

Metacat - Story #6548

Expand ORE model to allow relationships for derived datasets

05/06/2014 04:22 PM - Lauren Walker

Status:	Closed	Start date:	05/16/2014
Priority:	Normal	Due date:	
Assignee:	Lauren Walker	% Done:	100%
Category:	metacat	Estimated time:	0.00 hour
Target version:	2.5.0	Spent time:	0.00 hour
Description Design a new model for Metacat's OREs where relationships: - wasGeneratedBy - derivedFrom - used - etc. are used to describe datasets that are derived from raw data and metadata. These relationships may span OREs (e.g. an analyst may create data visualizations of another scientist's data and create a new package for those).			
Subtasks: Task # 6555: Convert PNG diagrams in ORE Model Expansion doc to plantUML diagrams Task # 6586: Index PROV relationships			
			Closed
			Resolved

History

#1 - 05/07/2014 10:42 AM - ben leinfelder

- Target version set to 2.5.0

Looking at the ORE spec, this is certainly acceptable practice - they just want to make sure that there isn't any orphaned node so that everything can trace back to either the aggregation or one of the aggregated resources within it. I believe we are also allowed to refer to objects that are aggregatedBy other resourceMaps, though I don't know if the derived resource map needs to also state that it "aggregates" the resource that it is deriving products from. Either way, I think this will be great.

Another thing to consider is doing ALL the semantic annotation assertions in the OREs. Not that it would be required, but it could be convenient since we already have a good precedent with folks starting to generate OREs for DataONE. My one concern is that the index parser would need to know how to handle the existing ORE packaging assertions as well as any SPARQL-based index processing we would want to do.

#2 - 05/16/2014 04:12 PM - Lauren Walker

- Status changed from New to In Progress

A page in the metacat docs has been added to describe changes to Metacat's ORE model.

#3 - 07/15/2014 03:27 PM - Lauren Walker

The model will need a second revision. Matt and I talked today about the "activities" in our model, which represent programs/scripts. There needs to be a place in the model for "runs."

A run would represent a single execution of a program. It would have properties like a start time, end time, parameters used, etc. Each run could possibly have unique parameters each time, especially if one or more functions creates a random number.

A program, (e.g. an R script), is not exactly an activity but another entity/data object.

If our model could store runs and separate the idea between programs and runs, data output can be reproducible since the run will have all the information needed to execute the program again using the exact same parameters.

#4 - 11/03/2015 11:11 AM - ben leinfelder

- Status changed from In Progress to Closed

This is being developed in GitHub: <https://github.com/DataONEorg/sem-prov-design/blob/master/docs/PROV-annotation-model/PROVmodel.md>