

Infrastructure - Task #3103

Task # 3074 (Closed): Phase one implementaion for morpho to connect dataone services

The design to modify the AbstractDataPackage class

2012-07-26 03:08 - Jing Tao

Status:	Closed	Start date:	2012-07-25
Priority:	Normal	Due date:	
Assignee:		% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	2013.10-Block.2.1	Story Points:	
Milestone:	None		
Product Version:	*		
Description			
<p>In Morpho code, a data package is presented by an AbtractDataPackage class. This class is oriented by EML document. It has both the metadata and data information. Now Dataone introduced a new class - DataPackage. This new class is collection of D1Object. The DataPackage can be serialized as an OAI-ORE ResourceMap that details the linkages among data objects and science metadata objects.</p> <p>Now Morpho will handle ORE document as well. So Ben suggested we may reuse the Dataone DataPackage. Dataone DataPackage object will be a member field in the AbstractDataPackage. We also may developer a utility method to transform AbstractDataPackage to the Dataone DataPackage.</p> <p>Methods in AbstractDataPackage involving the Dataone API:</p> <pre>public void serialize(String location); // save action in morpho including create and update public void delete(String location);// delete action in morpho public void load(String location, String pid); // open a package</pre> <p>We also redefined the methods in DataoneDataStore (FileSystemDataStore):</p> <pre>public File read(String pid) throws FileNotFoundException, CacheAccessException; public String save(AbstractDataPackage package) throws Exception; //this method will figure out which method should be used -- create or update. public boolean delete(AbstractDataPackage package) throws Exception;</pre> <p>Here is the algorithm of how serialize method to call DataStore method:</p> <pre>abstractDataPackage.serialize(String location) { if(location == Network location == Both) { dataoneDataStore.save(this); } if(location == local location == Both) { fileSystemDataStore.save(this); } }</pre> <p>Here is the algorithm for DataoneDataStore.save(AbstractDataPackage abstractDataPackage)</p> <pre>DataoneDataStore.save(AbstractDataPackage abstractDataPackage) { dataPackage = buildDataPackage(abstractDataPackage);//This method will build the DataPackage object dataPackage.save();//save method will figure out to use create or update method. }</pre>			

```

DataoneDataStore.buildDataPackage(AbstractDataPackage abstractDataPackage)
{
    Identifier oreID = getIdentifier();
    DataPackage dataPackage = new DataPackage(oreID)

    if(abstractDataPackage.getDataPackage() != null)
    {
        dataPackage.setNewPackage(false);
    }

    //metadata part
    if(abstractDataPackage.newMetadata())
    {
        //new metadata
        d1Object = new D1Object(abstractDataPackage.getIdentifier(), metadataByte[], abstractDataPackage.getFormatId(),
            abstractDataPackage.getSubmitter(), getActiveNodeid());
        d1Object.setNewObject(true);
        //generate sysMetadata
        sysMetadata.setSubmitter(abstractDataPackage.getSubmitter());
        .....
        d1Object.setSystemMetadata(sysMetadata);
        dataPackage.add(d1Object);
    }
    else if(isDirty())
    {
        //updated metadata
        Identifier newID = getIdentifier();
        abstractDataPackage.setIdentifier(newID);
        d1Object = new D1Object=(abstractDataPackage.getIdentifier(), metadataByte[], abstractDataPackage.getFormatId(),
            abstractDataPackage.getSubmitter(), getActiveNodeid());
        d1Object.setNewObject(false);
        d1Object.setDirty(true);
        //generate sysMetadata
        sysMetadata = abstractDataPackage.getMetadataSysMetadata();
        sysMetadata.setSubmitter(abstractDataPackage.getSubmitter());
        .....
        d1Object.setSystemMetadata(sysMetadata);
        dataPackage.add(d1Object);
    }

    //data part
    for(Entity entity: abatractDataPackage.getEntityList())
    {
        if(entity.isNewEntity())
        {
            .....
            dataPackage.addData(entity);
        }
        else (entity.isDirty())
        {
            ....
            dataPackage.addData(entity);
        }

        // add relationship
        dataPackage.addRelationship(abstractDataPackage.getIdentifier(), entityIDlist);

        // add sysMetadata to the dataPackage
        oreSysMetadata= abractDataPackage.getDataPackage().getSysMetadata();
        oreSysMetadata.set.....
        dataPackage.setSysMetadata(oreSysMetadata);

    }
}

```

```
// new method in the Dataone DataPackage class
DataPackage.save()
{
for(Map.Entry d1Object: dataStroe.entrySet())
{
if(d1Object.isNewObject())
{
d1Object.create();
}
else(d1Object.isDirty())
{

d1Object.update();
}
}

//create or update the ORE document
String ore = serializePackage();
if(this.newPackage())
{
D1Client.create();
}
else
{
D1Client.update();
}
}
```

History

#1 - 2012-07-29 23:47 - Jing Tao

Currently, when user update a data package in morpho, the data package will be assigned a new id after user click the save button (serialize method is called). This is the last step to update a package.

Now we will have an ORE data package object as member attribute. This ore object will keep the track the ids. So it should represent the most current id information. I think when the package get "dirty" at the first time, a new id should be assigned and the ore object should be updated. We don't have to wait until the last step to assign an id to the data package and update the ore data package object. Note: after the first time the data package getting dirty, the following updating will NOT to result in assigning a new id.

When user create a new data package, morpo will assign a temporary id to this package first. The id will looks like temp.100.1. After click "Save" button, a formal id will be assigned. I think we can assign a formal id at the beginning. The reason is the same as the above one.

#2 - 2012-07-29 23:49 - Jing Tao

- File *abstract-datapackage-class-diagram.png* added

#3 - 2012-07-29 23:51 - Jing Tao

- File *morpho-save-datapackage-dataone-sequence.png* added
- File *morpho-save-datapackage-local-sequence.png* added
- File *morpho-delete-datapackage-dataone-sequence.png* added
- File *morpho-delete-datapackage-local-sequence.png* added

#4 - 2012-07-31 17:26 - Jing Tao

- File *morpho-load-datapackage-sequence.png* added

#5 - 2012-10-11 15:33 - Dave Vieglais

- Target version changed from *Sprint-2012.37-Block.5.3* to *Sprint-2012.41-Block.6.1*

#6 - 2012-10-24 18:20 - Ben Leinfelder
- Target version changed from Sprint-2012.41-Block.6.1 to Sprint-2012.44-Block.6.2

#7 - 2012-12-12 16:51 - Chris Jones
- Target version changed from Sprint-2012.44-Block.6.2 to Sprint-2012.50-Block.6.4

#8 - 2013-03-01 18:33 - Ben Leinfelder
- Target version changed from Sprint-2012.50-Block.6.4 to 2013.10-Block.2.1

#9 - 2013-03-02 05:28 - Ben Leinfelder
- Status changed from New to Closed
- translation missing: en.field_remaining_hours set to 0.0

Files			
abstract-datapackage-class-diagram.png	452 KB	2012-07-29	Jing Tao
morpho-save-datapackage-dataone-sequence.png	73.5 KB	2012-07-29	Jing Tao
morpho-save-datapackage-local-sequence.png	73.3 KB	2012-07-29	Jing Tao
morpho-delete-datapackage-dataone-sequence.png	42.6 KB	2012-07-29	Jing Tao
morpho-delete-datapackage-local-sequence.png	18.5 KB	2012-07-29	Jing Tao
morpho-load-datapackage-sequence.png	49.4 KB	2012-07-31	Jing Tao