Kepler - Bug #1750

Allow KAR files to include jar/class files

10/27/2004 06:42 AM - Bertram Ludaescher

Status:	Resolved	**Start date:**	10/27/2004
Priority:	Immediate	**Due date:**	
Assignee:	Aaron Aaron	**% Done:**	0%
Category:	core	**Estimated time:**	0.00 hour
Target version:	2.0.0	**Spent time:**	0.00 hour
Bugzilla-Id:	1750		

Description

The issue of a remote Kepler actor (and workflow) repository has come up every now and then. It would serve multiple purposes:
- as an actor and service registry for discovery, similar to a UDDI registry for web services (only better ;-)
- to avoid loading each and every actor when installing Kepler: actors could be discovered dynamically (with an expanded actor search function), their interfaces could be loaded into the local users view (so actor signatures become visible); eventually for "local actors" their code would be loaded from the remote repository and plugged into the user's local installation.
Note that some actors will require jars from the repository to be downloaded and plugged-in, while others, most notably WS actors only required to download and plug-in the interface or a thin "interpreter layer"
- as a community repository for common tasks (semantic annotations for actor and workflow classification become very important)

A related issue is that of a browsable and searchable data repository.

Bertram

Related issues:

Blocks Kepler - Bug #4317: KAR's dependencies on modules need to be included ...	**Resolved**	**08/14/2009**
Blocks Kepler - Bug #4340: Resolve KAR file/module format	**Resolved**	**08/26/2009**

History

#1 - 01/20/2005 10:26 AM - Chad Berkley

this is related to the kar files and dynamically searching for objects in the kar files. we already have a service registry and storage mechanism, but we need a standard description for how to store items in each registry and what metadata is needed per registry.

#2 - 01/20/2005 10:42 AM - Chad Berkley

dynamic class loading issues also need to be worked out for this system. dan will assist chad on this.

#3 - 01/20/2005 11:13 AM - Matt Jones

Need to define the archive file format, including manifest structure and contents, and define its naming convention. We like .kar but it is used already. Also available are .ksw and .kjar.

#4 - 01/21/2005 08:12 AM - Christopher Brooks

Below is a little info about dynamic class reloading in Vergil from Steve Neuendorffer. Reloading a class is harder than loading it.

At 04:45 PM 9/9/2000 -0700, Steve Neuendorffer wrote:

> I've been looking at making Vergil capable of dynamic class reloading. With a little assistance from some discussions with Jorn, I figured out what I done incorrectly before. The basic principle is that you can reload classes If and only if:
> 1) The classes are NOT accessible to the bootstrap class loader i.e. they

are not listed in the JVM's classpath.
2) The interfaces that are used to refer to those classes are not
reloaded.

For example: we can reload actor.lib.Ramp by removing it from it current
location and putting it somewhere else. This works because all the
important thing about it (such as the actor and port interfaces) are not
reloaded. So I can say:
Classloader c = new URLClassLoader(SOMEURL, getClass().getClassLoader())
and then use that classloader to reflect the ramp actor.
Later I could throw away that class loader and create a new one and the
classes would get reloaded. Notice that the old classes (and instances)
are not removed from the JVM... some other mechanism is resplosible for
keeping track and reinstantiating everything.
I had added support in the MoMLParser for specifying a class loader
before. All I do now is just create a new classloader everytime the
classes in the "dynamicClassPath" need to be reloaded. I haven't checked
these changes in yet, but I will soon.

Steve

#5 - 11/02/2005 12:21 PM - Matt Jones

We have now decided to only include initial actor loading in this feature.
Reloading, and version and class name conflicts, will be dealt with after the
1.0.0 release. This simplified feature would allow just discovery, download,
and loading of remote actors and workflows.

#6 - 11/16/2005 09:37 AM - Chad Berkley

we decided to wait on this until after the next release.

#7 - 11/29/2007 01:35 PM - Dan Higgins

The Kepler code has been modified to allow for jar files inside of kar containers. At startup, when kars are loaded into the cache, the jar files are copied into the $KEPLER/lib/jars/ directory and dynamically loaded into the classpath. (see http://www.kepler-project.org/Wiki.jsp?page=AddingJarsToKarFiles)

Currently, a problem with loading kar files containing jars from the repository has been reported (as opposed to letting Kepler load them at startup). This bug needs to be investigated and perhaps entered as a new bugzilla bug.

Dan Higgins - November 29, 2007

#8 - 11/29/2007 03:01 PM - Dan Higgins

Concerning comment #7 --- see new bug # 3020

#9 - 04/13/2009 10:26 AM - Chad Berkley

A possible addition to this system would be to have the JVM automatically find jars in a kar file and add them to the classpath when the kar file was loaded.

#10 - 05/19/2009 08:52 PM - Aaron Aaron

KAR files will not contain class files or JAR files. Classes and jars will be contained within Kepler modules. KAR files will contain Kepler module dependency information.

#11 - 05/19/2009 11:22 PM - Matt Jones

KAR files used to contain class files, and wer eboth a successful and necessary part of Kepler 1.0. I am reopening this bug to restore this feature in 2.0 if it has been removed without broader discussion. This decision should be part of the discussion regarding the merging of the KAR and module packaging formats -- I think we should have a single packaging mechanism for archives, eliminating the distinctions and differences between KAR files and modules. Bug is reopened pending further discussion.

#12 - 08/26/2009 01:01 PM - Chad Berkley

This issue needs clarification as to why the module format and the kar format need to be separate. It seems that both are performing the same functionality and that by limiting the kar format to not include jars or classes we are limiting future functionality of kar files and limiting them to a superficial role. Fixing this bug would make bug 4317 obsolete.

#13 - 08/27/2009 08:58 AM - Bertram Ludaescher

Seems this is a long standing issue that requires further discussion.

Any pointers to design documents or similar?

Bertram

#14 - 08/27/2009 03:52 PM - Aaron Aaron

Currently evaluating the necessity of this based on further documentation and group discussion.

#15 - 01/13/2010 03:31 PM - Chad Berkley

This can be talked about further after the 2.0 release.

#16 - 01/14/2010 08:59 AM - ben leinfelder

did i miss something yesterday? - we are dropping a feature that was in 1.0.

(In reply to comment #15)

> This can be talked about further after the 2.0 release.

#17 - 01/14/2010 11:04 AM - Chad Berkley

Umm, you're right. I was triaging bugs and made a couple mistakes. I'll put this back to 2.0.

#18 - 01/14/2010 01:42 PM - David Welker

This is a controversial design decision and should not be in 2.0.

Including jars in KAR files is NOT a feature. It is an implementation detail.

The feature or requirement is this: to allow individuals to distribute necessary jar files with the actors they develop.

In my view, this requirement should be implemented through modules.

More generally, we should be careful to distinguish between requirements and implementation details. Having jars in KAR files is NOT a requirement. It is a mere implementation detail that has been proposed to achieve a particular objective.

#19 - 01/27/2010 05:04 PM - Aaron Aaron

It was agreed that the solution to this is to create a java utility for converting KAR files to Kepler modules. See bug 4702

#20 - 03/27/2013 02:18 PM - Redmine Admin

Original Bugzilla ID was 1750