GeoVoID: Geospatial Dataset Discovery in the Semantic Web

GeoVoCamp
Madison, WI
June 2, 2014

Todd Pehle

orbis
TECHNOLOGIES, INC.

Are you ready for the answers?

Agenda

- Motivation for GeoVoID
- Introduction to VoID Vocabulary
- GeoVoID Vocabulary
- Application of GeoVoID
- Next Steps

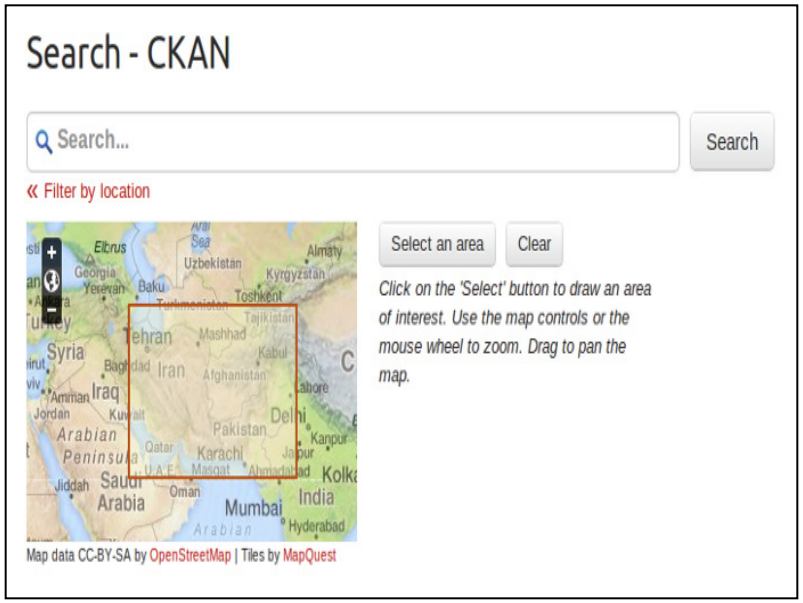
Motivation for GeoVoID

Data Discovery in Geo Web

1. Discover OGC Catalog
2. Search Catalog by Feature Type/BBOX
3. Discover OGC WFS Service
4. GetCapabilities
5. GetFeature
6. DescribeFeatureType
7. Add WFS Layer(s) to Map
8. Get Feature By ID

Data Discovery in Semantic Web

- VoID Capabilities:
  - General metadata
  - Structural
  - Class/property partitions
  - Linksets
- DCAT Capabilities:
  - Interoperability of Catalogs
  - Non-RDF Catalogs



Source: http://docs.ckan.org/en/latest/geospatial.html

- Data portals like CKAN (now called Datahub)
  - Offers BBOX dataset queries
  - Has extension support for CSW
- Offers more flexible discovery vs. centralized catalogs AND socialized links (VoID Repos, URI backlinks, etc.)

Geospatial Data Discovery with GeoVoID

Goals:

- Enable discovery of geographic feature data and services in LOD via:
  - Feature Type Discovery
  - Feature Type Spatial Extents
  - Dataset Spatial Extents
  - Thematic Attribution Schema Discovery (maybe)
  - GeoSPARQL Endpoint Discovery
- Reuse and extend existing LOD vocabs vs. reinvention adding additional heterogeneity
- GeoVoID serves *partially* as a WFS GetCapabilities and DescribeFeatureType for LOD

Introduction to VoID Vocabulary

VoID

- Small but useful vocabulary to describe published Linked Data datasets:
  - Dataset metadata: authors, service access, download URLs, etc.
  - Content metadata: Dataset, subsets, linksets, class and property partitioning, vocabulary usage, etc.
- Publishers create a void.ttl file and host in root directory or embed as RDFa in home page
- Non-authoritative publishers and aggregators can also create ‘void-stores’ hosting multiple VoID descriptions

