CSSE 477 - Homework 1 – OSGI Architecture

Introduction

Plugins have become one of the key mechanisms to develop extensible framework in almost all large-scale systems. We have recently studied about the architecture of different frameworks including Eclipse and OSGi that support plugins. We did a tutorial to figure out how a plugin can be implemented and expose services for other plugins to use in OSGI. Now it is time to get your hands dirty by developing a **pluggable framework** using OSGI.

Specification

I am not putting a lot of restrictions in the specification to encourage creativity. I am just listing down the base requirements in this section but there are a lot of creative considerations that you can make for this assignment.

User Interface

Your framework must be composed of at least two parts: **Plugin** and **Platform**. Platform among other things must have a GUI and must accept extensions in the form of OGSI bundles. Extension bundles (some but not all) should also be able to accept extensions from other bundles resulting in a pure pluggable architecture.

Base Requirement

1. The platform GUI must have three panels, say: **Listing Panel**, **Execution Panel**, and **Status Panel**.
2. The *Listing Panel* must somehow show all of the plugins installed in the application. User must be able to choose a plugin to execute.
3. When user chooses a plugin on the *Listing Panel*, the GUI content of the plugin must be displayed on the *Execution Panel*. Because of this requirement, all plugins must have GUI contents.
4. Both platform and plugins must be able to post status messages on the *Status Panel*.
5. You must show **an example** in which an execution panel plugin allows another plugin to contribute to its GUI.

Creative Considerations (Open-Ended)

- Life-cycle of a plugin (mostly handled by OSGI, but do you need any additional life cycle states other than the one provided by OSGI)
- What should happen when user switches between plugins
- How to install a plugin that is dropped in a special plugin install folder at runtime

Submission and Grading

You will need to get your work checked-off by your instructor. Demo your product to him either on Tuesday during class or Wednesday in his office. In any case, you must to bundle your projects in **one zip file** and turn it in on Moodle before the due date.

Have fun! ☺