

# Passport Data Module <sup>SP4</sup>



# Steps

1. MCPD and use cases
2. Data Models
3. Data Models
4. Prototype implementation details
5. Prototype implementation details

# Multi Crop Passport Data

## Passport information

Accession name	100127.
Species	oryza glaberrima .
Institute	IRRI.
Collection	IRRI - Oryza glaberrima Collection.
Country of origin	Gambia (Sub Saharian Africa).
Collecting mission	N/A.
Collecting date	0000-00-00.
Collecting number	AF 94-7.
Collecting site	.
Elevation (m)	0.
Latitude	0.
Longitude	0.
Donor Information	.
Acquisition date	.
Collecting source	Unknown collecting source.
Biological status	Unknown Sample Status.
In trust status	In trust
Other Numbers	
Remark	Designated (1994-09-14 00:00:00).

## Collecting mission information

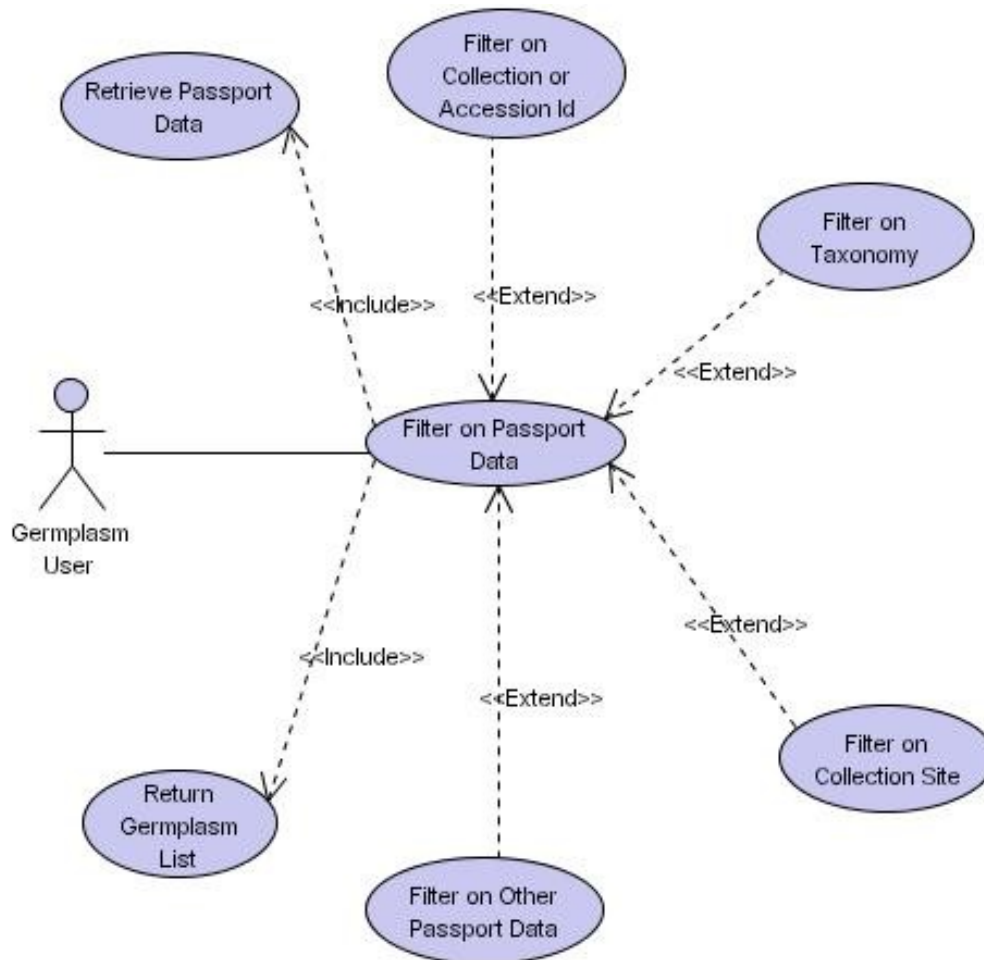
No cooperators recorded

## Distribution information

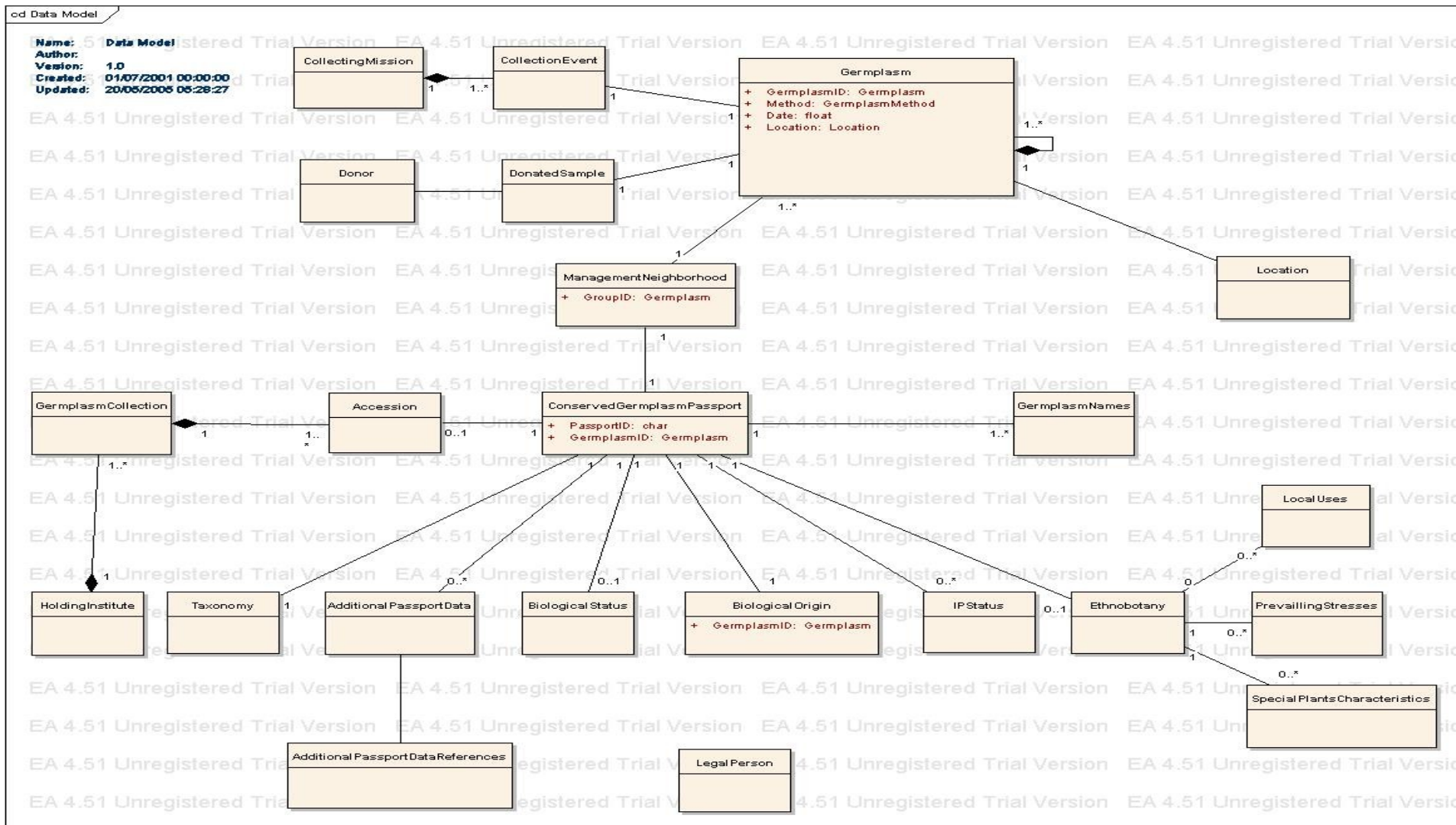
Date	1993-08-05
Cooperator	TAODAYUN.
Organization	YUNNAN ACADEMY OF AGRICULTURAL SCIENCES, YUNNAN.
Type of cooperator	National Agric. Research Ser..

- International standard developed jointly by IPGRI and FAO.
- List of descriptors to facilitate germplasm passport information exchange across crops.
- These descriptors have been used to provide data templates within GCP.