VoID General Dataset Metadata

Term	Purpose
dcterms:title	The name of the dataset.
dcterms:description	A textual description of the dataset.
dcterms:creator	An entity, such as a person, organisation, or service, that is primarily responsible for creating the dataset. The creator should be described as an RDF resource, rather than just providing the name as a literal.
dcterms:publisher	An entity, such as a person, organisation, or service, that is responsible for making the dataset available. The publisher should be described as an RDF resource, rather than just providing the name as a literal.
dcterms:contributor	An entity, such as a person, organisation, or service, that is responsible for making contributions to the dataset. The contributor should be described as an RDF resource, rather than just providing the name as a literal.
dcterms:source	A related resource from which the dataset is derived. The source should be described as an RDF resource, rather than as a literal.
dcterms:date	A point or period of time associated with an event in the life-cycle of the resource. The value should be formatted and data-typed as an xsd:date.
dcterms:created	Date of creation of the dataset. The value should be formatted and data-typed as an xsd:date.
dcterms:issued	Date of formal issuance (e.g., publication) of the dataset. The value should be formatted and datatyped as an xsd:date.
dcterms:modified	Date on which the dataset was changed. The value should be formatted and datatyped as an xsd:date.

VoID Access Metadata

SPARQL Endpoint URL:

```
:DBpedia a void:Dataset;
        void:sparqlEndpoint <http://dbpedia.org/sparql> .
```

RDF Data Dumps:

```
:NYTimes a void:Dataset;
        void:dataDump <http://data.nytimes.com/people.rdf>;
        void:dataDump <http://data.nytimes.com/organizations.rdf>;
            void:dataDump <http://data.nytimes.com/locations.rdf>;
        void:dataDump <http://data.nytimes.com/descriptors.rdf>; .
```

URI Lookup Endpoints:

```
:Sindice a void:Dataset ;
        void:uriLookupEndpoint <http://api.sindice.com/v2/search?qt=term&q=> .
```

VoID Structural Metadata

Datasets and Subsets:

```
:DBpedia a void:Dataset;
    void:subset :DBpedia_shortabstracts;
    void:subset :DBpedia_infoboxes; .
:DBpedia_shortabstracts a void:Dataset;
    dcterms:title "DBpedia Short Abstracts";
    dcterms:description "Short Abstracts of Wikipedia Articles";
    void:dataDump
    <http://downloads.dbpedia.org/3.3/en/shortabstract_en.nt.bz2>;
:DBpedia_infoboxes a void:Dataset;
    dcterms:title "DBpedia Infoboxes";
    dcterms:description "Information that has been extracted from Wikipedia
    infoboxes.";
    void:dataDump <http://downloads.dbpedia.org/3.3/en/infobox_en.nt.bz2>; .
```

VoID Structural Metadata – Continued

Linksets:

```
:DBpedia2DBLP a void:Linkset;
        void:target :DBpedia;
        void:target :DBLP;void:linkPredicate owl:sameAs;
```

Class and Property Partitions:

```
:MyDataset a void:Dataset;
    void:classPartition [ void:class foaf:Person; ];
    void:classPartition [ void:class foaf:Organization; ];
    void:propertyPartition [ void:property foaf:name; ];
    void:propertyPartition [ void:property foaf:member; ];
```

