MARKLOGIC SEMANTICS OVERVIEW

Stephen Buxton, Senior Director, Product Management, MarkLogic
It's not magic …
SEMANTICS IN A NUTSHELL
Triples – A Data Model

Data is stored in RDF triples, expressed as:

Subject: Jamie Vardy
Predicate: team
Object: Leicester

Subject: Leicester
Predicate: league
Object: Premier League

RDF triples
Triples – A Data Model

Data is stored in RDF triples, expressed as:

- Subject: User1
- Predicate: runs
- Object: App1

- Subject: App1
- Predicate: runsOn
- Object: Cluster1

See also: http://ontology.it/itsmo/v1/itsmo.html
Triples – A Data Model

Data is stored in triples, expressed as:

:Subject :Predicate :Object
:User1 :runs :App1
:App1 :runsOn :cluster1
URIs – Connecting the Graph

- **URI (Universal Resource Identifier)** – Like a URL, but it *Identifies* a thing, it does not *Locate* a Thing
- URIs make graphs from Triples and combine graphs from different sources
URIs – Connecting the Graph

- **URI (Universal Resource Identifier)** – Like a URL, but it *Identifies* a thing, it does not *Locate* a Thing
- URIs make graphs from Triples and combine graphs from different sources
URIs – Connecting the Graph

- **URI (Universal Resource Identifier)** – Like a URL, but it *Identifies* a thing, it does not *Locate* a Thing
- URIs make graphs from Triples and combine graphs from different sources
SPARQL – A Query and Update Language

Query with SPARQL, gives us:

- **Simple lookup** – Find people who run App1
- **Graph traversal** – Find people who run an application that depends on Cluster1
- **Inference** – Find people who depend on Cluster1
- **Update** – App1 now runs on Cluster2
SELECT ?user
FROM <myGraph>
WHERE {
  ?user :runs :App1
}
SPARQL

Graph Traversal

“Find people who run an application that depends on Cluster1”

SELECT ?user
FROM <myGraph>
WHERE {
}

SELECT ?user
FROM <myGraph>
WHERE {
}
SELECT ?user
FROM <myGraph>
WHERE {
}
SPARQL Update

“App1 now runs on Cluster2”

```
DELETE { ?app :runsOn :Cluster1 }
INSERT { ?app :runsOn :Cluster2 }
WHERE

{ ?app :name 'my application'
}
```
Visualization – Your Data At A Glance
Visualization – Your Data At A Glance
Ontologies and Inference
Ontology – A Vocabulary With A Shared Meaning

1. I want to assert some facts (data)
   
   `<Dept1> owns <App1>`

2. I want to say something about the data (ontology)
   
   `<App1> is licensed software`

3. I want to say something more general about the data model (ontology)
   
   Every piece of licensed software is an asset

4. For #2 and #3, I need some language (ontology language)
Ontology – A Vocabulary With A Shared Meaning

My **Ontology** says:

```xml
<App1>   <isA>  <licensedSoftware>
<licensedSoftware> <subClassOf> <asset>
<hardware>  <subClassOf> <asset>
```

My **Ontology Language** says:

Formal definition of `<isA>`, `<subClassOf>`

**If**

A is a B **and**

B is a subclass of C

**then** A is a C

"Show me all the assets that Dept1 owns"

Popular Ontology Languages:
- RDFS
- OWL
- RDFS-Plus
- OWL-Horst
"But I know what owns/is a /subclass means …"

"show me all the assets that <Dept1> owns"

The data says:

<Dept1> owns <App1>

The ontology says:

<App1> <isA> <licensedSoftware>
<licensedSoftware> <subClassOf> <asset>

The ontology language says:

If A is a B and B is a subclass of C then A is a C

Therefore <App1> is an asset that <Dept1> owns.

"show me all the zigs that <bis> aks"

The data says:

<bis> aks <bab>

The ontology says:

<bab> cuf <deb>
<deb> bon <zig>

The ontology language says:

If A cuf B and B bon C then A cuf C

Therefore bab is a zig that bis aks
Inference – Automatically Infer "New" Data

Inference = Data + ontology + ontology language

- Data is triples in the database
- Ontology is triples in the database
- Ontology language is a vocabulary with a shared meaning – rules
  - Apply any set of rules to any query!
  - Make up your own rules!
- Simpler, more robust data modeling
- New insights
SEMANTICS IN A NUTSHELL
MarkLogic Architecture

**INTERFACE LAYER**
- JSON, XML, RDF, Geo, Binaries
- mlcp
- REST API
- Graph / SPARQL

**QUERY LAYER**
- JavaScript
- XQuery
- SQL
- SPARQL

**INDEXES / CACHE**
- Universal Index
- Geospatial Index
- Reverse Index
- Triple Index
- Triple Cache

**STORAGE LAYER**
- Scalability and Elasticity
- ACID Transactions
- Automated Failover
- Triple Store
Oh the things you can do …
Use Triples when you want to ...

