Module Architecture Development

Background

At a face-to-face meeting in April 2009, the Fedora committers agreed to begin moving toward OSGi as a module framework for Fedora. Over the remainder of 2009, several of us gained experience with OSGi:

- Eddie worked on a Fedora-on-OSGi project, taking first steps to build the entire Fedora webapp as an OSGi bundle.
- Eddie experimented with building Mulgara as an OSGi bundle.
- Chris changed all Akuba modules to be built as OSGi bundles.
- Bill and Andrew are actively using OSGi in DuraCloud, to aid in the deployment of arbitrary services.

When the committers met in London in Feb 2010, we discussed what we have learned about OSGi and came to the conclusion that, while OSGi provides several capabilities that we still see as valuable (dynamic re-configuration, classloader isolation, and a standard module lifecycle), it has a significant learning curve and brings with it a new set of dependency management issues. We also noted that moving to a modern dependency injection framework would not be as difficult, nor necessarily incompatible with an eventual move to OSGi.

So, rather than concentrating on "Moving to OSGi" as a goal in itself, we resolved to improve Fedora’s modularity in other, more tractable ways, while using the knowledge we’ve gained over the last year to increase Fedora’s "OSGi-Friendliness" over time.

Goals

We have identified a few concrete steps we can take to move Fedora’s Module Architecture forward:

- Document, maintain, and apply best practices for being “OSGi Friendly”
- Change Fedora’s Modules to use a modern dependency injection framework
- Retain the long-term goal of having a Fedora OSGi bundle that can be used by other apps

Getting Involved

Want to help? Great! Please signal your interest by adding your name below, contributing to this wiki page, and participating in related discussions on the fedora-commons-developers mailing list.

Lead: ☺ Chris Wilper

Interested Contributors: 👍 Dan Davis, 👍 Andrew Woods, 👍 Asger Askov Blekinge