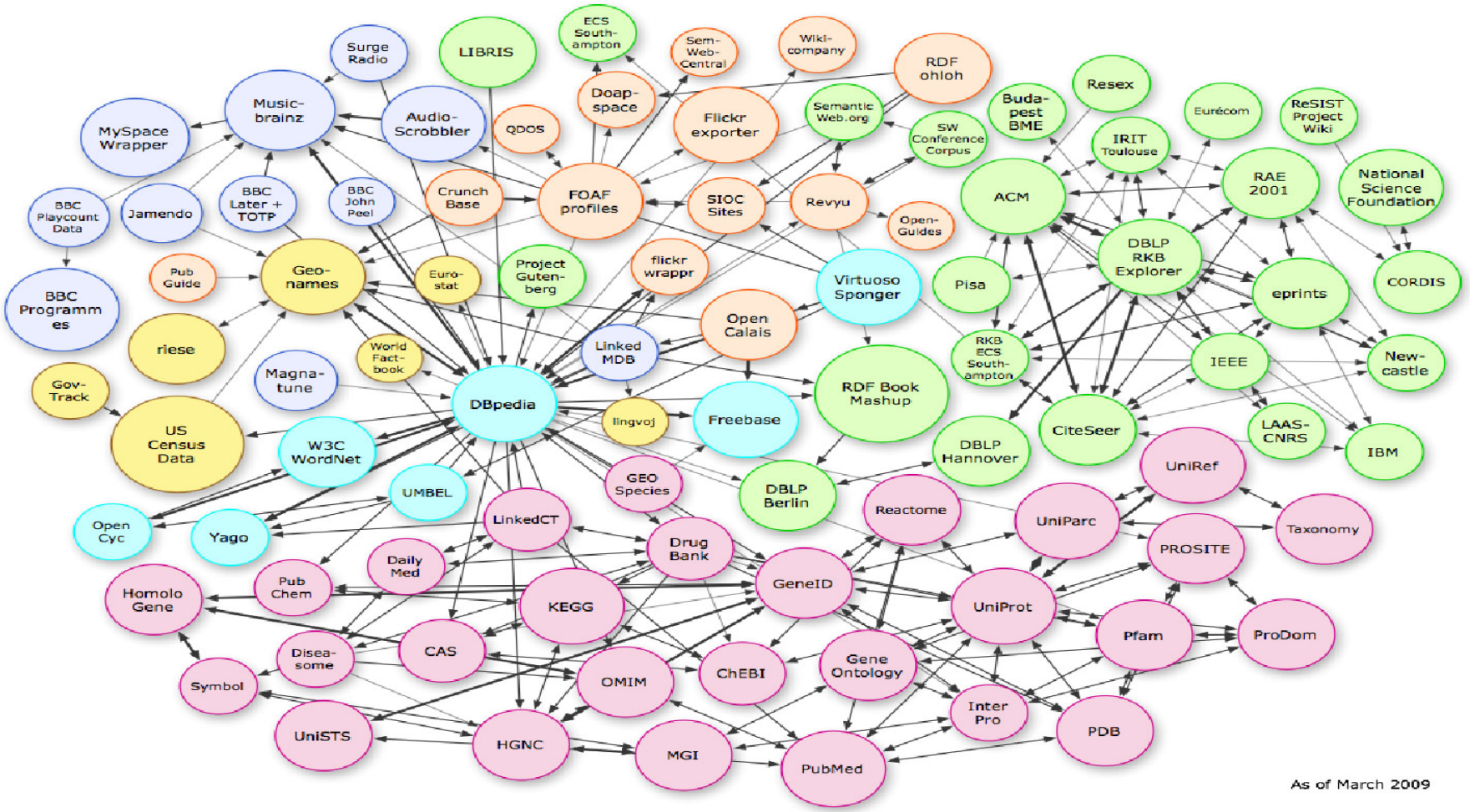
Vocabularies:

```
:LiveJournal a void:Dataset;
            void:vocabulary <http://xmlns.com/foaf/0.1/>; .
```

VoID Structural Metadata – Dataset Statistics

Property	Purpose
`void:triples`	The total number of triples contained in the dataset.
`void:entities`	The total number of entities that are described in the dataset. To be an entity in a dataset, a resource must have a URI, and the URI must match the dataset's `void:uriRegexPattern`, if any. Authors of VoID files may impose arbitrary additional requirements, for example, they may consider any `foaf:Document` resources as not being entities.
`void:classes`	The total number of distinct classes in the dataset. In other words, the number of distinct class URIs occuring as objects of `rdf:type` triples in the dataset.
`void:properties`	The total number of distinct properties in the dataset. In other words, the number of distinct property URIs that occur in the predicate position of triples in the dataset.
`void:distinctSubjects`	The total number of distinct subjects in the dataset. In other words, the number of distinct URIs or blank nodes that occur in the subject position of triples in the dataset.
`void:distinctObjects`	The total number of distinct objects in the dataset. In other words, the number of distinct URIs, blank nodes, or literals that occur in the object position of triples in the dataset.
`void:documents`	If the dataset is published as a set of individual documents, such as RDF/XML documents or RDFa-annotated web pages, then this property indicates the total number of such documents. Non-RDF documents, such as web pages in HTML or images, are usually not included in this count. This property is intended for datasets where the total number of triples or entities is hard to determine. `void:triples` or `void:entities` should be preferred where practical.

Application of VoID



orbis TECHNOLOGIES, INC.

