Kepler - Bug #1546

dynamic data and actor views using ontologies

04/30/2004 11:06 AM - Matt Jones

Status:	Resolved	**Start date:**	04/30/2004
Priority:	Immediate	**Due date:**	
Assignee:	Chad Berkley	**% Done:**	0%
Category:	interface	**Estimated time:**	0.00 hour
Target version:	1.0.0Alpha3	**Spent time:**	0.00 hour
Bugzilla-Id:	1546		

Description

Current lists of actors (and planned data sets) are static in that the tree is statically written into a MoML model and displayed. This severely limits the user's ability to find appropriate actors (and data sets) as the number of actors grows. The current tree is a combination of functional and project oriented folders, with no consistent classification.

This proposal is to generate dynamic views of the actors and data sets by organizing the actors into trees using simple ontologies and controlled vocabularies. Each actor (in its MoML code) and each data set (in its metadata description) would contain term references that are drawn from one or more ontologies. For example, an actor might be classified as belonging to the Class "SimulationModel" while another actor might belong to the Class "AnalyticalModel". If both AnalyticalModel and SimulationModel are subclasses of "Model", then we could display a dynamically generated tree like this:

```
Model
 |__ SimulationModel
 |__ AnalyticalModel
 |__ NumericalModel
 |__ IndividualBasedModel
```

with each of the Actors displayed at the appropriate node in the tree. Of course, if SimulationModel has subclasses itself, those could either be collapsed to show all models under SimulationModel, or additional levels of the tree can be added.

The same scenario applies to data sets, allowing people to browse data according to a particular classification ontology. For example, data could be classified as applying to certain types of measurements:
PhysicalMeasurement
ChemicalMeasurement
BiologicalMeasurement |__ MolecularMeasurement |__ CellularMeasurement |__ TissueMeasurement |__ OrganismMeasurement |__ PopulationMeasurement |__ CommunityMeasurement |__ EcosystemMeasurement

Although this example is somewhat contrived, it illustrates the type of ontology one might use. Need to talk to some domain scientists to determine an appropriate set of classifications for data.

Switching classification schemes would be done dynamically, on-the-fly. The set of ontologies that are available for display would need to somehow be limited to a meaningful set (all of the classes in even a small, simple ontology would overwhelm the user). This could probably be set through a configuration. In addition, the ontologies would need to be stored in a Kepler-accessible location, possibly included with the release.

History

#1 - 06/07/2004 11:09 AM - Chad Berkley

Another major issue with this is how to tag atomic actors that have no xml description. One option is to have some sort of LUT, possibly in a web service, for linking components to their ontological description. also need to figure out how to add this information to existing moml files without upsetting the moml parser.

#2 - 06/07/2004 11:10 AM - Chad Berkley

need to create screenshots and a possible prototype UI diagrams for the ontology browser. need to identify changes to the data and actor tabs for choosing views.

#3 - 06/08/2004 01:39 PM - Shawn Bowers

We need to clarify the "query language" that will be provided to allow a user to dynamically view actors and datasets via ontological information.

The simplest "query language" would be to allow a user to select a single concept to organize the view by. For example, by giving "AnalyticalModel", a tree would be created with "AnalyticalModel" as the root node, the various subtypes of "AnalyticalModel" as internal nodes, and the actual actors/workflows as leaves of the tree. (Note that we prune internal nodes here if no actors/workflows exist along the path.)

As more complex "query expressions" are permitted, how results should be displayed becomes less clear. For example, we could permit single concept query plus role restrictions such as "AnalyticalModel and uses StatisticalModel". In this case, it isn't clear how the root node should be displayed, or how further subtypes of the root are best displayed. We could permit multiple concepts in query expressions (implying that each concept is "anded"). Multiple concepts with role restrictions. And, arbitrary concept expressions (e.g., "AnalyticalModel using StatisticalModel or StatisticalRegressionModel"). Each of these would seem to complicate the display.

As a first cut, I think we should stick to single-concept queries.

#4 - 06/14/2004 04:14 PM - Shawn Bowers

This set of slides is a very early version of the sparrow api, including the dynamic view operation. Note that the version is as of yet incomplete and unimplemented. (I put it under this bug because at this point it is specific for 1546.)

#5 - 07/14/2004 11:39 AM - Shawn Bowers

Current design document for the sparrow java api (sparrow-api)

#6 - 10/12/2004 11:06 AM - Chad Berkley

This is now working via Jena. Shawn and I have changed the kepler configuration to only include a long moml list of actors and we have added an id property which uses lsids for unique identification. Shawns code then takes the list of actors and an ontology written in OWL and rebuilds the actor library based on the ontology. This should be working by the alpha3 release.

#7 - 03/27/2013 02:17 PM - Redmine Admin

Original Bugzilla ID was 1546

Files

sparrow-api-early.pdf	22.8 KB	06/14/2004	Shawn Bowers
sparrow-api-early.pdf	42.6 KB	07/14/2004	Shawn Bowers