A methodology of state mapping

based on OSGi for runtime evolution

1. Introduction
2. Background

Recently the maintenance of software after it has been shipped is large cost. Component based development has been made to address this issue. Modular design is one of the most benefits of component based application therefore allows application to evolve with relative ease. The OSGi specifications define a standardized component-oriented, computing environment for networked services that is foundation of enhanced service-oriented architecture. The OSGi service platform is driven by the dynamic nature of the OSGi framework. Installing a new bundle, registering a new service, or updating an existing bundle does not require a restart of the JVM.

1. Issue

However, although the difficult problem about replacement of a component at runtime without stopping application has been solved, but the state contained in the active version of the component must somehow be transferred to its successor. Since lack of state management in OSGi specification, programmers have to manually implement the state mapping functionality. It may limit the efficiency of system at runtime.

1. Approach

In order to automatically transfer the state of the older version to new version, we based on different stages in state mapping process. Develop a methodology to deal with state mapping based on Equniox which is a implementation of OSGi specification. According to two aspects: design time aspects and runtime aspects.

Design time aspects describe how to develop a new version of bundle. Run time aspects describe the process of replacement of bundle.

And develop a framework to do the state mapping process.

1. Contribution

This paper proposes a methodology which more efficient to support bundle replacement process, and a framework which we design and develop to support state mapping management during the bundle’s life cycle. The framework was designed as far as possible little human intervention for state mapping.