GeoVoID Vocabulary:

The Geospatial Vocabulary of Interlinked Datasets (GeoVoID) is an RDF Schema vocabulary extension of VoID for expressing metadata about RDF datasets with a geospatial aspect.

GeoVoID

geovoid:spatialCoverage: A definition of the spatial area in which all of the elements of this dataset are found.

geovoid:temporalCoverage: A property for describing the temporal extent (vs. transaction time) of items in the dataset.

geovoid:boundingArea: An enclosing geometry of the spatial area in which all of the elements of this dataset are found.

geovoid:boundingBox: Enclosing rectangle of the spatial area in which all of the elements of this dataset are found.

GeoVoID – Continued

geovoid:boundingFeature: An enclosing place in which all of the elements of this dataset are found.

geovoid:boundingGeohash: An enclosing place in which all of the elements of this dataset are found.

geovoid:spatialVocabulary: A spatial vocabulary that is used in the dataset.

geovoid:temporalVocabulary: A temporal vocabulary that is used in the dataset.

geovoid:geometryPartion: A partition for describing the types of geometries used within a dataset or within a particular class partition.

GeoVoID Surface Semantics

dc:coverage

- geovoid:spatialCoverage
  - boundingArea
    - boundingBox
  - boundingFeature
  - boundingGeohash
- geovoid:temporalCoverage

GeoVoID Surface Semantics – Continued

void:vocabulary

- geovoid:spatialVocabulary
- geovoid:temporalVocabulary

void:classPartition

- geovoid:geometryPartition

GeoVoID = 5 star Linked Data Schema ☺

Example GeoVoID Document Metadata

```
# rdf document metadata
<> a void:DatasetDescription;
    dcterms:title "Test Dataset Description";
    dcterms:description "This is a document containing VoID and
GeoVoID descriptions of an example dataset.";
    dcterms:creator <http://example.org/bob>;
    dcterms:created "04-01-2011"^^<xsd:date>;
    # can assert qualitative spatial coverage of dataset
    geovoid:spatialCoverage <http://example.org/the_earth>;
    foaf:primaryTopic <http://example.org/ds1>;
    foaf:topic <http://example.org/ds2>;
    # bounding box of dataset
    geovoid:boundingBox “POLYGON( -180 -90, 180 90)”;
    .
```

orbis TECHNOLOGIES, INC.

