

NamedGraphs - Design Issues and Use Cases

This document collects the design issues raised in the discussion about named graphs and describes several use cases for named graphs.

1	<i>Design Issues</i>	2
1.1	de re versus de dicto	2
1.2	bNode Scope	2
1.3	Set of Graphs versus Nested Subgraphs	2
1.4	Relation between Document, Graphset and Graph	3
2	<i>Use Cases and their Requirements</i>	3
2.1	Data Syndication	3
2.2	Signing RDF graphs	4
2.3	Scoping Assertions	5
2.4	Defining Access Rights	7
2.5	Expressing Privacy Preferences	7
2.6	Trust and Believes	7
3	<i>Glossary</i>	8

Syntax used in the examples

I use the **TriG** syntax for the examples in this document. Just another new RDF syntax :-)

Example 1: G1 (S P O . S P' O' . S' P O'')

Meaning, there is an asserted graph G1, containing three statements.

Example 2: G2 [_:x P O]

Meaning, there is an unasserted graph G2, containing a triple with a bnode.

As **query language**, I use something inspired by MacGregor's quad query language.

Example: Select all Persons in Graphs stated by Chris.

```
SELECT  ?x
WHERE
  (?y ?x rdf:type ex:Person)
  (null ?y dc:author ex:Chris)
```

The first element in the patterns is the graph name. "null" is used, if the graph name doesn't matter.

1 Design Issues

This section is keeping track of the different issues raised in the discussion.

1.1 *de re versus de dicto*

Issue: Is de dicto really necessary?

Arguments contra de dicto

1. A point of view is that Semantic Web applications only require de re, because communication presupposes a shared conceptualization, whereas de dicto denies this.
2. The de dicto interpretation of URIs conflicts with TBL's intended usage of URIs, see <http://www.w3.org/DesignIssues/Axioms.html#unique>
3. Graham's argumentation in <http://www.ninebynine.org/RDFNotes/UsingContextsWithRDF.html#xtocid-6303973>
4. There are also practical problems with a de dicto/de re translation which would be required for integration information from different authors. The translation would require complex trust mechanisms, like web-of-trusts, role-based trust mechanisms or probabilistic reasoning, in order to evaluate whom you trust to use the right interpretation of a vocabulary. This gets fussy and might be over-engineered for many applications.

Issue: Do you really have to state explicitly that a graph is unasserted or does the context indicate this? Think of N3 formula or a trust and believe vocabulary.

1.2 *bNode Scope*

The different options are:

1. Graph
Benja favours graph scope in this mail.
2. Graphset / File
The TriX paper has blank nodes with file scope.
Graham: Graph and Graphset scope needed.
3. Global
Meaning bnode semantics combined with a global identifier.
Usefull for merging graphs and for naming graphs with bnodes.

Issue applying to Option 1 and 2: How do you state something about a graph named with a bnode and contained in a different document?

1.3 *Set of Graphs versus Nested Subgraphs*

Issue: Are all graphs in a graphset treated equal or is there a special "top level graph"?

The top level graph idea leads towards nested subgraphs and N3 formula.

Discussion

- Bob MacGregor: Main-graph could contain assertions about the other graphs, like:
G1 (S P O)
top-level (G1 asserted false)
- Jeremy: Nested graphs are too confusing on syntax level.
- Examples in Section 2: Graph sets seem to fit better for most examples.

1.4 Relation between Document, Graphset and Graph

Issue: What is the relation between a document, a graphset and a graph?

One is tempted to use the document URL as graphset URI.

```
Doc1.trix: G1 ( S P O ) G2 ( S' P' O' )  
Doc2.trix: G3 (Doc1.trix is crap)
```

I think there should be a clear difference between content level (graphs) and distribution level (documents) and a clear difference between URIs (naming) and URLs (retrieving). Currently these things are mixed (see test cases) which leads to confusions.

Issue: Has the graphset any semantic meaning?

I would say no.

2 Use Cases and their Requirements

2.1 Data Syndication

Use Case Description

- Different data sources/authors exchange information based on a shared conceptualization.

Requirements

- Statements about graphs
- Provenance tracking (source, author, data)
- Provenance chains: "A said that B said that C."
- Querying specific graphs and groups of graphs.
- Deleting specific graphs from the repository.