# Use cases

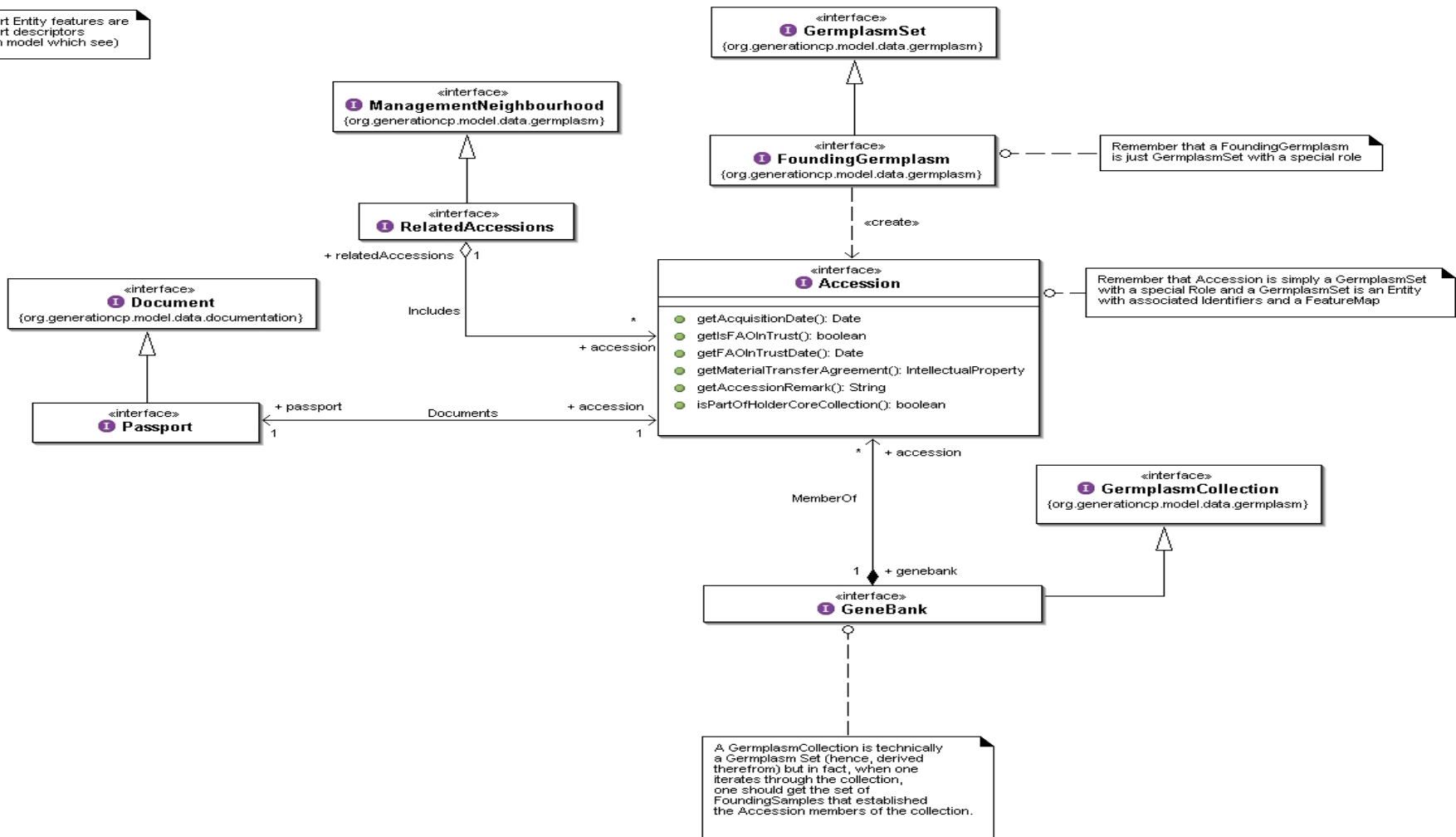


# Data Model (Phase 1)



# Data Model (Phase II)

Passport Entity features are passport descriptors (domain model which see)

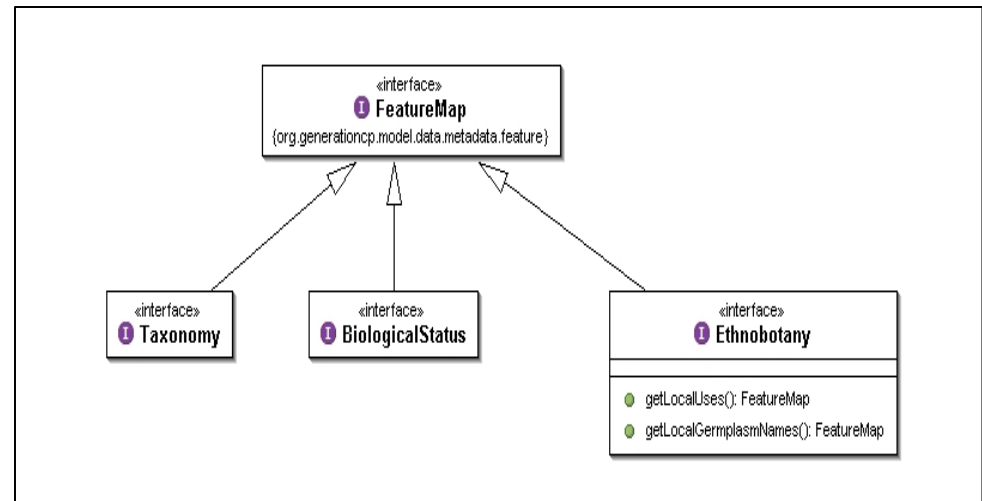
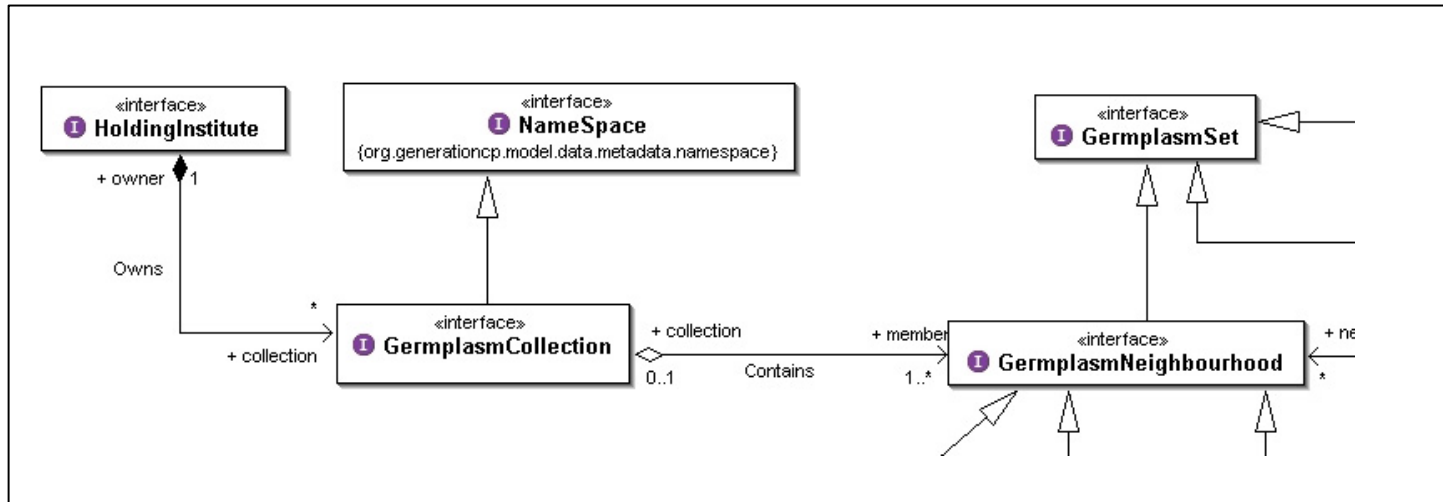


