intriguing
arousing the curiosity or interest
but also
making secret plans to do something illicit

Biomoby
Integration and intriguing semantics

Martin Senger
martin.senger@gmail.com
First thing first...

• There are two Biomoby branches
  – this update is about “Moby-S” (Moby Services)
  – the other one is “S-Moby” (Semantic Moby)
    • [http://semanticmoby.org/](http://semanticmoby.org/)

• Acknowledgement
  – Mark Wilkinson, PI and creator of Biomoby
  – many groups around the world working with and for Biomoby, e.g.
    • Generation Challenge Programme of the Consultative Group for International Agricultural Research
    • The PlaNet Consortium (a network of European plant databases)
    • The Australian Centre for Plant Functional Genomics
    • The National Institute for Bioinformatics, Spain (Genome Espania)

• Where to find more
  – [http://biomoby.org](http://biomoby.org)
Biomoby in a nutshell

...for those not yet initiated
I need data.

Why should I use Biomoby?

- Because you get data from hundreds of services
- Because these data and services can interoperate (exchange their data)
- Because you need to run programs to consume data (semi-)automatically
  - if you can get what you need just by clicking on web pages, you do not need Biomoby
I have data.

Why should I use Biomoby?

• Because your data can be shared (accessed by others)

• Because Biomoby helps to get your data visible (almost without programming)
  - it does not help, however, to create web pages showing your data in web browsers

• Because you can add-value to your data by linking them to other Biomoby-aware data
What, actually, is Biomoby?

- A registry (a computer) that knows where to find services around the world
- A registry (a computer) that knows what data are being served by these services, and how the data are related to each other
- A standard (a specification) telling how to access such data (how to call such services)
- Growing number of software tools (programs) that allow to provide, to get, to browse and to combine such data
- A community of dedicated (and often nice) people to help, and to have a beer with you…
Big picture

A Biomoby repository

Biomoby services

Register services

Find services

Biomoby protocol
(this is a protocol, not Mark Wilkinson)

Call (use) services

Bimoby clients
Bottom line

- **Biomoby services are your responsibility**
  - you are a service provider, you implement your service (but Biomoby project has tools to help you - *Moses* for Java, Perl libraries, …)

- **Biomoby data types are community responsibility**
  - otherwise it would limit how they can be shared and re-used
  - you are part of the community: register your data types
What is registered

- **Ontology 1:** **Data types**
  - What data represent and how they are related
  - They all sit in one hierarchical tree (ISA)
  - They have children
    - HAS (more of this kind)
    - HASA (maximum one)
What is registered

• Ontology 2: Namespaces
  - define the scope of your data
    • geographically (where a database is located)
      - e.g. “NIAS_OryzaMutant”
    • semantically (what kind of database data are in)
      - Example: If you have a datum identified by a string “163483”, you have no clue what it is, unless you say “the namespace is “NCBI_gi”. Another example of a namespace is “ICIS_Germplasm”.
  - no hierarchy – just a plain control vocabulary
What is registered

- Ontology 3: **Service Types**
  - a hierarchy of all kinds of services
  - it helps to discover your service
  - it is not yet mature enough
    - changes expected here
    - collaboration with myGrid,…
What is registered

• **Ontology 4: Services**
  - where they are (an endpoint)
  - where to find more about them (a URL with an RDF document that is partly maintained by the service provider)
  - what input and output data they can consume and provide
Biomoby major trick how to gain interoperability between services

• Each service must understand data type as declared in the registry
  – this is usual

• Each service must be able to ignore more specific data, if they come, and not to break itself on them
  – this is usual in programming languages but it is not that common in Web Services world
  – it is possible because data types are related in a hierarchy
Biomoby update

...what happened over the past year
New funding...

- Keep and enhance current Biomoby
- Research on Biomoby 2
  - more about service discovery
    - semantics as a hype or reality?
  - cautious approaches to S-Moby
    - could we have just one?
jMoby: Biomoby for Java

• major pieces are
  – Java libraries (API) for accessing registry
    • Central.java
  – Generators of Biomoby service skeletons
    (MoSeS = Moby Services Support)
    • a framework that you extend by your own implementation to create your own services
    • coming soon: fully generated services accessing data using BioCASE, Soaplab and Hibernate
      – no need to write any implementation code for services
  – Dashboard…
To write a Biomoby service, one needs:

- To extract data from a SOAP envelope
- To expect incoming data in different encoding (data can be a String or a byte array)
- To extract data from a Biomoby XML envelope
- To separate data into individual “jobs” (a request can consist of many of them)
- [To get installation parameters from the surrounding servlet engine]
- To do something meaningful with data (to create results)
- To convert results back into response “jobs”
- To wrap results into a Biomoby XML
- To send data back in a SOAP envelope

Done by SOAP toolkit (e.g. Apache Axis)
Done by you ("business logic")
Done by Moses
An example: a full Biomoby service “HelloBiomobyWorld”

```java
package org.jmoby.tutorial.service;

import net.jmoby.samples.HelloBiomobyWorldSkel;
import org.biomoby.shared.MobyException;
import org.biomoby.shared.parser.MobyPackage;
import org.biomoby.shared.parser.MobyJob;
import org.biomoby.shared.datatypes.*;

public class HelloBiomobyWorldImpl
    extends HelloBiomobyWorldSkel {
    public void processIt (MobyJob request, MobyJob response,
                           MobyPackage outputContext)
        throws MobyException {
        set_greeting (response, new MobyString ("Hello, World!"));
    }
}
```
There is definitely more...

- Biomoby plug-in to Taverna
- Asynchronous service invocation
- Perl-Moses
- ...

...
Summary
What is Biomoby good at...

- It has many running services
- It provides data models in a reasonably flexible way
- It has a potential to discover services in a modern way!
  - see also “MOBY 2” and Semantic Moby
- It has a potential to annotate services in a non-centralised way
What is Biomoby less good at…

• It has many crapped services

• It does not use fully potential of Web Services (WSDL etc.)
  – perhaps it does not need to be SOAP-based at all (the pure HTTP can do the same here)

• The potential for service discovery by reasoning yet to be proved
Thank you...