Kepler - Bug #2234

creating KAR file loses port semantic annotations

10/19/2005 03:37 PM - Matt Jones

Status:	Resolved	**Start date:**	10/19/2005
Priority:	Immediate	**Due date:**	
Assignee:	Chad Berkley	**% Done:**	0%
Category:	core	**Estimated time:**	0.00 hour
Target version:	1.0.0alpha9	**Spent time:**	0.00 hour
Bugzilla-Id:	2234		

Description

I created an R actor on the canvas, annotated it and its ports with terms from the ontologies, and then saved the actor as a KAR file. When I drag the new actor from the tree to the canvas, the clone that is put on the canvas lacks its port annotations. Looking in .kepler/actorLibrary, the library file is missing the annotaitons for the port as well. Both the actorLibrary and the cloned copies should have all all of the original annotations.

This was tested against the Kepler CVS HEAD on 19 October 2005 ~ 2pm.

History

#1 - 11/02/2005 01:40 PM - Matt Jones

THis probably involves cloning more than just annotaitons -- need to verify that the clone in both directions puts all of the needed components into the MoML model (when creating KAR file, when registering in library, and when cloning from library to the canvas).

#2 - 12/14/2005 02:04 PM - Chad Berkley

this won't be fixed until alpha 9.

#3 - 01/20/2006 02:15 PM - Chad Berkley

One reason this is messed up is that the Semantic Type Editor (STE) creates properties with the naming scheme "semTypeX" where X is a number. The agreed upon name for this property was "semanticTypeX" where X is a number. The STE also seems to take control of any of the org.kepler.sms.SemanticType attributes and change their names, which breaks the kar ingestion system. The STE needs to be cured of its derelict activities and needs to learn to play nice with the kar system :)

#4 - 01/20/2006 02:44 PM - Shawn Bowers

I disagree. The responsibility should be on the kar ingestion system (kis) to take a broader view and detach itself from a fixation on naming conventions and move to the more definitive notion of the class (e.g., org.kepler.sms.SemanticType) ;-) Note that in MoML, one can edit the xml files by hand, and thus, we want to be careful about relying on property naming conventions, which is generally less robust (more brittle). Note also that one will never find a semantic type property without the associated class name.

#5 - 01/20/2006 02:47 PM - Chad Berkley

Another reason this doesn't work is that the KAR creation code seems to find all semanticType attributes and change the value of all of them to the same thing. seems the kar code needs to learn to play nice too.

#6 - 01/24/2006 10:36 AM - Chad Berkley

Semantic Annotations are no longer lost when creating kar files. This issue is fixed.

#7 - 03/27/2013 02:19 PM - Redmine Admin

Original Bugzilla ID was 2234