Semantic Web & BI

Triples (Quads)

Sources of contextualized triple graphs

Analysis & Discovery

Declarative application modelling
Sets API & Functional Model

Triples and their contexts are arranged in a three sets elements dispositions so each set represents one part of the triple and the intersections between two of the sets represents a ‘Kind’ corresponding for the third set. Each Kind aggregates ‘attributes’ and ‘values’ which accounts for the classes and metaclasses of the Resources of the third set themselves. Elements with the same attributes are of the same class and classes sharing the same values are of the same metaclass.

Element, Set, ElementType, Resource, Predicate. Hierarchies. Functional features. Type tree relation with hashed keys. All sets elements are subclasses of Resource and have all four triple/quad resource components (ctx, subject, predicate, object). The kind of Resource each part will holds depends on which Resource subclass is used.

Every set element is an instance of Resource, class which accounts for an individual or reifyied Resource (individual URI or another triple Resource treated as a Resource). Traversal over the sets structure is achieved performing joins between corresponding set resources components and using element types as functors in a functional environment.
SPO Model (Facts)

<table>
<thead>
<tr>
<th>Occurrence</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>Predicate</td>
<td>Object</td>
</tr>
<tr>
<td>Predicate</td>
<td>Subject</td>
<td>Object</td>
</tr>
<tr>
<td>Object</td>
<td>Predicate</td>
<td>Subject</td>
</tr>
</tbody>
</table>

**Triples:**

Occurrences (Subject ex.):
[context] [SubjectURI] [classID] [metaClassID]

Kinds:
[metaClassID] [classID] [attribute] [value]

Contexts:
[context / time] [Subject] [Predicate] [Object]
SPO Model (Facts)

SPO Model is the basic abstraction for source triples/graphs which will be aligned/merged/augmented via the other set models.

Kinds aggregate classes/metaclasses hierarchies encoded in class/metaclass resources. Reifyied Kinds accounts for type hierarchy trees into each SPO set. A class is determined by the intersection of Subjects having the same attributes. A metaclass is the set of same attribute values for a given class.

Sets api manipulation. Sets hold hashed keys relating an element’s position into set types tree to an elements map. Functional algorithms.

Abstract Predicates, possible kind individuals: query.
Semiotic Model (SCO, Contexts)

Signs (SPOs)
Roles
Concepts (SPO Kinds)
Topics
Context
Individuals

Objects (SPO Triples)

<table>
<thead>
<tr>
<th>Occurrence</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign</td>
<td>Concept</td>
<td>Object</td>
</tr>
<tr>
<td>Concept</td>
<td>Object</td>
<td>Sign</td>
</tr>
<tr>
<td>Object</td>
<td>Concept</td>
<td>Sign</td>
</tr>
</tbody>
</table>

**Triples:**

- **Occurrences (Object ex.):**
  - [context] [ObjectURI] [classID] [metaClassID]

- **Kinds:**
  - [metaClassID] [classID] [attribute] [value]

- **Contexts:**
  - [Topic] [Object] [Concept] [Sign]
Semiotic Model (SCO, Contexts)

Identify equivalent Resources and merge. Similar object graphs structures.

Signs / Individual

Concepts / Context

Objects / Topic
Behavior Model (TSP)

### Triples:

**Occurrences (Topic ex.)**

[context] [TopicURI] [classID] [metaClassID]

**Kinds:**

[metaClassID] [classID] [attribute] [value]

**Contexts:**

[Purpose] [Topic] [Scenario] [Player]
Behavior Model (TSP)

Topics / Role

Scenarios / Definition

Players / Performance

Purposes (Patterns)

Topics as a pattern to “reacts” to input triples. Merge topics. Update sets tree hash map keys. Identify similarity, later in Semiotic Model identify equivalence.
Applications


Linked Topics Purpose driven apps. Declarative, domain driven development.

Backend: p2p nodes running services/clients (apps) declarative clients / agents.

Triple encoding and peers resolution. Registry, dataflow.

Applications and client agents evolve from domain data (and schema) specifications.