Remember that a FoundingGermplasm is just GermplasmSet with a special role

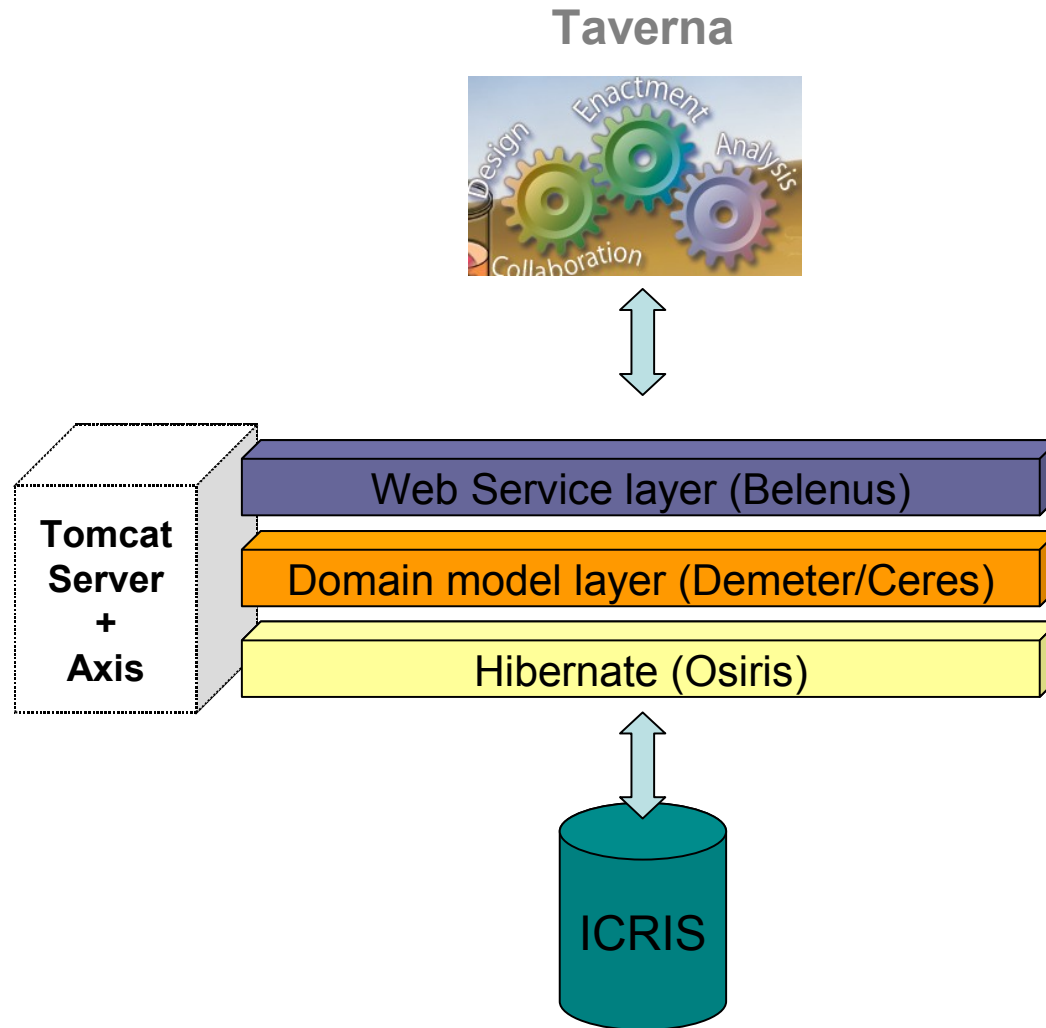
Remember that Accession is simply a GermplasmSet with a special Role and a GermplasmSet is an Entity with associated Identifiers and a FeatureMap

A GermplasmCollection is technically a Germplasm Set (hence, derived therefrom) but in fact, when one iterates through the collection, one should get the set of FoundingSamples that established the Accession members of the collection.

