

Kepler - Bug #2246

TOOLBAR: status on opening workflows and blank frames

11/04/2005 10:58 PM - Matthew Brooke

Status:	In Progress	Start date:	11/04/2005
Priority:	Normal	Due date:	
Assignee:	Christopher Brooks	% Done:	0%
Category:	interface	Estimated time:	0.00 hour
Target version:	3.X.Y	Spent time:	0.00 hour
Bugzilla-Id:	2246		
Description			
-- should be easy - replace existing gif images with Laura's new ones, and add code to incorporate new icons for rollover/pressed states.			

History

#1 - 11/04/2005 11:06 PM - Matthew Brooke

also need to go thru' all new toolbar icons and check colors are to spec; graphic designer got mis-matches on some of them

#2 - 11/17/2005 12:34 PM - Matthew Brooke

There are four icons for each toolbar button. Per Laura, these represent the four states:

- Normal (unpressed)
- Rollover of normal unpressed state
- Pressed state
- Rollover of pressed state

#3 - 12/13/2005 11:43 PM - Christopher Brooks

Hi Matthew,
If you want, I can take this bug over and work on getting the icons updated in the Ptolemy tree.
I'd need a pointer to the new icons.

_Christopher

#4 - 12/14/2005 09:11 AM - Matthew Brooke

Christopher -

do you **ever** sleep??? ;o) That would be wonderful. The icons are in the relatively-newly-created "kepler-docs" cvs repository, at:

kepler-docs/dev/usability/graphics/24x24_toolbar_gifs

See previous comment for key to which images are which. FYI, I looked at the code briefly a while ago, and it looks like we'll have to add an extra rollover state that isn't currently used in ptii/kepler - I think it was the "Rollover of pressed state", but i could be wrong. Anyway, at the time, it seemed relatively simple - just using an attribute that was already implemented in javax.swing.Action. Now I've written this, I'm not sure any of the toolbar buttons would even **use** a "Rollover of pressed state", but i'm sure i can be proven wrong :-)

I didn't want to be so presumptuous as to reassigning the bug to you, but please go ahead and do so if you want to - if not, I'm fine with it staying as is.

Thank you!

#5 - 12/14/2005 03:10 PM - Matthew Brooke

Actually, it was a method in AbstractButton, not an attribute in Action, although I'm still not 100% clear on which (setRolloverSelectedIcon(), maybe??).

#6 - 12/16/2005 02:07 PM - Christopher Brooks

I have the new icons for the zoom*, run, pause and stop buttons working. I'll check it in once I'm sure alpha8 is tagged.
The changes involve diva/gui/GUIUtilities.java, so diva.jar will need to be remade. The other changes are in ptolmy/vergil. I renamed the images slightly to match our current images.

Open issues:

1) The background of the new buttons is different than the default current background. Setting the background in Java is fairly difficult (broken). I can take a look at this, but it could be tricky. Are we sure we want a new background color? The problems we ran in to with getting the background right in Ptpplot were fairly serious. I'm not a great UI programmer, so someone else might have a better solution for this.

2) The port and relation icons on the right side are not gifs, they are Diva figures. Revamping them to gifs is not for the faint of heart. The icons are generated in vergil/actor/ExternalIOPortController.java. It might be better to focus on other tasks before returning to these images.

#7 - 12/16/2005 02:51 PM - Matthew Brooke

Christopher -

Are we sure we want a new background color?

for some background info (no pun intended ;o), see my comments and Laura's responses in bug [#2271](#)

Maybe a sensible first shot would be to redo the GIFs with transparent backgrounds, so they will show whatever background color is currently configured in the Java code. That way, they will work with the current UI, and also with any future changes.

This would be quite simple to do if we had the original Photoshop or Illustrator files, which I believe Laura might be able to get from the graphics guy who created the icons? We could try just editing the gifs, but this might lead to ragged outlines because of anti-aliasing.

#8 - 12/16/2005 04:42 PM - Christopher Brooks

Hi Matthew,

Thanks for the intel on the background color. Since there is a separate bug open, I'm not going to address that here. I'll append to that bug later.

As a short term hack, I made the slightly darker blue transparent with:
`mogrify -transparent "#BEC8D3" `ls -1 *.gif | grep -v _``
This is not perfect, it would be best to regenerate these gifs.
Once alpha8 is tagged, I'll check in my work.

I looked into port icons and we could do one of two things:

1) Modify the diva layout to match the images. The advantage of this is that the buttons and the icons that end up on the canvas would be exactly the same. This is a total win, less duplicate code etc. However, I doubt if we can easily get rollovers to work.

2) Modify how we handle creating the icons so that we use the gifs on the toolbar. The advantage of this is that we would easily get rollovers to work etc. The disadvantage is that each icon is now in two formats (diva and gif).

[#1](#) is easier, but [#2](#) would probably look better. I'll chew on this.