- Store and query hundreds of billions of facts and relationships
- Explore a graph
- Visualize a graph
- Leverage standards: data + query
- Infer new information
  - better insights
  - simpler data modeling
- Semantics of data
  - integration
Use Documents when you want to …

- Easily store heterogeneous data (transactional data, records, free-text)
- Schema-agnostic
  - modeling freedom
  - integrate without ETL*
- Search flexibility and specificity
- Fast app development
Sidettrack – Documents and Data

Article
- Title
- Date
- Abstract

Body
- Section
  - Section
    - Paragraph
    - Paragraph

Trade
- Type
- Date
- Amount

Parties
- Seller
- Channel

Buyer
- Name
- Affiliation
- PaidBy
Benefits of a Document Store and Triple Store Combined

All the benefits of each, plus:

- Docs can contain triples, Triples can annotate docs, Graphs can contain docs
  - Faster data integration using semantics as the glue
  - Ideal model for reference data, metadata, provenance
  - Ability to run really powerful queries
- Massive speed and scale
- Simplicity of a single unified platform
- Enterprise features (security, HA/DR, ACID transactions,...)
TRIPLES AND DOCUMENTS
Triples Alongside Documents

- Senior Manager: rank
- Compliance Officer: role

User1:
- basedIn: Geneva
- runs

App1:
- runsOn: Cluster1
- requires: TopSecret
- accesses: Database1

App1 runsOn Cluster1 and requires TopSecret which accesses Database1. User1 is based in Geneva and runs on Cluster1.
Show me documents that mention App1 (or its dependencies)
- … and "trades" or "markets"
- … that were valid yesterday afternoon
- … that were produced near HQ
- see Intelligent Search, Infobox
- Show me instructions to access App1
  - App1 user guide
  - How to get TopSecret access
  - Scope of Database1
  - see Dynamic Semantic Publishing
Documents as Part of the Graph

- Senior Manager
- rank
- basedIn
- Geneva
- Compliance Officer
- role
- runs
- order
- user guide
- tutorialMovie
- deep dive
- requires
- license
- accesses
- runsOn
- Cluster1
- TopSecret
- Database1
- runs
- requires
- Cluster1
- TopSecret
- Database1
- requires
- Cluster1
- TopSecret
- Database1
Documents as Part of the Graph

- Document as opaque object
  - Show me all the instructional documents related to App1
- Search inside the document
  - Show me all the applications that managers use that expire in the next 6 months
Triples About Documents – Extended Metadata

- Senior Manager
- Compliance Officer
- User1
- Cluster1
- App1
- TopSecret
- Database1
- Geneva
- Delaware
- 2016-12-31
- High risk person
- English
- JSON
- runsOn
- requires
- runs
- basedIn
- role
- expires
- order
- jurisdiction
- Ts and Cs
- language
- format
- accesses
- runsOn
- requires
- basedIn
- role
Triples are a natural way to represent metadata about documents

*Extended* because that metadata is part of the graph

Example: show me all orders for a TopSecret app that will expire soon
Triples About Documents - Integration

The Semantics of Data

Vendor equivalent Seller equivalent Provider

App1 order format XML

License Computer Asset

Semantic Data Model

The Semantics of Data

Vendor equivalent Seller equivalent Provider

App1 order format XML

License Computer Asset

Semantic Data Model
Triples About Documents

- Data Integration: Dirty data
  - Show me license documents from vendor Acme
- Data Integration: Overlapping data
  - Show me all assets from vendor Acme
Triples as part of a document
<order id="12345">
    <VENDOR>Acme Corp</VENDOR>
    <payment>
        <amount>3427</amount>
        <unit>USD</unit>
        <period>annual</period>
    </payment>
    <sem:triple>
        <sem:object>2016-12-31</sem:object>
    </sem:triple>
    <sem:triple>
        <sem:predicate>http://youruri.com/predicates/TsAndCs</sem:predicate>
        <sem:object>http://youruri.com/terms/34567</sem:object>
    </sem:triple>
    <description> .... </description>
</order>
Set of Triples with XML [JSON] annotation

```
<userInfo>
  <source>myApp44</source>
  <confidence>100</confidence>
  <location>37.52 -122.25</location>
  <icd9-proc-code>1111</icd9-proc-code>
  <temporal>
    <systemStart/><systemEnd/>
    <validStart>2014-04-03T11:00:00</validStart>
    <validEnd>2014-04-03T16:00:00</validEnd>
  </temporal>
  ...
  <sem:triple>
    <sem:subject>http://youruri.com/users/11111</sem:subject>
    <sem:object>http://youruri.com/applications/1111</sem:object>
  </sem:triple>
  <sem:triple>
    <sem:subject>http://youruri.com/users/11111</sem:subject>
    <sem:object>http://youruri.com/applications/3333</sem:object>
  </sem:triple>
</userInfo>
```
Triples as part of a document

