Semantic MOBY Overview

Gary Schiltz, NCGR
gss@ncgr.org

October 27, 2004

W3C workshop on Semantic Web for Life Sciences

NATIONAL CENTER FOR GENOME RESOURCES

What is MOBY?

- An effort to facilitate the integration of web-based bioinformatics resources
- Name is from an acronym (MOBY-DIC): Model Organism Bring Your own - Database Interconnectivity Conference
- Two complementary open source projects
  - MOBY Services (web services oriented)
  - Semantic MOBY (semantic web oriented)

What is Semantic MOBY?

- ⌘ An architecture and set of conventions for building and deploying distributed, web-based software applications [in the field of bioinformatics]
- ⌘ A prototype reference implementation of the major architectural components
    - A discovery server to enable consumers of bioinformatics resources to programmatically find such services
    - A set of Java classes to facilitate building repeating architectural components
    - A brokering service to enable end users (biologists) to use web browsers to query the discovery server, engage appropriate services, and display results

Semantic MOBY components (roles)

- *Ontology Providers* define shared RDF classes and properties
- *Service Providers* perform data transformation and retrieval
- *Service Consumers* engage service providers to gain access to their services
- *Discovery Servers* discover shared ontologies and service providers, and match requests from service consumers with service providers
- *Invocation Brokers* provide browser-based interfaces to search for and engage providers

Ontology Providers

Ontology Providers define RDF classes and properties

⌘ Definitions retrieved via HTTP GET of “slash-style” URL

Example: http://brebiou.cshl.org/ontologies/Panel

(N3, prefixes omitted)

```
cshlterms:Panel
    a owl:Class.
```

Example: http://www.semanticmoby.org/ontologies/core/Provider

```
moby:Provider
    a owl:Class ;
    rdfs:subClassOf [
        a owl:Restriction ;
        owl:onProperty moby:operatesOn ;
        owl:minCardinality "1"^^xsd:nonNegativeInteger
    ] .
```

Service Providers

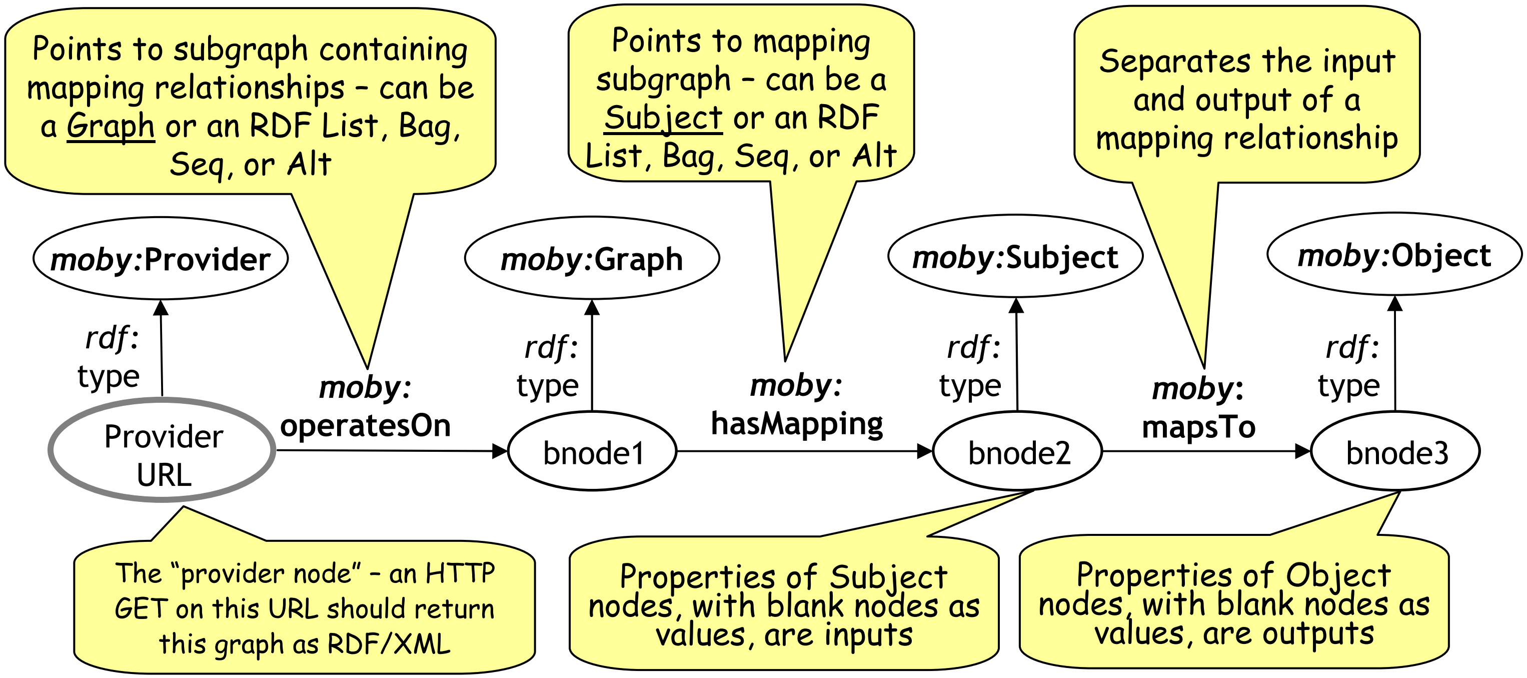
Service Providers perform data transformation and retrieval services

