Kepler - Bug #2959

Need for Multiport input for RExpression actor

09/10/2007 02:32 PM - Dan Higgins

Status:	Resolved		**Start date:**	09/10/2007
Priority:	Normal		**Due date:**	
Assignee:	ben leinfelder		**% Done:**	0%
Category:	actors		**Estimated time:**	0.00 hour
Target version:	1.0.0		**Spent time:**	0.00 hour
Bugzilla-Id:	2959			

Description
Kevin Drury added a multiport input to an RExpression actor and noted that only the 1st channel was input. Currently, one can add a multiport input but it is not recognized as a multiport by the actor (NOTE: The Expression actor also allows one to add a multiport but does not recognize it or do anything with the multiple channels.)
Ideally, a multiport should probably just concatenate data from multiple attached channels.

History

#1 - 02/07/2008 04:14 PM - ben leinfelder

I propose (and have made the modification locally) that an RExpression actor with a multi port should:
1. handle input from all of its channels
2. create variables available in R that are named after the *upstream* source of the data for each channel.

In the case of multiports, the name of the input port of the RExpression actor will be ignored, and only the upstream name (the name of the output source port) will be used.
This allows an EML actor to feed as many "column as vector" structures into the RExpression actor and also preserves the names of the columns from the EML source (think about plot() and axis labeling and how cool that will be!).

#2 - 02/08/2008 10:09 AM - ben leinfelder

A note on variable naming conflicts:
-Take the case of 2 EML actors sitting upstream from an RExpression actor. Both EML actors are configured such that their output format is "as column based record". This sets their output ports to be named "Record" and requires that those names not be changed. When both EML actors feed into the same multiport on the R actor, the "Record" variable name is reused (which essentially overwrites the first "Record" value).
How to best resolve this case?
1. Append index numbers to the upstream variable name (Record1, Record2, ...RecordN)
2. Abandon the use of the upstream port name so that the RExpression actor is more autonomous (and portable) and as such does not rely on the configuration of other actors in any particular workflow. Any multiport input would use the name of the input multiport on the R actor with index numbers appended to the port name as variables available in R. (Multiport1, Multiport2, ...Multiport3).

By abandoning the upstream portnames, we loose the slight metadata advantage those names might give us, but we also ensure that RExpression actors can be reused in any workflow without the need to script modification.

#3 - 02/08/2008 11:11 AM - Matt Jones

The RExpresion actor should definitely not depend on upstream port names, because that makes the script non-portable -- every time you connect the R actor to a new actor in the workflow, you'd have to rename the variables in the R script. Way too brittle. Much better to use a serial numbering scheme for the channels as you suggested.

#4 - 02/09/2008 02:24 PM - ben leinfelder

The latest incarnation is to put any multiport input into an R list. The list variable name is the name of the input multiport on the R actor. Items in the list can be accessed directly using listName1 (for the first item in the list - this would be data from the first connection added to the R actor on that multiport).

In the case of records (aka dataframes) from EML actors, multiple EML sources can be connected to the same multiport, and a simple R script:
do.call("rbind", multiportName) will concatenate those dataframes into a big dataframe. This is equivalent to sql "union all"

Also, I've added support for multiport output so that the same data tokens can be sent to multiple downstream actors if the workflow design requires such.

#5 - 02/12/2008 03:13 PM - ben leinfelder

I've added a demo workflow that illustrates the use of multiports (both input and output) and am working with Kirsten to include the new features as part of the User Manual documentation.

Retargeting for 1.0.0 release

#6 - 02/21/2008 02:54 PM - ben leinfelder

I'm quite happy with how this is working and feel it is ready to be included in the 1.0.0 release - closing the bug. (especially since it now has Matt's float type support built on top of the multiport change)

#7 - 03/27/2013 02:21 PM - Redmine Admin

Original Bugzilla ID was 2959