- Embed triples in a document
  - Triples and document have the same security, transactions, backup, temporality, …

- Annotate triples in an entirely generic way (XML or JSON)
  - Provenance
  - Confidence
  - Bitemporal

- Query across triples and documents in the same query
  - SPARQL, restrict result to some source, confidence range, bitemporal range
  - Search, restrict result to documents that contain some facts or metadata
Use Triples when you want to …

- Store and query hundreds of billions of facts and relationships
- Explore a graph
- Visualize a graph
- Leverage standards: data + query
- Infer new information
  - better insights
  - simpler data modeling
- Semantics of data
  - integration
Use Documents when you want to ...

- Easily store heterogeneous data (transactional data, records, free-text)
- Schema-agnostic
  - modeling freedom
  - integrate without ETL*
- Search flexibility and specificity
- Fast app development
Document Store and Triple Store Combined

All the benefits of each, plus:

- Docs can contain triples, Triples can annotate docs, Graphs can contain docs
  - Faster data integration using semantics as the glue
  - Ideal model for reference data, metadata, provenance
  - Ability to run really powerful queries
- Massive speed and scale
- Simplicity of a single unified platform
- Enterprise features (security, HA/DR, ACID transactions,...)
Summary

- Semantics is not magic …
- MarkLogic is an Enterprise Triple Store
- MarkLogic is a Document Database
- Documents + Triples + Combinations means:
  - model your data in the right way
  - integrate data from many sources in many shapes
Semantics Use Cases
Leading Organizations Using MarkLogic Semantics

- Intelligent Search
- Semantic Metadata Hub
- Dynamic Semantic Publishing
- Recommendation Engines
- Compliance
Before MarkLogic

**Slow, expensive delivery**

Repair manuals
Technical bulletins
Recall notices

Expert advice

Shredded content
Expert content

Manual data entry

After MarkLogic

Comprehensive digital content delivery

- Repair manuals
- Technical bulletins
- Recall notices
- Expert advice
- Historical Repair Orders

Sophisticated search and predictive repair analytics

- 100 Million Docs
- 28 Manufacturers
- +10GB/month
Getting smarter with semantics

1. Link different terms that mean the same or similar things
   - gasket ↔ oil pan gasket

2. Compositional hierarchy to relate each part to the whole (“partonomy”)
   - Engine
     - Engine cooling
     - Conditioner
     - compressor

196,000+ Unique Vehicles

Vocabulary 1
Vocabulary 2
Vocabulary 3
Vocabulary 4
Leading Organizations Using MarkLogic Semantics

- Intelligent Search
- Semantic Metadata Hub
- Dynamic Semantic Publishing
- Recommendation Engines
- Compliance
Entertainment Company – Semantic metadata hub

Inception | Pre-Production | Production | Post-Production | Distribution | Archive

- Script
- Script Supervision
- Budget Scheduling
- SyncOnSet
- Prop Inventory
- TOPS
- Editorial
- Avid
- DETE
- Taxonomies
- Semantics
- Technical Administrative Metadata
- Descriptive Metadata
- Content
- Customers!

Complex Data Integration
Entertainment Company – Semantic metadata hub

Data Model Using Documents + Data + Triples

Dynamic, semantic metadata hub
Leading Organizations Using MarkLogic Semantics

- Intelligent Search
- Semantic Metadata Hub
- Dynamic Semantic Publishing
- Recommendation Engines
- Compliance
For 2012 Olympics, semantics helped BBC manage content for web pages with real-time updates—without additional support.

Semantic Inference

1. Diego Costa plays for Chelsea
2. Chelsea is in the Premier league
3. Diego plays in the Premier league

X 10,000
Leading Organizations Using MarkLogic Semantics

- Intelligent Search
- Semantic Metadata Hub
- Dynamic Semantic Publishing
- Recommendation Engines
- Compliance
Recommendation Engines
Recommendation Engines
Leading Organizations Using MarkLogic Semantics

- Intelligent Search
- Semantic Metadata Hub
- Dynamic Semantic Publishing
- Recommendation Engines
- Compliance
Amgen – Integrated Drug Terminology Database
Connecting internal and external knowledge via the power of open graph data

MarkLogic advantage:
- **Semantics / graph** – the power of native RDF support to ingest standardized datasets, combined with enterprise proof points
- **Flexible model** – potential to further expand from primarily RDF to documents and document-like metadata
Leading Organizations Using MarkLogic Semantics

- Intelligent Search
- Semantic Metadata Hub
- Dynamic Semantic Publishing
- Recommendation Engines
- Compliance
It's not magic ...