GeoVoID Access & Structural Metadata

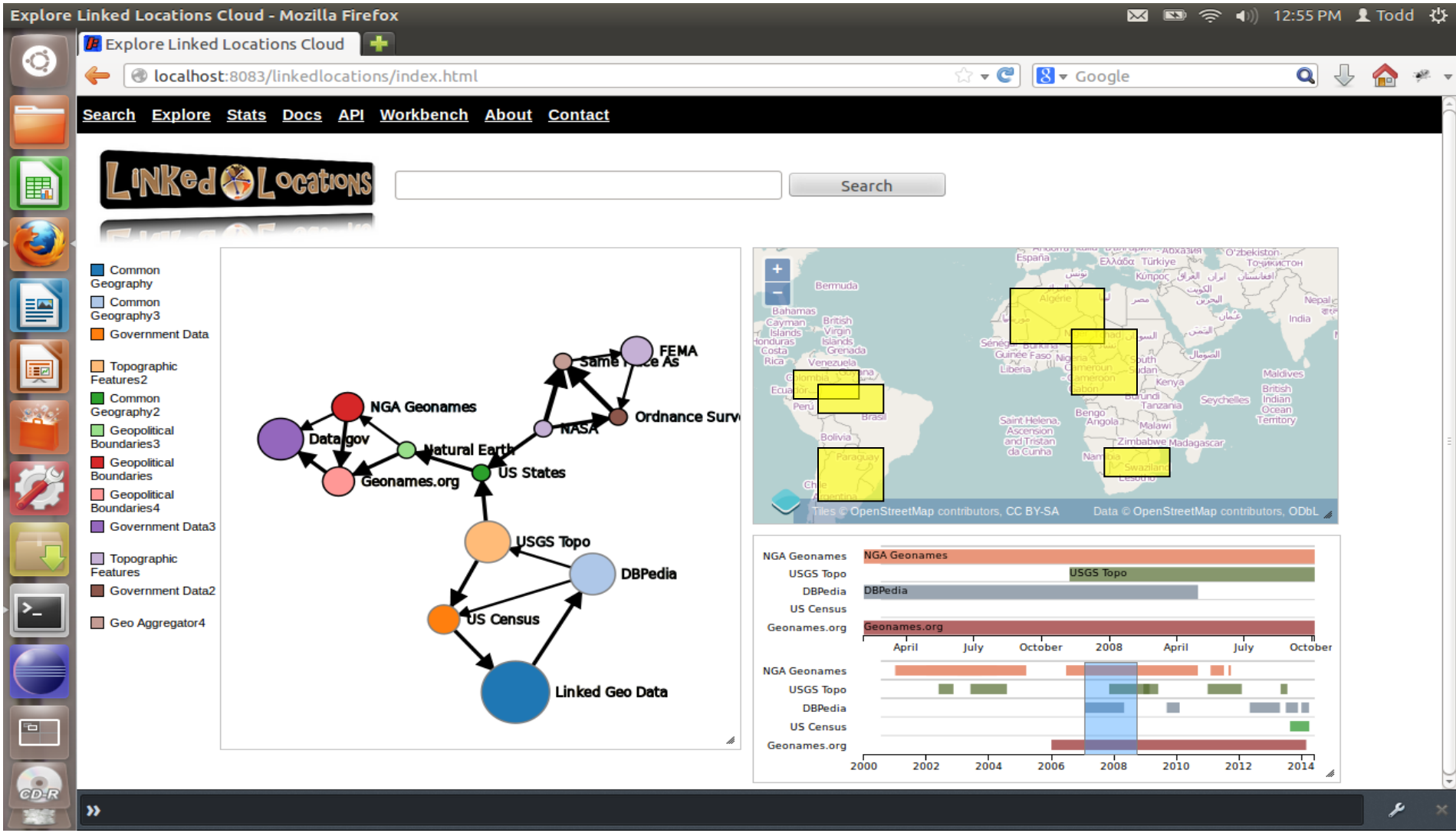
```
# access metadata; No need to redefine a “GeoSPARQL” endpoint
void:sparqlEndpoint <http://example.org/ds1/sparql/url>;
# structural metadata
# can deref to get representative schema info for geo datasets
void:exampleResource <http://example.org/ds1/example/resource1>;
void:exampleResource <http://example.org/ds1/example/resource2>;
# can discover geovocabs used in geo datasets
void:vocabulary <http://example.org/vocab1>;
void:vocabulary <http://example.org/vocab2>;
# a subset combined with a spatial extent = spatial partition
void:subset <http://example.org/ds1/part1>;
void:subset <http://example.org/ds1/part2>;
# number of geo features in geo dataset
void:entities 33123;
void:triples 10500444;
```

GeoVoID Class & Property Partitions

```
void:classPartition [
    void:class <http://example.org/ont#Road>; # Road = Feature Type
    void:entities 95;   # Number of Road features
    # schema partitions for Road feature type
    void:propertyPartition [ void:property ogc:disjoint; ];
    void:propertyPartition [ void:property rdfs:label; ];
    # geographic feature type partitions can have geospatial extents
    geovoid:boundingBox “POLYGON(-180 -90, 180 90)”;
    geosparql:rcc8-ntpp <http://example.org/the_whole_wide_world>;
    # geometry partitions for Road feature type
    void:classPartition [
        void:class <http://www.opengis.net/rdf#LineString>;
        void:entities 95; ];
    void:classPartition [
        void:class <http://www.opengis.net/rdf#Polygon>;
        void:entities 29; ]; ];
```

Application of GeoVoID

Application of GeoVoID



Next Steps

- Finish GeoVoID generator software
  - Test assortment of clustering techniques
- Apply GeoVoID to Semantic Web Datasets
  - Empirical evidence of the Geospatial Semantic Web!
- Host dataset descriptions in GeoVoID-store
  - Service and app via Linked Locations
- GeoSPARQL Service Descriptions
  - Hopefully we'll create here in Madison ☺

Thanks!

GeoVoID URI:
http://purl.org/geovocamp/ontology/geovoid

Additional Thanks:

Dave Kolas

GeoVoCamp-SantaBarbara-2014