance or supporting information supplied at the dataset level automatically applies to all assertions within that dataset

When queried, the assertions in named graphs can be combined with the dataset meta-data and turned into full nanopublications. Thus, any nanopublication that is generated in this way will have its own URI and provenance, providing all of the citability benefits of the nanopublication approach. Prenanopublications are therefore simply an efficient way to encode large homogeneous datasets while retaining all of the capabilities associated with nanopublications.

Figure 3: *Prenanopublication from large datasets.* Raw data are analysed and individual, scientific assertions are created. Each set of triples corresponding to a single assertion is wrapped in a named graph (NG). Additionally, provenance is assigned at the dataset level using a specific descriptor (technical details on how site descriptors can be used to generate large datasets of prenanopublications are currently under development and will be released at nanopub.org in 2012). When the data is queried and returned, the assertions can be combined with the provenance information to generate 'bona-fide' nanopublications.

How To: Technical Implementation

There are several implementation issues that should be considered when composing nanopublications:

- Make "good" RDF for your data
- Identify triples that correspond to individual assertions and wrap them as named graphs
- Add provenance information at either the dataset or assertion level
- Create a dataset descriptor and publish your RDF to the linked data cloud

To publish your data as nanopublications, follow these steps:

1. Create good quality RDF

Nanopublication is a mechanism to wrap provenance around RDF and therefore has been designed to have as few restrictions as possible regarding how the RDF is generated. Yet, while providers do not need to follow any special rules for producing prenanopublication RDF, it does make sense to consider how RDF encoding might ultimately facilitate the identification and retrieval of assertions in the dataset.

Nanopublication makes no absolute vocabulary/ontology mandates for the RDF generation itself – this is up to the producer. However, the Open PHACTS consortium highly recommends the re-use of existing ontologies, URIs, and data models, which is in line with community-established principles for linked data [2] (see Vocabulary Recommendations below).

Creating a semantically organised model for RDF can be advantageous (as discussed in [4]) and good examples include schemes for text mining results [5] and biological pathways (BioPax, [6]). For more information, also see Semantic Web For the Working Ontologist [7].

Importantly, good RDF requires that the entities within the data be defined unambiguously using URIs. For this tools such as ConceptWiki [8], NCBO BioPortal[9], Identifiers.org [10] and other authorities provide the stable URIs. Finally, if working with experimental data, where possible create MIBBI-compliant [12] data using tools such as ISA [13]. The Appendix provides additional tips on creating good RDF.

2. Do you have Nanopublications?

As emphasised herein, nanopublications offer a mechanism to capture data elements and associate them with provenance. However, not every data point that exists in life science databases requires publishing as a nanopublication. Before considering this approach, the "assertions" in the data need to be defined (such as curated facts, experimental results, etc.) and the question of who would be citing this information, and why, should be carefully considered.

3. Encode the Assertions in the Data

Once the assertions have been defined, the corresponding triples that constitute the assertion should be identified and organised into a named graph. This named graph should, of course, have a “good” URI, meaning the URI is stable, uses a domain owned by the provider, is opaque, is dereferencable and conforms to general URI best practices [2]. The actual URI form is left to the discretion of the individual provider. Following [1] we also recommend the addition of an rdf:type triples to describe each of the named graphs as shown in the nanopublication schema.

4. Vocabulary Recommendations

Open PHACTS recommends the use of established, open, public vocabularies wherever possible to facilitate integration with other resources. Below we present a non-exhaustive list of vocabularies that describe concepts commonly used in life science data. We also advise in using existing terms before minting new ones. To aid with this, the use of the NCBO Bioportal [9] and the ConceptWiki [8] are highly recommended in identifying existing concepts.

Name	**Covers**	**Link**
Dublin Core Terms	Core attribution	http://dublincore.org/
SWAN Ontology	Lightweight Provenance	http://swan.mindinformatics.org/
Open Provenance Model	Provenance	http://openprovenance.org/
Nanopublication Ontology	Nanopublication concepts	http://www.nanopub.org/nschema
Publishing roles ontology	Publication concepts	http://vocab.ox.ac.uk/pro
Data Cube	Statistics	http://publishing-statistical-data.googlecode.com/svn/trunk/specs/src/main/html/cube.html
Experimental Factor Ontology	Common Experimental Concepts	http://www.ebi.ac.uk/efo/
Creative Commons Vocabulary	Licencing	http://wiki.creativecommons.org/CC_REL
Orcid(*)	Author identifiers	http://orcid.org/
Semantic science ontology	Predicates for common scientific relationships	http://semanticscience.org/ontology/sio-core.owl

(*) Note semantic web support of this vocabulary not yet available from the system owner

5. Handling Supporting Information

The supporting information section of a nanopublication provides background and contextual information to support the assertion. Importantly, the aim of supporting information is not to represent the entire experiment or reference within this section but to provide “just enough” to enable first-pass filtering over large nanopublication sets. Clearly, the amount of supporting information included is an empirical and somewhat personal decision. However, we offer the following suggestions:

- The primary application for supporting information is rapid, contextual filtering.
- The supporting information should be simple, ideally single triples to represent the species, cell type, assay method, *in silico* vs. empirical data etc.
- The supporting information may be URIs to further information. In such cases, these URIs should be dereferencable and provide experimental metadata in an easily discoverable

form – i.e. provide a good level of information at the specified URI without the need for extensive graph traversal. Providers should use standard experimental vocabularies (MIBBI, OBI etc) to provide well-established predicates and entities for this data.