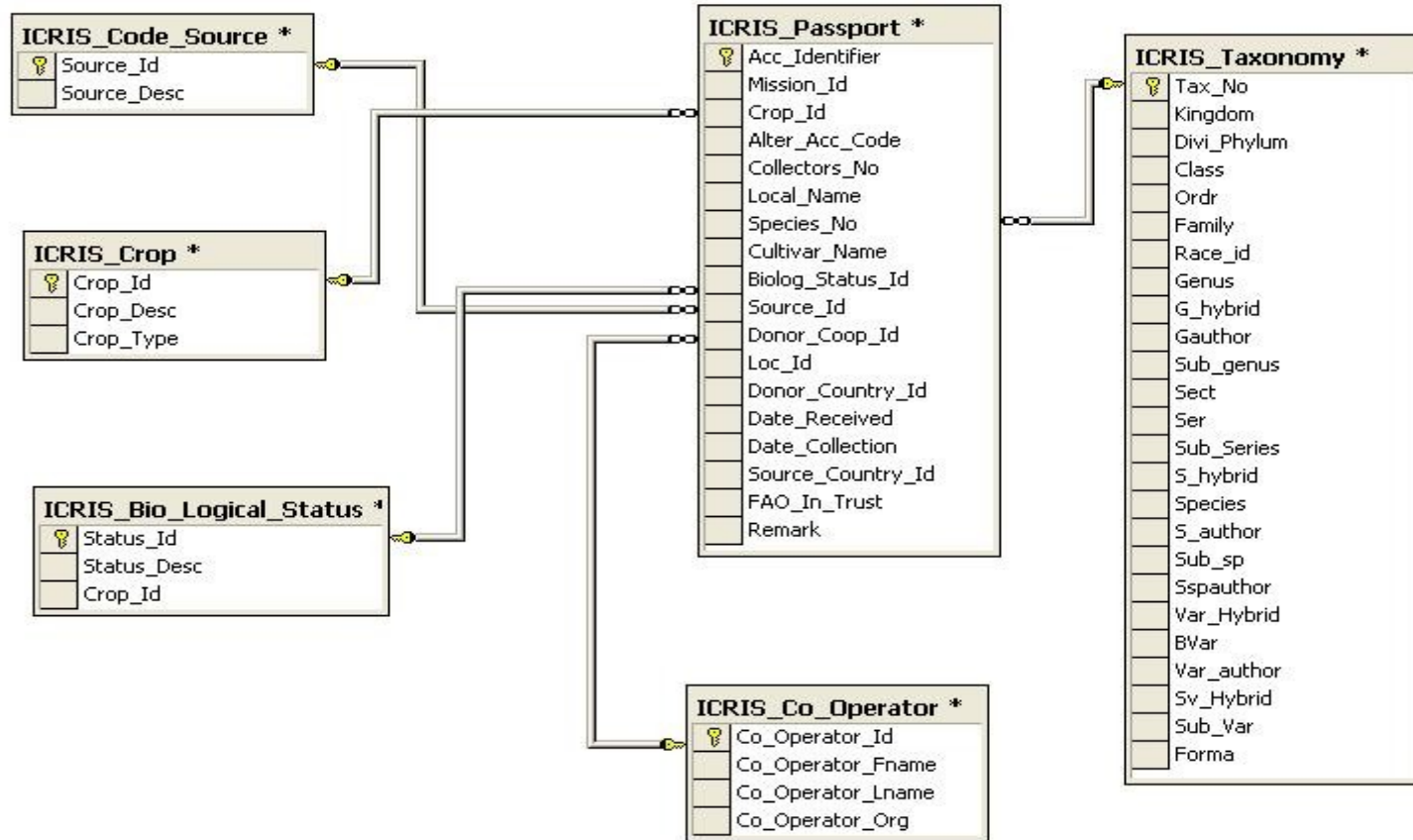
# Data Model (Phase II)