#9 - 12/16/2005 08:40 PM - Matthew Brooke

In kepler-docs/dev/usability/graphics/svg, we already have new svg figures for the port and relation icons (see in*.svg etc), which correspond to the toolbar icons in kepler-docs/dev/usability/graphics/24x24_toolbar_gifs, so maybe [#2](#) wouldn't be so bad after all? It seems like the more intuitive way to go, coming at this from a UI programming/code maintenance point-of-view - as a mere mortal, I'd rather knock out a couple of gifs than deal with diva rendering on the toolbar ;-)

Extra info: the new actor icon scheme for Kepler will use 2 formats for the actor icons - an svg for the workflow icons and a gif for the actors in the

library tree; you can see these in the aforementioned directories, too. Since our system is based around having double copies of icons in different formats, does this further allay your fears about #2? (or does it just convince you completely that we're doomed to failure?? ;-)

#10 - 12/18/2005 04:02 PM - Christopher Brooks

I've checked my changes in to the ptll cvs tree.
Note that diva.jar was updated.

I changed the blue background to transparent so that the UI does not look funny pending the resolution of bug #2271
(http://bugzilla.ecoinformatics.org/show_bug.cgi?id=2271)

I'm now working on getting the ports on the canvas to match the gif toolbuttons. One issue is that the new gif images are 2 pixels longer than the old ports. I'm not sure if this will mess up the layout or not.

#11 - 12/18/2005 10:10 PM - Christopher Brooks

I modified how the ports are rendered so that they match the the port gifs. I also modified how the start, pause and stop buttons work so that when we open a model, the stop button is selected. If we hit the start button, the stop button is deselected and the start button is selected until the model ends or we get an error. The pause button works in a similar fashion.

This bug is ready to be closed, pending feedback from users about the buttons.

#12 - 12/19/2005 09:39 AM - Laura Downey

Okay sorry but I'm confused here. When a workflow is open, no action has been taken on the workflow, therefore actions have no state (e.g., run, stop, pause) and should have no selection or highlight until user activated. The exception to this might be in formatting (e.g., the left justification icon is highlighted upon opening Word). But run, pause and stop are not default formats.

#13 - 12/19/2005 11:06 AM - Christopher Brooks

I reopened this one because I had mistakenly closed it.

When the graph viewer opens, the initial state of the model is that it is not running (stopped). My latest hacks have the stop button selected to indicate this. It is possible to open up some of the model viewers, such as the run control panel, and have the model start running immediately. In fact, I think Edward's work will bring up the graph viewer and run a model automatically if necessary.

I see Run/Pause/Stop as a sort of stop light and that the model should always be in one of those three states. When one drives up to an intersection, the lights are in one of those three states (unless the power is out).

My changes from yesterday were somewhat experimental. I like the idea of using Run/Pause/Stop as a way of indicating the state of the model. We could go back to having these be push buttons instead of toggle buttons. What got me started was that we now have button icons for various states, like rolling over a selected button, which formerly would not make much sense since it was impossible to do (I think). So, I tried to make use of the extra button states and give the user more info.

There is also the interesting case where a model is running and the user opens a composite actor. I think the buttons should reflect the state of the model (Running).

Anyway, I'm open to suggestions on this one. My button state changes were experimental.

Another potential problem is that the ports are now a little larger than they were. This could cause problems elsewhere.

#14 - 12/19/2005 11:28 AM - Laura Downey

My interpretation is that the model is in no state upon opening, it has not been run, stopped or paused yet -- it is available for the user to initiate those actions. And a model can't be stopped, until it has been run so if a

user hasn't run it, how can it be stopped upon opening? Now if there was a way to open a model and specify it immediately starts to run, then indicating this by the run toolbar icon being "depressed" would be a correct indication. But to me that would still be the user specifying that the model open and run so they are in essence initiating that action.

I don't understand your comment about utilizing the extra rollovers? The rollover state of a selected versus unselected button is what was added, as this is standard and is nothing more than a visual indication as to where the cursor is positioned on the toolbar. It is just a subtle visual indication that the cursor is positioned over something that is selected or "on."

#15 - 12/19/2005 11:39 AM - Matthew Brooke

I hadn't really thought about this before (apart from being a little confused as to why we had button icons for the "rollover pressed" state).

I think Christopher's suggestion makes a lot of sense - toggle buttons would give the users more info about what state the workflow is in, and I can't think of any reason why this might potentially confuse users.

If a model is not running when open, then it makes sense to me that it is in the "stopped" state, rather than "stateless" (without getting too philosophical ;-). In software and consumer electronics devices, I've seen "Stop" mean 2 things:

- 1) "not running, and when play is pressed, will carry on where I left off", and
- 2) "not running, and when play is pressed, will start back at the beginning"

Since (1) is essentially the same functionality as "pause", I think users will consider "stop" in Kepler as being (2), so it seems appropriate to have the stop button highlighted on startup.