- The supporting information for large datasets, such as extensive experimental and other background data, should be included at the dataset level.

6. Versioning Of Nanopublications

A full description of versioning and integrity key supplied with nanopublications with Open PHACTS is currently being prepared. However, in the interim we recommend that all nanopublications contain a version label using Dublin Core dct:hasVersion or a SWAN pav:versionNumber. Additionally, if providers update nanopublications they may wish to use dct:replaces to indicate the update or provide a prior version number with pav:previousVersion.

How providers determine a new versus an updated assertion is their decision; should they decide that the updated content sufficiently changes the science represented by the nanopublication, they may wish to create a new nanopublication with a new URI. The existing nanopublication can then be "deprecated" by including a dct:isReplacedBy <new_nanopub_uri> triple.

7. Extending the Nanopublication Model

The nanopublication schema is designed to be extensible, such that other elements can be created as needed. To do this, one would simply extend the ontology, to describe the named graph that represents the component, and then follow the coding pattern for provenance and supporting information to add additional elements to nanopublications.

8. Publishing Data

Once a provider has created nanopublication data, it should be published in a manner that is both accessible and well described. This means that data should be published according to principles set out in [11], with an RDF file and dataset descriptor available via the providing organisations' web site. In addition to provenance/supporting information, the use of the dataset descriptor allows the Open PHACTS update detector to automatically identify updates and refresh the OPS cache. Where possible the descriptor should provide licensing information using the creative common's vocabulary (see Vocabulary Recommendations, V7). Full technical details of the descriptor are being developed and will be released shortly in an updated guidelines document at nanopub.org.

References

[1] Anatomy Of A Nanopublication, [Link]
[2] Linked Data Book [Link]
[3] The Value of Data [Link]
[4] Interactively Mapping Data Sources into the Semantic Web [Link]
[5] Representing Text Mining Results for Structured Pharmacological Queries [Link]
[6] Biopax [Link]
[7] Semantic Web For The Working Ontologist [Link]

[8] ConceptWiki [Link]
[9] NCBO Bioportal [Link]
[10] identifiers.org [Link]
[11] The Sindice guide to publishing data on the semantic web [Link]
[12] MIBBI: Minimum Information for Biological and Biomedical Investigations [Link]
[13] Investigation-Study-Assay [Link]
[14] Wikipedia page on SI Base Units [Link]
[15] SKOS Simple Knowledge Organization System [Link]

APPENDIX: GENERAL RDF RECOMMENDATIONS

The aim of these guidelines is to describe how existing RDF can be represented as nanopublications. Providers should familiarise themselves with current best practices, such as those outlined in [2],[4],[5] and [7]. In order to assist those new to this area in creating the most compatible form of RDF for their data, OPS provides some general guidance below, but it should be understood that these are not required in order to create nanopublications and are merely suggestions.

Vocabularies, Entities and URIs

- Use as few ontologies as possible (i.e. if you can choose between two ontologies for one concept, try to use same ontology for neighbouring concepts / subtree in schema).
- Use open widely used community standards where possible.
- All entities should be described using identifiers (rather than free text). Further:
    - Provide an rdfs:label with a language tag (e.g rdfs:label "Amsterdam"@en) giving a brief human readable label for the entity.
    - Where appropriate we also recommend using a Dublin Core description (dc:description).
    - Where entities are described using uncommon or private vocabularies, provide mappings based on the SKOS [15] specification to common vocabularies (Orchid, UniProt, UMLS etc) wherever possible.
- URIs should be
    - Opaque, free from semantics
    - Stable (likely to have long term persistence, such as purl's or identifiers.org URIs)
    - Dereferencable i.e. resolve to valid RDF

Expressing Quantitative Data

There are multiple ways to express quantities with associated units in RDF. OPS recommends the following approach:

- Use Custom Datatypes to express units. e.g. "4"^^<http://qudt.org/1.1/vocab/unit#Joule>
- For a unit ontology, we suggest QUDT. To find the units themselves see the following QUDT Unit Vocabulary
- Use SI units. More specifically, convert data to **SI-base units** [14]

Other Technical Guidance

- Turtle and Trig format is preferred over RDF/XML.
- Avoid blank nodes where possible.
- When presenting your model of your data, present the datamodel, i.e. class- and property hierarchy, then provide an example.

Extending the Nanopublication Model

The nanopublication schema is designed to be extensible, such that other “units” can be created as required. To do this, one would simply extend the ontology, to describe the named graph that represents the component, and then follow the coding pattern for provenance and supporting information to add additional “units” to nanopublications.

Authors & Working Group Members

Erik Schultes – LUMC, Christine Chichester – NBIC, Kees Burger – NBIC, Paul Groth – VU, Spyros Kotoulas – VU, Antonis Loizou – VU, Valery Tkachenko – RSC, Andra Waagmeester – Maastricht, Sune Askjaer – Lundbeck, Steve Pettifer – Manchester, Lee Harland - Pfizer/CD, Carina Haupt – UBO, Colin Batchelor – RSC, Miguel Vazquez – CNIO, José María Fernández – CNIO, Jahn Saito – Maastricht, Andrew Gibson (Outside Expert) – Amsterdam, Louis Wich - DTU