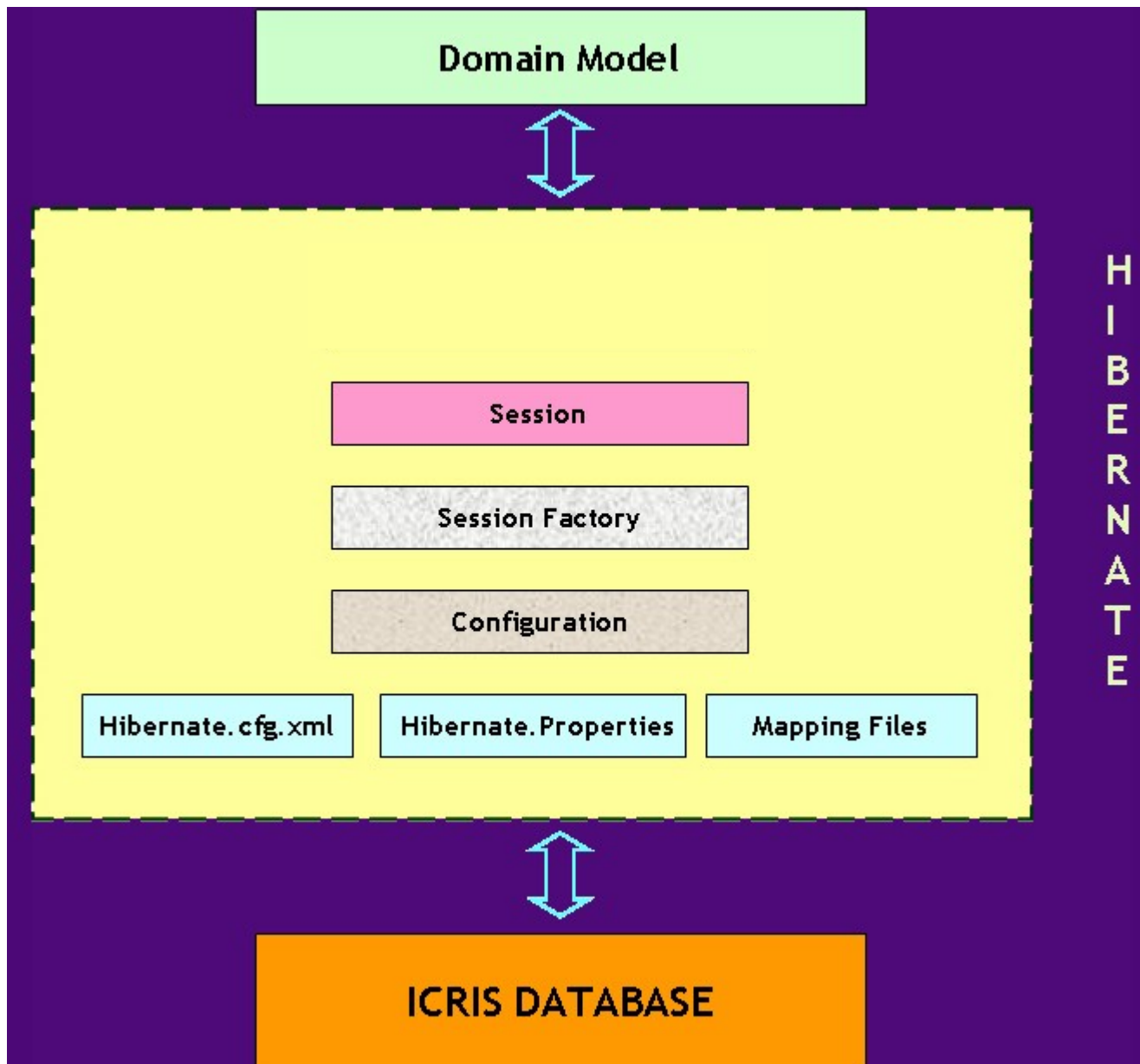


# Passport Data Module prototype



# ICRIS tables with passport data





# Demeter

```
GermplasmSetClient.java | GermplasmSet.java | GermplasmClient.java | Germplasm.hbm.xml | Germplasm.java x
```

```
private String gID;
private String taxonomy;
private String id;
private String CropName;
private String LocalName;
private String CultivarName;
private String SourceCode;
private String ReceivedDate;
private String CollectionDate;
private String FAO;
private String Remarks;
private String bioStatusId;
private String LocationId;
private String CollectionNo;
private String DonorId;

/**
 * @return Returns the collectionNo.
 */
public String getCollectionNo() {
    return CollectionNo;
}

/**
 * @param collectionNo The collectionNo to set.
 */
public void setCollectionNo(String collectionNo) {
    CollectionNo = collectionNo;
}

/**
 * @return Returns the donorId.
 */
public String getDonorId() {
    return DonorId;
}
```