Example: Simple Provenance Tracking

```
G1 (Monica ex:hasName "Monica Murphy".  
    Monica rdf:type ex:Person)
```

```
G2 (G1 dc:author Chris.  
    G1 dc:date "2/10/2004")
```

```
G3 (Monica ex:hasSkill ex:Programming)
```

```
G4 (G3 dc:author Peter.  
    G3 dc:date "2/3/2004")
```

Query: Find all information about Monica, which has been stated in 2004.

```
SELECT ?a ?x ?y ?z  
WHERE  
(?a ?x ?y ?z)  
(null ?a dc:date ?b)  
AND ?b > "1/1/2003"
```

Example: Provenance Chains

Peter states, that Chris said that Andy said, that Monica Murphy is a person.

```
G1 (Monica ex:hasName "Monica Murphy".  
    Monica rdf:type ex:Person)
```

```
G2 (G1 ex:saidby Andy.  
    G1 ex:DocumentURL Doc1.trix.  
    G1 dc:date "2/10/2004")
```

```
G3 (G2 ex:saidby Chris.  
    G2 ex:DocumentURL Doc2.trix.  
    G2 dc:date "2/10/2004")
```

```
G4 (G1 dc:author Peter.  
    G2 dc:author Peter.  
    G3 dc:author Peter.)
```

```
G5 (G4 dc:author Peter.  
    G4 dc:date "2/10/2004")
```

2.2 Signing RDF graphs

Requirements

- Statements about graphs
- Or more strictly statements about the equivalence class of a RDF graph
- Grouping mechanism for graphs.
- See Jeremy's paper.

Example: Provenance and Signing

```
G1 (Monica ex:hasStatus Admin.  
    Monica rdf:type ex:Person  
    G1 ex:author Andy.  
    G1 dc:date "2/10/2004")
```

```
G2 (G1 ex:hasSignature "xd2shfl22k4jdsre...".  
    G1 ex:Signer Andy)
```

```
G3 (Andy ex:publicKeyURL http://bla.bla.bla)
```

Query: Get me all authors of the statement “Monica ex:hasStatus Admin” together with their public key and their signature of the graph in which “Monica ex:hasStatus Admin” occurs.

```
SELECT ?a ?b ?c ?d
WHERE
(?d Monica ex:hasStatus Admin)
(null ?d dc:signer ?a)
(null ?a ex:publicKeyURL ?b)
(null ?d ex:hasSignature ?c)
```

Example: Scoped Assertions and Signing

```
_:G1 (Monica rdf:type ex:Employee)
_:G2 (Monica rdf:hasAccessTo ex:RestrictedWebSite)
G3 (_:G1 log:implies _:G2)
```

Problem: You want to sign all graphs together and not the single graphs?

A possible solution:

```
G4 (GraphGroup1 hasMember _:G1.
    GraphGroup1 hasMember _:G2.
    GraphGroup1 hasMember G3.
    G4 ex:saidby Andy.
    G4 dc:date "2/10/2004")

G5 (G4 ex:hasSignature "xd2shf122k4jdsre...".
    G4 ex:publicKeyURL http:bla.bla)
```

Issue: Are RDF Collections suitable for graph groups or are there better solutions?

2.3 Scoping Assertions

- The examples in this section are taken from Graham Klyne: Circumstance, provenance and partial knowledge.
<http://www.ninebynine.org/RDFNotes/UsingContextsWithRDF.html>
- See also R.Guha: Contexts: A Formalization and Some Applications. -
<http://www.guha.com/guha-thesis.ps>

Requirements

- Shared Conceptualization
- No quotation, see
<http://www.ninebynine.org/RDFNotes/UsingContextsWithRDF.html#xtocid-6303973>
- Something like N3 formula
- Logical vocabulary
- Decontextualization and lifting rules

1. Graham’s Metal/Water Example:

```
A consistsOf Metal .
B consistsOf Water .
```

```
{ Metal denserThan Water } log:implies { A sinksIn B } .
```

translates to

```
G1 (A consistsOf Metal .  
    B consistsOf Water .  
    _:G2 log:implies _:G3)  
_:G2 (Metal denserThan Water)  
_:G3 (A sinksIn B)
```

2. Graham's Logical Assertion Example:

```
{ Metal sinksIn Water } a Truth .
```

translates to

```
_:G1 (Metal sinksIn Water.  
      _:G1 a Truth)
```

3. Graham's Combining Different Theories Example:

```
{ Mass a FixedValue } in NewtonianMechanics .  
{ Mass a Variable } in RelativityTheory .  
{ RelativityTheory approximates NewtonianMechanics }  
  when { RelativeVelocities lessThan halfC } .
```

translates to

```
G1 ( _:G2 in NewtonianMechanics .  
      _:G3 in RelativityTheory .  
      _:G4 when _:G5)  
_:G2 (Mass a FixedValue)  
_:G3 (Mass a Variable)  
_:G4 (RelativityTheory approximates NewtonianMechanics)  
_:G5 (RelativeVelocities lessThan halfC)
```

