OSGi Support

Over the years and after multiple investigations of OSGi, the OpenMRS Developer Community has decided not to invest further in adopting OSGi within OpenMRS.

Primary mentor	Ben Wolfe
Backup mentor	TBD
Assigned to	Filip Spiridonov

Abstract

Ben built a modular framework for us in OpenMRS when OSGi was still fairly early in its design. While the module framework within OpenMRS has been one of the platform's greatest assets, allowing for more rapid uptake and development across implementations, there are benefits to migrating towards OSGi, since it represents a increasingly adopted standard for modular frameworks.

The OpenMRS ticket for this work is TRUNK-1596.

Objectives

- Create an enterprise war that is smaller and easier to distribute. (jars provided as osgi packages to the osgi container)

Extra Credit

- Replace the openmrs module infrastructure with an osgi loading/unloading of services

Design Ideas

The ideas for migrating toward OSGi are still very raw and largely undocumented. We have spoken with architects of the Eclipse OSGi framework, but are still in an investigatory stage.

One example to consider is Atlassian. They have migrated their plugins to OSGi and even supply an OSGi wrapper for backwards compatibility. They also have some useful documentation on their architecture.

Example exploratory steps to take:

1. Turn the openmrs war into a WAB and install into an OSGi server like Felix or Equinox (and of course get tomcat or some other servlet container running in that same osgi container)
2. (optional) Remove unnecessary jars from the war/wab file and make those dependencies of the osgi container instead
3. Turn an omod into an osgi jar and install into the container
   - Potentially the biggest step in the process
   - Do we have to call module services and openmrs services in a new way?
   - Can we pull libraries/dependencies out of the omod?
4. Add wrapper ability (in the webapp) that allows admin to upload an omod and it puts in the osgi strings and drops the now-osgi jar into the container
5. (optional) create a script that adds necessary osgi components to an existing omod source folder

Resources

- OSGi Alliance