- *Provider description graph* is an RDF graph that defines the provider
    - Retrieved by HTTP GET of “slash-style” URL
    - Canonical graph structure [see next slide] defines allowable inputs and outputs in context of larger graph structure
- Service is engaged by HTTP POST of modified provider description graph to provider’s URL

Service Providers (cont.)

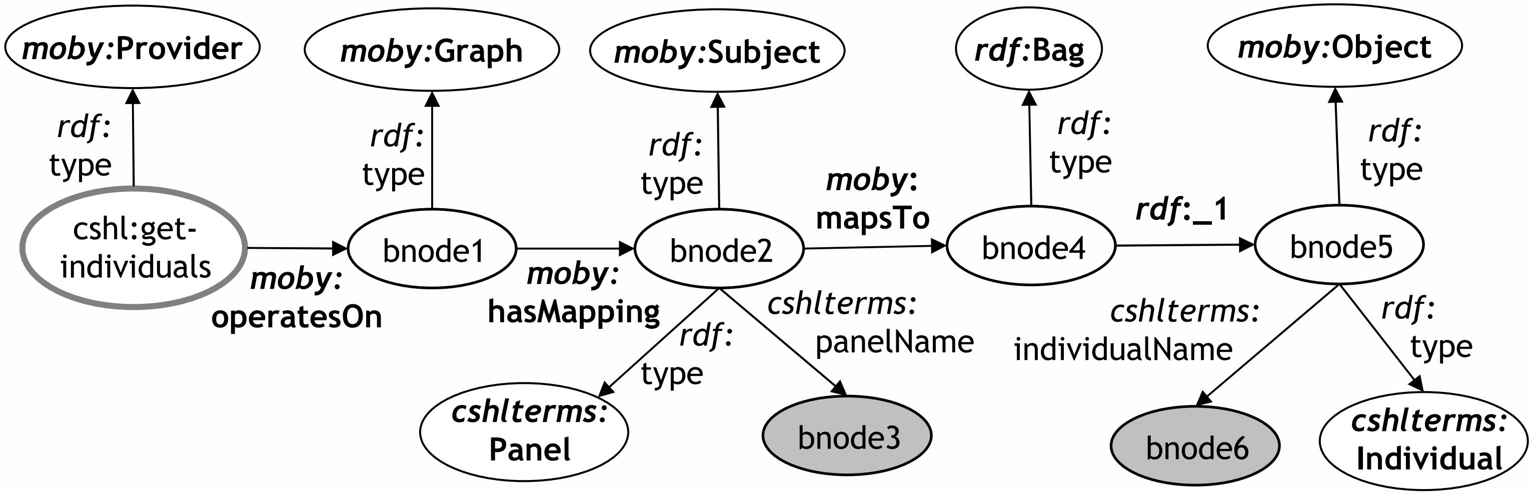
Canonical Structure of Provider Description Graphs

- ⌘ Defines providers that are compliant with architecture
- ⌘ Facilitates writing tools to parse graph into higher-level objects that define associative relationships (“mappings”)
- ⌘ Identifies inputs and outputs in context or larger graph structure


Points to subgraph containing mapping relationships – can be a Graph or an RDF List, Bag, Seq, or Alt
Points to mapping subgraph – can be a Subject or an RDF List, Bag, Seq, or Alt
Separates the input and output of a mapping relationship
moby:Provider
moby:Graph
moby:Subject
moby:Object
rdf: type
rdf: type
rdf: type
rdf: type
moby: operatesOn
moby: hasMapping
moby: mapsTo
Provider URL
bnode1
bnode2
bnode3
The “provider node” – an HTTP GET on this URL should return this graph as RDF/XML
Properties of Subject nodes, with blank nodes as values, are inputs
Properties of Object nodes, with blank nodes as values, are outputs

Service Providers (cont.)

