

Kepler - Bug #4178

error handling with MoMLCommandLineApplication

06/22/2009 12:23 PM - Daniel Crawl

Status:	Resolved	Start date:	06/22/2009
Priority:	Normal	Due date:	
Assignee:	Christopher Brooks	% Done:	0%
Category:	general	Estimated time:	0.00 hour
Target version:	2.0.0	Spent time:	0.00 hour
Bugzilla-Id:	4178		
Description <p>It would be nice if MoMLCommandLineApplication called Manager.run() instead of execute() so that any execution listeners in the model would be notified of errors.</p> <p>Additionally, MoMLCommandLineApplication is an execution listener and uses executionError() to detect model errors (in its parent class), so it seems like a bug to call execute().</p>			

History

#1 - 07/10/2009 06:01 PM - Christopher Brooks

I've changed ptolemy.moml.MoMLCommandLineApplication so that it calls Manager.run() instead of Manager.execute().

Manager describes the differences between execute(), run() and startRun():

--start--
<p> There are three methods that can be used to start execution of a system attached to the manager. The execute() method is the most basic way to execute a model. The model will be executed <i>synchronously</i>, meaning that the execute() method will return when execution has completed. Any exceptions that occur will be thrown by the execute method to the calling thread, and will not be reported to any execution listeners. The run() method also initiates synchronous execution of a model, but additionally catches all exceptions and passes them to the notifyListenersOfException() method <i>without throwing them to the calling thread</i>. The startRun() method, unlike the previous two techniques, begins <i>asynchronous</i> execution of a model. This method starts a new thread for execution of the model and then returns immediately. Exceptions are reported using the notifyListenersOfException() method.
--end--

The lineage of this file is complex.
In the beginning, there was MoMLApplication, which was good.
However, MoMLApplication used the blessed Configuration system, so there was a need for a simpler solution, which was ptolemy.actor.gui.MoMLSimpleApplication, which was good, but maybe not so simple.

Initially MoMLSimpleApplication called manager.execute(), and then this change happened:

--start--

r43198 | hyzheng | 2006-07-10 15:17:16 -0700 (Mon, 10 Jul 2006) | 2 lines

call manager.run instead of manager.execute
--end--

Then, manager.run() was replaced:
--start--
r43200 | hyzheng | 2006-07-10 19:07:29 -0700 (Mon, 10 Jul 2006) | 4 lines

to make tcl tests that use "createAndExecute" pass, we have to put the exception handling into the execute() method because the run() method does not throw normal exception but some runtime exception.

The tcl tests include sdf/test/SDFSchedulerErrors.tcl and de/test/DependencyLoop.tcl.
--end--

Then, MoMLSimpleApplication was moved from actor.gui to moml to avoid some dependency issues.

Then, this change was made:

--start--

r53382 | cxh | 2009-05-01 13:01:37 -0700 (Fri, 01 May 2009) | 2 lines

Use Manager.startRun() so that we more safely catch errors in a multithreaded context.

--end--

After awhile, it was seen that we needed a class that would handle command line parameters, so MoMLCommandLineApplication was born. And it was good (again).

MoMLCommandLineApplication was calling execute(), but now it calls run(). This will avoid the problems that Haiyang had in 43198 and 43200 because MoMLCommandLineApplication is derived from MoMLSimpleApplication, which is used by the test system.

Note that MoMLSimpleApplication implements ChangeListener and ExecutionListener.

The reason for this is:

--start--

r35781 | cxh | 2005-01-04 17:03:08 -0800 (Tue, 04 Jan 2005) | 3 lines

implement ExecutionListener so that if a PN model throws an exception we can notice it and report it.

--end--

Looking at the ptolomy mail from around then, I found my email:

--start--

I did
cd \$PTII/ptolomy/domains/test; make
and split up the results below with lines of ### chars

When running in auto/knownFailed tests I get:

#####

```
----- testing $PTII/ptolomy/domains/test/auto/knownFailedTests/PNSRTimedtest.xml (Known Failure)
java.lang.RuntimeException: Execution error .PNSRTimedtest.Clock2
at ptolomy.actor.gui.MoMLSimpleApplication.executionError(MoMLSimpleApplication.java:189)
at ptolomy.actor.Manager.notifyListenersOfThrowable(Manager.java:620)
at ptolomy.actor.process.ProcessThread.run(ProcessThread.java:222)
Caused by: ptolomy.kernel.util.InternalErrorException: Subtracting a positive infinity from a positive infinity yields a NaN.
at ptolomy.actor.util.Time.subtract(Time.java:440)
at ptolomy.actor.util.TimedEvent$TimeComparator.getVirtualBinNumber(TimedEvent.java:133)
at ptolomy.actor.util.CalendarQueue._localInit(CalendarQueue.java:556)
at ptolomy.actor.util.CalendarQueue.put(CalendarQueue.java:246)
at ptolomy.domains.pn.kernel.TimedPNDirector.fireAt(TimedPNDirector.java:193)
at ptolomy.actor.lib.TimedSource.initialize(TimedSource.java:172)
at ptolomy.actor.lib.Clock.initialize(Clock.java:343)
at ptolomy.actor.process.ProcessThread.run(ProcessThread.java:131)
Warning: '.PNSRTimedtest.NonStrictTest' The test produced only 0 tokens, yet the correctValues parameter was expecting 1 tokens.
168 ms. Memory: 3520K Free: 1699K (48%)
```

I modified MoMLSimpleApplication so that it implement ExecutionListener so that we now properly handle exceptions thrown in threaded domains.

---end--

I reverified that indeed we want to have MoMLSimpleApplication implement ExecutionListener, otherwise we get a warning message:
"No executionListeners? Error message was:" from Manager.notifyListenersOfException().

#2 - 03/27/2013 02:25 PM - Redmine Admin

Original Bugzilla ID was 4178