4. Combination of Scoping and Provenance

```
G1 (A consistsOf Metal .  
    B consistsOf Water .  
    Metal denserThan Water)
```

```
G2 (A sinksIn B)  
G3 (A swimsIn B)
```

```
G4 (G1 log:implies G2)  
G5 (G1 log:implies G3)
```

```
G6 (G1 dc:author Chris.  
    G2 dc:author Chris.  
    G4 dc:author Chris)
```

```
G7 (G1 dc:author Peter.  
    G3 dc:author Peter.  
    G5 dc:author Peter)
```

2.4 Defining Access Rights

Requirements

- Graph level: Statements about graphs

```
G1 (Monica ex:hasStatus Admin.  
    Monica rdf:type ex:Person  
    G1 requiresAccessRight Admin)
```

A similar approach is used by Intellidimension: RDF Gateway - Context Based Security.
<http://www.intellidimension.com/default.rsp?topic=/pages/rdfgateway/dev-guide/security/context.rsp>

2.5 Expressing Privacy Preferences

Use Case

- A user wants to restrict the purposes for which published data should be used.

Requirements

- Statements about graphs
- Statements about statements

See

- Plattform for Privacy Preferences: <http://www.w3.org/P3P/> and <http://www.w3.org/TR/p3p-rdftype/>
- Paper: Building Privacy into the Semantic Web. <http://semanticweb2002.aifb.uni-karlsruhe.de/proceedings/Position/kim2.pdf>

```
G1 (Monica ex:hasName "Monica Murphy".  
    Monica rdf:type ex:Person  
    G1 ex:allowedUsage p3p:AllPurposes)
```

```
G2 (Monica ex:eMail mailto:monica@murphy.org  
    G2 ex:disallowedUsage p3p:telemarketing)
```

The examples use a shortened form of P3P.

2.6 Trust and Believes

- Statings are not asserted but uncertain!
- Tracking of provenance information (source, author, data)
- Statings about graphs
- Statings about statements
- Complex queries
- Ranking of query results
- Justification of query results
- See my cRDF stuff.

Example: Believes as Statings about Statements

Peter wants to state that he doesn't believe Chris that Monica has the skill programming.

Possible solution: He states that “Monica has skill programming” is false.

```
G1 (Monica ex:hasName "Monica Murphy".  
    Monica rdf:type ex:Person  
    Monica ex:hasSkill ex:Programming)
```

```
G2 (G1 dc:author Chris.  
    G1 dc:date "2/1/2004")
```

```
G3 (Monica ex:hasSkill ex:Programming)  
G4 (G3 ex:truthValue ex:false)
```

```
G5 (G3 dc:author Peter.  
    G4 dc:author Peter.  
    G3 dc:date "2/3/2004".  
    G4 dc:date "2/3/2004")
```

There is no connection between Chris’ stating and Peter’s stating. Peter says that he doesn’t believe that Monica has the skill, but he doesn’t directly reference Chris’ stating.

Issue: Is this a problem? First I thought “yes” and that you would need statings about statings (not only about statements). Now I don’t see this problem any more. Jeremy?

.... Will be continued ...

3 Glossary

- **'de dicto' (of the speech) vs. 'de re' (of the thing):** Pat : The contrast can be illustrated by the distinction between direct quotation of speech, as in "Louis said, 'Superman is Clark Kent' " vs. "Louis said that Superman is Clark Kent" . The first, de dicto, reports Louis' actual words (and is false, in the story) while the second, de re, reports what she said about someone, using the speaker's words (and if the speaker knows more about Superman than Louis does, might well be true: even though Louis herself wouldn't identify the guy using the term "Superman", she might well have said that Clark Kent was Clark Kent, and of course as we know, Clark Kent *is* Superman.) ... A way to summarize all this is that RDF makes the blanket assumption that all URIs are talking about one single 'reality' and so they always refer in the same way.
See <http://lists.w3.org/Archives/Public/www-rdf-comments/2004JanMar/0050.html>
- **Triple** = just the syntactic form. De dicto, without shared conceptualization.
- **Statement** = Assertion. De Re, with shared conceptualization.
- **Locally asserted Statement:** Assertion within a context, unasserted outside the context. De Re, with shared conceptualization. E.g. Statement within a N3 formula.
- **Stating** = De-Re in social context. A stating is the result of somebody claiming a Statement. The truth value of Statings is uncertain and might even depend on subjective points of view.
- **Context:** There is no general definition of the term context (see 20 years of AI literature). We use the term in the following sense: The graph is the context of a statement.