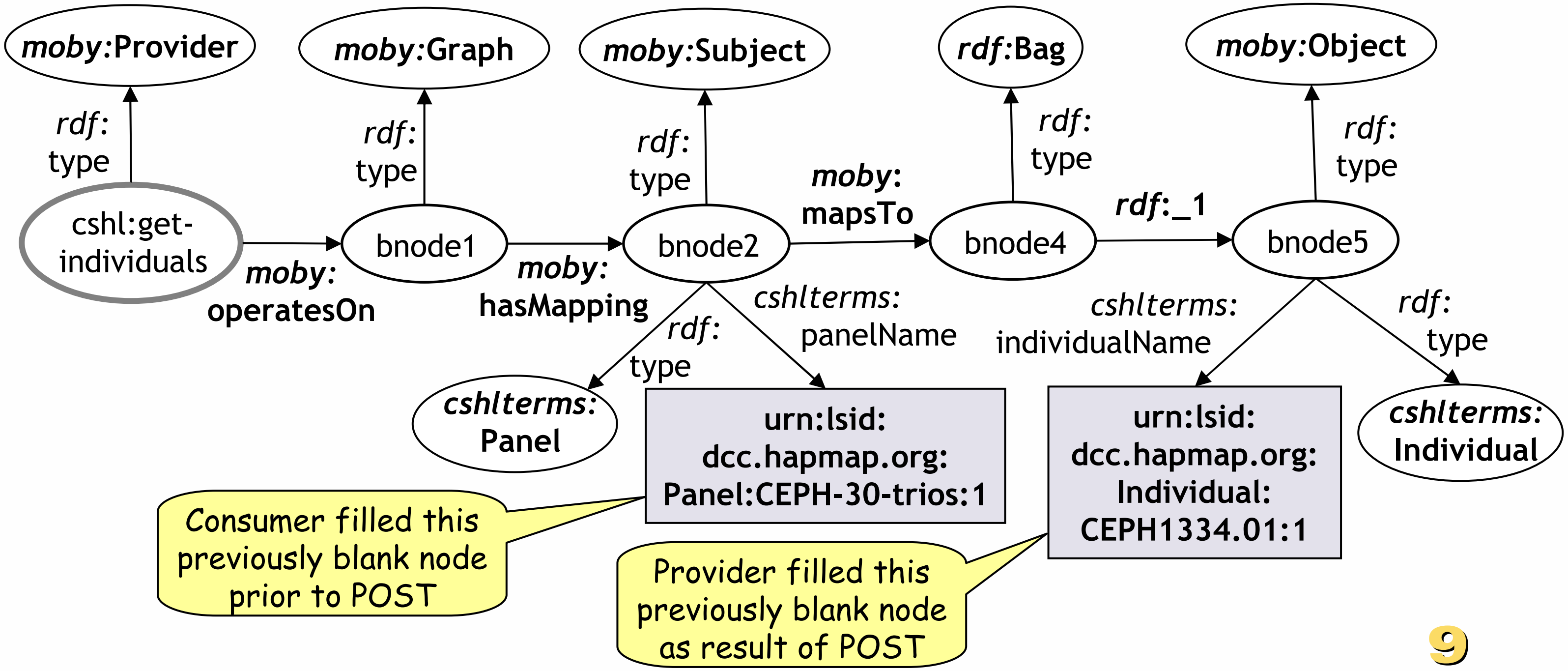
Example: http://brebiou.cshl.org:8080/get-individuals



Service Consumers

Service Consumers make use of service providers

- Engage by filling "input" blank nodes in provider graph and sending to provider URL by HTTP POST
- Provider reads inputs from graph, performs its action, fills blank nodes, returns modified graph as POST result



Consumer filled this previously blank node prior to POST

Provider filled this previously blank node as result of POST

Discovery Servers

Discovery Servers index shared ontologies and service providers, and match requests from service consumers with service providers.

In the prototype reference implementation:

- ⌘ Graphs are stored using Jena2
- ⌘ Queries for matching graphs are expressed as RDF graphs
- ⌘ Discovery server converts queries to RDQL
  - Blank nodes become RDQL variables
  - Query is executed
  - Each result set binding is used to create a copy of the query graph as a member of the set of query results

