TELEIOS 2nd User Community Workshop
10 - 11 May 2012

Session 3 – Query Builder User Interface

Robert Gregor

Spatial Information Management

Fraunhofer Institute for Computer Graphics Research IGD
Session Outline

- Introduction
- Basic Concept
- System Overview
- Integration with other HMI components
- Roadmap
- Questions & Discussion
Introduction – What we had
DLR's EOWEB

Queries available EO products by selecting...

...bounding rectangle

...acquisition time range

...classes / types from EO product type taxonomy

Technically speaking:

- Requires matching Java plugin version available and installed

- Certain browsers / platforms have issues even if correct plugin version is installed and enabled

- What about organizational security policies: disabled plugins, firewalls, (filtering-)proxies, ...?
Among the User Stories: “I want to...”

- “...query the products based on their semantic content...”
- “...save the parameters of a query and share them with my colleagues...”
- “...retrieve information about fires using semantic, geographical and temporal constraints such that I can mash-up the retrieved information with information from other systems...”

Among derived requirements:

Develop various new user interfaces while...

“These user interfaces shall be adaptable to the particular needs of the individual users and their conjectures and shall provide extended EOWEB portal capabilities.”
Introduction - What we're also working on: “Semantic Technology”

For NOA and DLR Use Case:

- Ontologies that “capture” semantic structure of the available meta-data have been developed
- Existing meta-data can be converted into stRDF
- Meta-data is stored in Strabon and can be queried via stSPARQL

Well, couldn't we exploit all that for a HMI approach which allows end users to query EO product meta-data in a more flexible way?
Basic Concept – What we came up with: ...connecting some dots...

Let's design a new HMI component...

...the Query Builder

- Impose as few restrictions as possible on query formulation against semantically structured, spatial data sets
- Maintain usability for end users
- Let meta-data ontologies “drive” large parts of the UI facilities
- Stick to HTML5, avoid plugins, be cross-browser compliant
Allow users to...

- reuse and adapt predefined queries
- save and share queries
- express complex spatial / topological constraints

Provide different interfaces for different users:

- “Simple editor” based on sequence of Filter Facets
- “Graph-based editor”
- text based stSPARQL editor
Basic Concept: Mockup Overview

BASE <http://www.earthobservatory.eu/ontologies/dlrOntology.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
SELECT DISTINCT ?patch ?xmlFile
WHERE {
  ?prod rdf:type Product .
  ?patch rdf:type Patch .
  ?label rdf:type Label .
  ?prod hasXMLFilename ?xmlFile.
  ?prod hasImage ?img .
  ?img consistsOf ?patch .
  ?patch hasGeometry ?geo .
  ?patch hasLabel ?label .
  ?label hasPercentage ?coverPercentage .
  { ?label correspondsTo LowDensityResidual } 
    UNION 
    { ?label correspondsTo Parking } 
    FILTER(?coverPercentage >= 1 && ?coverPercentage <= 100)
  } 
FILTER NOT EXISTS (?geo strdf:disjoint stRDF^^POLYGON((lat0 lon0, lat1 lon1, lat2 lon2, lat3 lon3))
}
Basic Concept: Filter Facet Boxes

alignment

provide Facet Box Types for:

- taxonomies
- spatial location / extent
- temporal Properties
- “kinds of topologies”
- generic box type for primitive properties
Basic Concept:
Simple Query Editor
Basic Concept:
Graph-Based Query Editor (II)

- add / remove facets
- toggle predicate negation
- toggle between conjunctive and disjunctive selection mode
System Overview
System Overview – other involved technologies / frameworks

Client:

- JavaScript Application
- Exposes JavaScript API
- XS-support
- Developed mostly in Java

Server:

- Main application is OSGi RFC 66 compliant Web Application Bundle (WAB)
- Entirely structured as set of OSGi bundles
- Pure Java
System Overview – other involved technologies / frameworks

Client:

- Vaadin (+ addons)
- GWT
- OpenLayers
- Web-Browser

- JavaScript Application
- Exposes JavaScript API
- XS-support
- Developed mostly in Java

Server:

- Vaadin (+ addons)
- Eclipse Gemini Web
- TinkerPop
- Apache Tomcat
- Eclipse Blueprint
- Neo4J
- Eclipse Equinox
- Java Runtime Environment

- Main application is OSGi RFC 66 compliant Web Application Bundle (WAB)
- Entirely structured as set of OSGi bundles
- Pure Java
Roadmap

- **Focus on:** Implementation of Graph-Based Query Editor

- **Next Milestone:** 01.09.: working component for integration with other TELEIOS HMI components
  - Strabon Integration
  - Persistence for Queries, Presets
  - Query Sharing
  - LDAP Integration

- **TBD:**
  - Implementation of other Query Editors (hierarchical filter facets, text based)
  - Linked Data Integration

Additional development in **FP7: plan4business?**
Open Questions & Discussion

Questions?

What about your Use Case / Scenario?

What data? How to query?

Suggestions?

Linked Data?