#16 - 12/19/2005 11:49 AM - Laura Downey

I suspect some people would view things as no state upon opening, and obviously others would view the workflow as in the stop state. I tend not to design in a cue to users in terms of action unless they have specified and initiated that action. When a user does something, they immediately get the feedback of their action and running, pausing and stopping workflows are major actions initiated by the user.

That said, I doubt opening a workflow with the stop button depressed is going to create any kind of major confusion. Those people that think "hey why is this stopped when I didn't do anything?" will quickly learn that is just the default upon opening a workflow.

#17 - 02/02/2006 01:00 PM - Matthew Brooke

closed - fixed by Christopher

#18 - 02/13/2006 01:42 PM - Laura Downey

Re-opening this bug for a couple of reasons.

1) toolbar icons don't display correctly for the first four icons in the toolbar -- no rollovers appear, no selections appear and I suspect no rollovers of selected icons appear either.

2) it seems very unusual to have the stop button selected when no workflow is even loaded. Why are we communicating a stop state of something that isn't there?

#19 - 02/14/2006 05:08 PM - Christopher Brooks

The rollovers for the zoom buttons work in Vergil, but not in Kepler because vergil/basic/BasicGraphFrame is duplicated. I might have time to update the duplicate copy of BasicGraphFrame tomorrow, if not, then someone else will need to do this. The fix is to look at the Zoom*Action inner classes.

About the stop button, we covered it in the the bug and in email.

My position is that the buttons should reflect the state of the model. A blank model is not running, so the stop button should be depressed. It should be possible to open a model and have it automatically start, so that the user will see it running and the run button is depressed.

The run control panel can do this, where when an applet is opened the model is already running.

If someone else wants to take a shot at modifying the pttl code so that the stop button is not depressed, then that would be fine with me. I won't have time to get back to this any time soon.

#20 - 02/15/2006 08:59 AM - Matthew Brooke

To help with tracking for the release, I am opening a separate bug (see Bug [#2355](#)) for the "icons don't display correctly for first four icons in the toolbar" issue, and am leaving this bug to deal with the issues regarding the toolbar status when a workflow is first opened. Changing the name of this bug to reflect this.

#21 - 02/15/2006 09:05 AM - Matthew Brooke

This proposal would probably solve the issue with empty workflows: I think "inappropriate" actions should be grayed out when they don't apply (for example, probably all the toolbar buttons should be grayed out when there is no workflow, as should many of the menu items - "save" etc. They should then be enabled when a workflow is opened).

PTII/old kepler doesn't do this - so we currently have a situation whereby the user can press the "run" button with an empty workflow, and then gets a crazy cryptic error message.

Any thoughts on this? If everyone agrees, we can enter a bug to enable/disable menu and toolbar actions appropriately, depending on the context. Please note, however, that this will require a significant amount of work to implement properly, so we may still need to come up with an interim solution.

#22 - 02/16/2006 11:09 AM - Laura Downey

added bug 2359 to address enabling/disabling of menu/toolbar items as appropriate

#23 - 04/13/2006 02:24 PM - Christopher Brooks

I took another look at this bug. In many ways, we could probably close this, it falls under the newer bug "enable/disable menu items and toolbar items as appropriate" http://bugzilla.ecoinformatics.org/show_bug.cgi?id=2359 I think this bug started out as one using the new button images and has slowly transmogrified. If someone wants to close this, I'm all for it. I've lowered the priority slightly on this one, since it does not prevent someone from working or lose data. I'm tempted to make it an enhancement, but it is a borderline case of a bug.

The bug here is that when we open a blank frame, Laura would like none of the run/start/stop buttons to be enabled until the user starts creating a model.

As a short term hack, I adjusted the error message so that if the user does hit Run on an empty model, the exception says: "Cannot preinitialize an empty model, please add actors and a director."

To really fix this bug, we should probably add another state, called RUNNABLE, which means that the model contains one actor. (Note that if there is no director in a model, then the default Director is chosen, which fires the actors in the order they were placed in the model.) Then, when the user drags an actor in, we could set the manager state to RUNNABLE and the UI Run button would be enabled.

Another hack would be to try and modify RunnableGraphController.managerStateChanged(), and put the logic of checking if the model has an actors there. However, that code is only called when the manager state changes, so it would never see that the model has actors so the Run button would never be enabled.

I do think that the current functionality we have for opening a preexisting model is correct. The model is in the stopped state, so the stop button should be highlighted. Some model viewers, such as the MoMLApplet can auto start models, and MoMLApplication can take a -run argument which automatically runs the models. Thus, having the stop button highlighted makes sense here if the model is runnable.

#24 - 03/27/2013 02:19 PM - Redmine Admin

Original Bugzilla ID was 2246