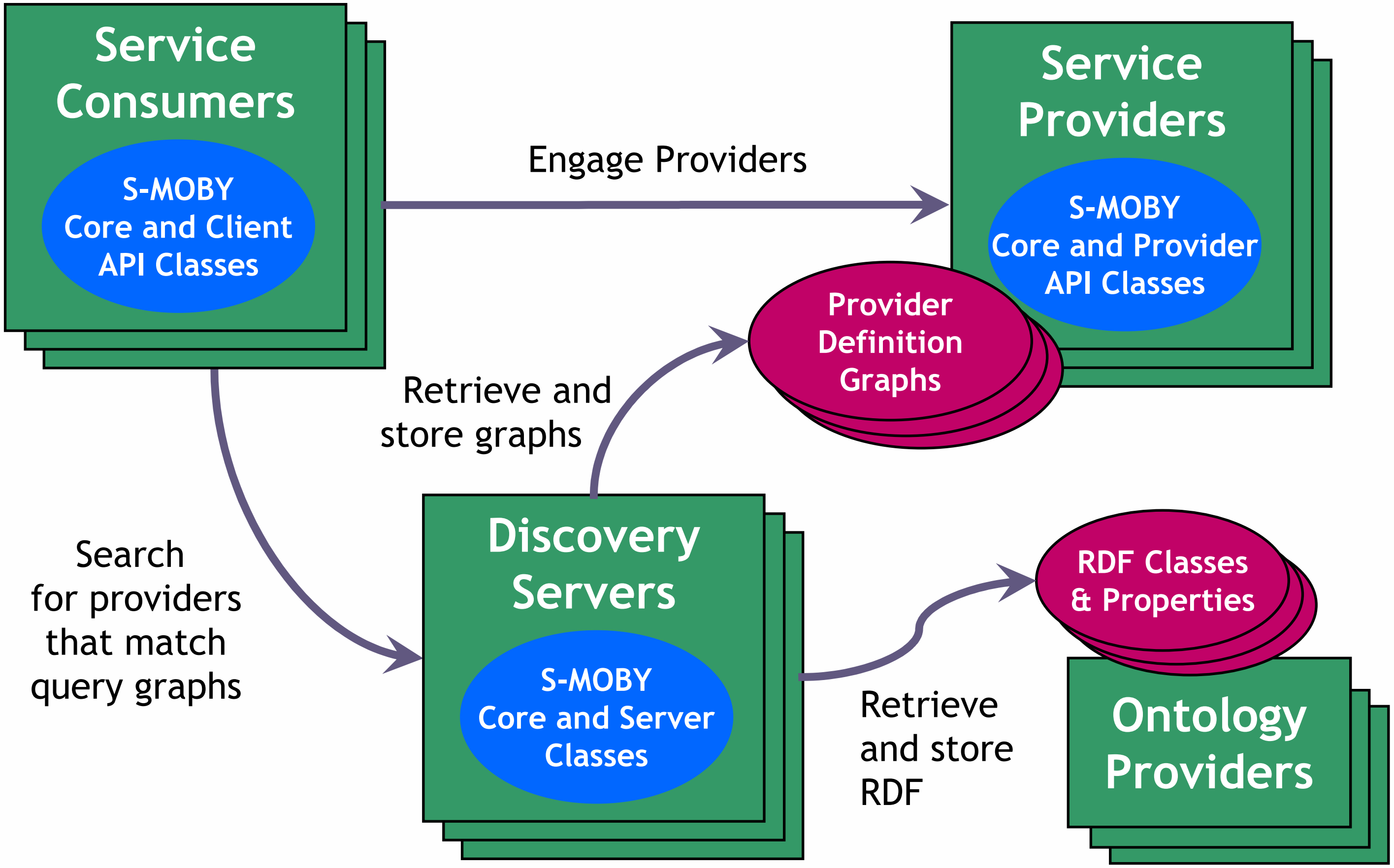
Invocation Brokers

Invocation Brokers provide browser based interfaces to search for and engage service providers

In the prototype reference implementation:

- When a provider graph is discovered, its **`Provider`**, **`Subject`**, and **`Object`** nodes are examined for other `rdf:type` properties; any moby:keyword properties of the classes are associated with the provider graph
- The user can search for providers by keyword by **`Provider`**, **`Subject`**, and **`Object`**
- For each matching provider, a link to the Semantic MOBY invocation service is provided
  - Selecting providers with a valid **`moby:inputURI`** redirects to that URI for collecting necessary inputs to the provider
  - Selecting providers without a valid input URI generates an input GUI on the fly [not yet implemented]

Semantic MOBY Architecture


Service Consumers
S-MOBY Core and Client API Classes
Engage Providers
Service Providers
S-MOBY Core and Provider API Classes
Provider Definition Graphs
Retrieve and store graphs
Search for providers that match query graphs
Discovery Servers
S-MOBY Core and Server Classes
RDF Classes & Properties
Retrieve and store RDF
Ontology Providers

Semantic MOBY Current Status

- Reference implementation prototype completed
- Java API for working with graphs is currently read only; must use Jena API to modify graphs
- A few bugs to fix :-)
- Little funding remains :-) :-)
- Distributed under Perl Artistic License - see www.biomoby.org
- MOBY Autumn 2004 meeting in Santa Fe – see www.semanticmoby.org/meeting