Writable | Smart Insert | 1 : 1

# CERES

J GermplasmSetClien... J GermplasmSet.java J GermplasmClient.java X RML Germplasm.hbm.xml J Germplasm.java J HibernateSessionF... RML hibernate.cfg.xr

```
try{

    HashMap htb=new HashMap();
    ArrayList alGermplasm = new ArrayList();
    Session session = HibernateSessionFactory.currentSession();
    Transaction tx = session.beginTransaction();
    String qry = "from Germplasm germ where germ."+key1+" "+filter+" :strId";
    System.out.println(qry);
    List GermplasmSet = session.createQuery(qry).setString("strId",key3).list();

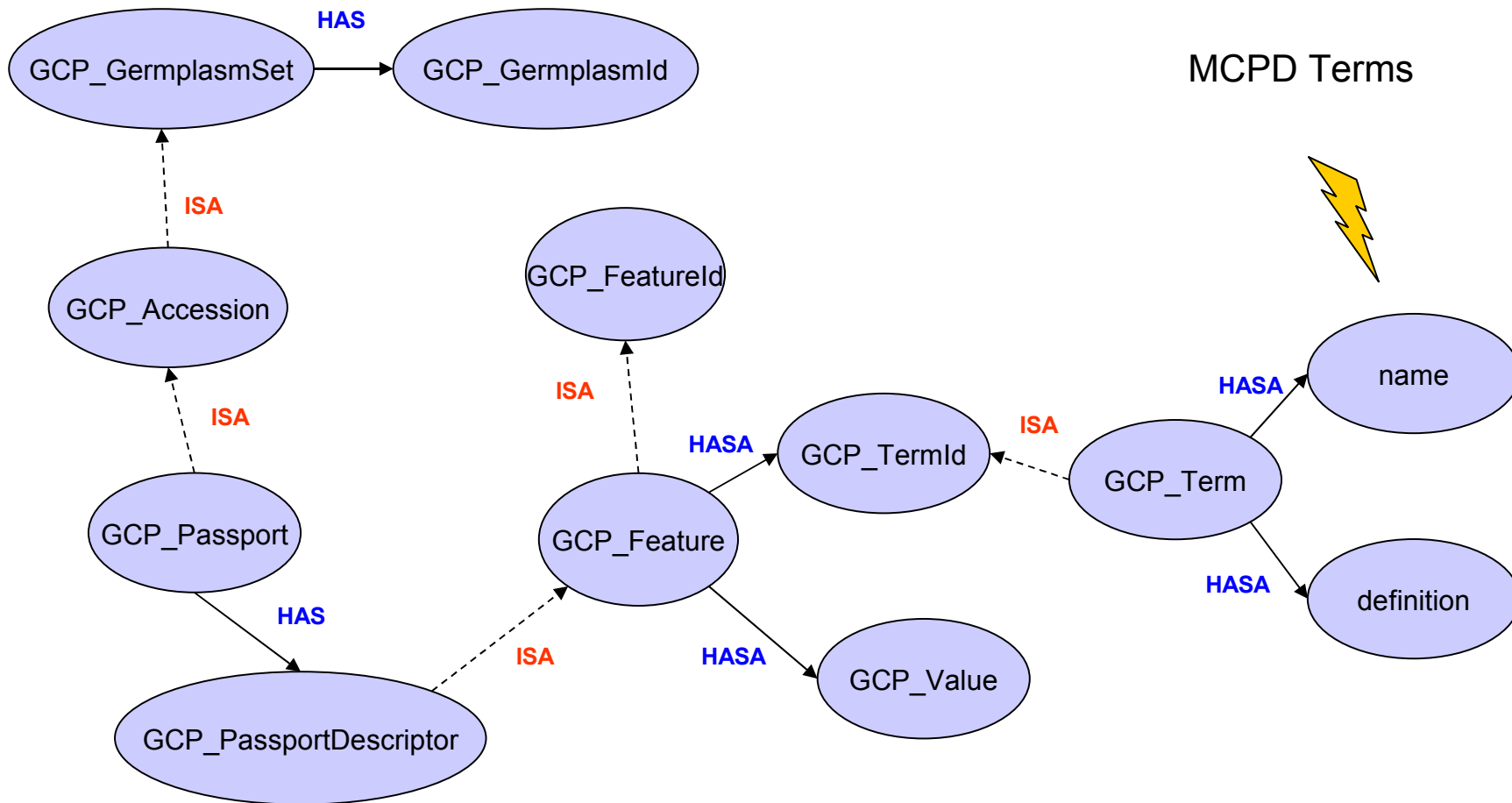
    for (Iterator iter = GermplasmSet.iterator(); iter.hasNext();)
    {
        Germplasm germplasm = (Germplasm) iter.next();
        htb.put ("GID",germplasm.getGID());
        htb.put ("Id",germplasm.getId());
        htb.put ("BioStatusId",germplasm.getBioStatusId());
        htb.put ("Taxonomy",germplasm.getTaxonomy());
        htb.put ("Cropname",germplasm.getCropName());
        htb.put ("LocalName",germplasm.getLocalName());
        htb.put ("CultivarName",germplasm.getCultivarName());
        htb.put ("LocationId",germplasm.getLocationId());
        htb.put ("CollectionNo",germplasm.getCollectionNo());
        htb.put ("DonorId",germplasm.getDonorId());
        htb.put ("SourceCode",germplasm.getSourceCode());
        htb.put ("ReceivedDate",germplasm.getReceivedDate());
        htb.put ("CollectionDate",germplasm.getCollectionDate());
        htb.put ("FAO",germplasm.getFAO());
        htb.put ("Remarks",germplasm.getRemarks());
        alGermplasm.add(new HashMap(htb));
    }

    tx.commit();
```

# Belenus

- Implementation of Moby services
  - Need for Moby data types
  - Need for semantics

# Translation of domain models into data types



# Web services

1. Get an accession list based on the taxonomy
3. Get an accession list based on the collection name
5. Get an accession list based on additional descriptors
7. Get passport details for an accession id
9. Get passport details for an accession list

# Dashboard

Data Type Registration
Service Registration
Namespace Registration

**New Service**

**Service name**  
 getPassport  **authoritative**

**Authority**  
 www.icris.icrisat.org

**Contact email**  
 jayashree@cgiar.org

**Service endpoint - URL**  
 http://your.server:8080/axis/services/getPassport

**Service RDF Signature**  
 Use RDF signature  
 RDF endpoint - signature URL  
 Where to store RDF document  
 C:\DOCUME~1\mrouard\LOCALS~1\Temp\service.rdf

