

CGIAR Generation Challenge Program

Web Service Working Group

Comparison of LSID and BioMOBY technologies
and suggestions for GCP implementation of these
technologies

Mark Wilkinson

Martin Senger

Ben Good

Alex Garcia

Jon Mendoza

Mathieu Rouard

July 6-9, 2004

preliminary recommendations

CONTENTS

Syntax of an LSID

Syntax of a MOBY Object

Properties of an LSID

Properties of a MOBY Object

Fundamental differences between MOBY and LSID systems

Using LSIDs and MOBY Objects in the Generation Challenge Program

Example Scenario

Syntax of an LSID:

urn:lsid:ncbi.nlm.nih.gov:gi:163483:2

urn	naming domain
lsid	naming schema
ncbi.nlm.nih.gov	authority
gi	namespace (a domain in which there are identifiers)
163483	identifier
2	version

Syntax of a MOBY Object:

```
<GenericSequence namespace="NCBI_gi" id="163483" >  
... (data)  
</GenericSequence>
```

GenericSequence	Datatype (Class/Representation)
namespace=' NCBI_gi'	Semantic Type (a conceptual type of data)
id=163483	Identifier (with version concatenated)

Properties of an LSID:

- Primary function is to be an identifier
- opaque (*no semantic information*)
- carries no data
- globally unique
- has an API for resolution to data or metadata
- Cannot be “anonymous” (by definition)

Properties of a MOBY Object:

- *Primary function is to represent data & semantics*
- May carry two pieces of semantic information
 - Datatype (e.g. GenericSequence)– semantic description of the contents of the object. This piece of information must exist.

- Semantic Type (e.g. namespace="NCBI_gi") – description of the underlying data entity that is being represented in the Datatype. This piece of information is optional.
- Can be “anonymous”, carrying data that is not represented in any data entity anywhere else on the Internet (namespace and id are NULL)
 - This *may be* useful for data that is transient, or resulting from an analytical outcome where the output data is not stored.
 - The BioMOBY specification encourages retention of input namespace/id information in the output data object whenever possible.
- API exists, “MOBY Central”, to map Objects to services (service discovery) that can operate on them
 - resolution to data
 - resolution to metadata
 - analytical services
- API exists to describe service interfaces in a machine-readable way.

FUNDAMENTAL DIFFERENCE BETWEEN MOBY AND LSID

- Both LSID identifiers and MOBY Objects are able to be resolved to the data they represent... however:
 - The MOBY System is also able to discover and invoke analytical/transformational services, and *depends on the semantics of the MOBY Object for both the data retrieval and analysis functions*. This is accomplished through mapping ontologies of datatypes and semantic types onto data instances.
 - The LSID resolver service is able to retrieve the data and metadata identified by an LSID *without additional semantic information*, but has no API for the invocation of analytical services using this data.

Using LSIDs and MOBY Objects in the Generation Challenge Program:

Three rational services can be provided by any service provider:

- | | | |
|------------------------|----|---------------------|
| 1. Consume LSID | -> | Produce Raw Data |
| 2. Consume LSID | -> | Produce MOBY Object |
| 3. Consume MOBY Object | -> | Produce MOBY Object |

Service providers are encouraged to use whichever is most suitable for their circumstance. The following guidelines will assist in choosing the most reasonable option:

An LSID arriving at a resolver with a `getData` call should produce a MOBY Object in all cases where the output data is “analyzable”; i. e. if it would be useful to pass such data onto another analytical service, or to retrieve additional cross-referencing data about it from a third party, using the MOBY Central discovery system. If the data is for display only (in that context), then there is no reason to output MOBY Objects from the LSID `getData` call. The same is true if you are displaying data from a non-GCP member who will not be outputting MOBY Objects. If you have an LSID that is resolvable outside of the GCP program, it will likely resolve to raw data. If you want to use this in the MOBY system you should wrap this raw data into a new MOBY object such that it can be used by the MOBY discovery system.

An LSID arriving at a resolver with a `getMetadata` call should return Metadata preferably in the form of LSIDs that can themselves be used to retrieve data and/or metadata. This will be particularly useful in cases where the service provider is producing raw data in response to a `getData` call, since the LSIDs in the metadata can be used to accentuate (through `getData` calls) the raw data provided.

MOBY Objects arriving at MOBY Services should be treated as per the MOBY-S API (see <http://www.biomoby.org>)

Example Scenario:

I am an LSID resolution service provider. I receive an LSID representing a Germplasm ID (GID):

```
urn:lsid:cgiar.org:GID:11743
```

with a `getData` call. I have decided to return a (hypothetical) `BasicGermplasmObject` MOBY Object from this `getData` service. My output might look like this:

```
<BasicGermplasmObject namespace="CGIAR_GID" id="11743">
  <CGIAR_Passport namespace=" " id="11743" title="Name='Passport' ">
    <String namespace=" " id=" " articleName=" Location" >IRRI </String>
    <String namespace=" " id=" " articleName=" Taxonomy" >Oryza sativa</String>
  </CGIAR_Passport>
</BasicGermplasmObject>
```

And a browser or other client might simply print the following to the screen:

```
CGIAR_GID: 11743
Location: IRRI
Taxonomy: Oryza sativa
```

I am then passed the same LSID with a `getMetadata` call. The RDF Triples I return might look like this:

```
prefixes {gid = urn:lsid:cgiar.org:GID:
ncbi = urn:lsid:ncbi.nlm.nih.gov:taxon:
loc = urn:lsid:cgiar.org:Location:}
```

gid:11743	has_Location	loc:12
gid:11743	has_taxon	ncbi:9886
gid:11743	derived_from	gid:11774

Notice that, in the response to the `getData` call, I have created a data Object that has literal values (e.g. "IRRI") in its content, while in the response to the `getMetaData` call I have passed identifiers – i.e. references – to the underlying data represented by those literals. In this way, I am able to provide simple, useful, readable output data from with the `getData` call – data which can be further analysed by calls to other BioMOBY services, and pass references to other, possibly complex, data objects from the `getMetaData` call. the `loc:12` LSID could be resolved to data or metadata with additional information about IRRI. The `ncbi:9886` LSID could be resolved to data or metadata for more information about *Oryza*, and the `gid:11774` LSID could be resolved to data or metadata for more information about the derivation of that germplasm. This gives client programs a very broad scope of data retrieval possibilities, allowing for a "surfing" of metadata using the LSID system, and the resolution of a reference to its literal value for further analysis using the BioMOBY system.