**Service type: Retrieval**

**Description**

- nem
  - MyTestingServiceType\_1110
    - Parsing
    - Registration
    - Resolution
    - Retrieval**
    - Rub1
    - Searching
    - ServicesAdministrativos
    - SetOperation
    - Shim
    - Testing

Fill new Service when selected in browser panel

Primary Inputs and Outputs
Secondary Inputs

Add input data

	Article name	Data Type	Collection	Namespaces
<input checked="" type="checkbox"/>	GCPTemp_GermplasmId	GCPTemp_GermplasmId	<input type="checkbox"/>	ICRIS_Germplasm

Add output data


	Article name	Data Type	Collection	Namespaces
<input checked="" type="checkbox"/>	GCPTemp_Passport	GCPTemp_Passport	<input checked="" type="checkbox"/>	ICRIS_Germplasm

**Data Types**

- Object
  - AlleleAssociation
  - Amigo
  - BasicGFFSequenceFeature
  - Boolean
  - CDNALibrary
  - ChinookServiceObject
  - CompartmentPrediction
  - Computador
  - Coordinates
  - DateTime
  - DIGDescription
  - DnaSequenceHolder
  - Edge
  - GR
  - GR\_Protein
  - GR\_REF
  - Gramene\_GR\_MUT
  - H-invDB\_cDNA
  - H-invDB\_locus
  - HAMAP
  - Haplotyping\_Study
  - HGNC
  - HSSP
  - HUGO
  - ICIS\_Germplasm
  - ICIS\_Study
  - ICRISAT
  - IMGA
  - IMGT\_HLA
  - IMGT\_HGM

# Moby client

Scufl Workbench v1.3.1, built Fri Dec 09 15:44:49 CET 2005  
Tools and Workflow Invocation

**Taverna Scufl Workbench v1.3**  
myGrid  <http://taverna.sf.net>  
djm,jf,mg,pl,ktg,mp,ms,tmo,mf,ek,pa,jb et al.

**Advanced model explorer**

Workflow: Metadata for 'getAccessionsByTaxonomy'

Load Load from web Save New subworkflow Offline Reset

Workflow object	Retries	Delay	Backoff	Threads	Critical
Workflow model					
Workflow inputs					
Workflow outputs					
Processors					
getAccessionsByTaxonomy	0	0	1	1	<input type="checkbox"/>
getAccessionPassport	0	0	1	1	<input type="checkbox"/>
getAccessionsByName	0	0	1	1	<input type="checkbox"/>
Data links					
Control links					
getAccessionsByName::getAccession					
getAccessionsByTaxonomy::getAcce					

**Workflow diagram**

Save as Configure diagram

```

    graph TD
      A[getAccessionsByTaxonomy] --> C[getAccessionPassport]
      B[getAccessionsByName] --> C
  
```

Rendering done.

**Available services**

Search list Watch loads

- GCP\_Identifier - Identifier for GCP entities
  - GCP\_AccessionId - Accession Identifier
    - GCP\_Passport - GCP Passport Details
      - GCP\_Passportdes - GCP Passport Details
  - GCP\_ChromosomeId - Identifier for chromosome entities
    - GCP\_Chromosome - Chromosome for GCP
  - GCP\_GeneId - Identifier for gene entities
    - GCP\_Gene - Gene for GCP
  - GCP\_GermplasmId - Identifier for germplasm entities
    - GCP\_GermplasmPedigreeTree - The germplasm pedigree
    - GCP\_GermplasmPreferredName - The germplasm preferred name
  - GCP\_MapAssignmentId - Identifier for map assignment entities
    - GCP\_MapAssignment - Map Assignment for GCP
      - GCP\_Locus - Locus for GCP
        - GCP\_QTL - QTL(Quantitative Trait Loci) for GCP
  - GCP\_MapId - Identifier for map entities
    - GCP\_Map - Map for GCP
  - GCP\_PhenotypeId - Identifier for phenotype entities
    - GCP\_Phenotype - Phenotype id and phenotype description
  - GCP\_StudyId - Identifier for study
    - GCP\_MapStudy - Map study for GCP
    - GCP\_StudyDetails - The germplasm study details
  - GCP\_TermId - Identifier for controlled vocabulary (ontology)
    - GCP\_Term - Full definition of a controlled vocabulary (ontology)
      - GCP\_Datatype - GCP\_Datatype
    - GCP\_Value - GCP\_Value
  - GCP\_MapPosition - The position in a map
  - GCP\_MappableElement - Mappable element for GCP
  - GCP\_Marker - Marker for GCP
    - GCP\_Trait - Trait for GCP
  - GCP\_MarkerID - Name of the marker
  - GCP\_NameQuery - A search string (possibly a wildcard) of a name
  - GCP\_Taxonomy - Gives details of genus and species

# Future work

## By May

- To finalize the prototype.

## After May

- To be discussed