A methodology of state mapping based on OSGi for runtime evolution  
1. Introduction  
1. Background  
  Recently the maintenance of software after it has been shipped is large cost. Component-based development have 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 the foundation of an 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.  
  
2. Issue  
  However, Although the difficult problem about replacement of a component at runtime without stopping the application  have been solved, but the state contained in the active version of the component must somehow be transferred to its  successor.  
  Since lack of support about state management in OSGi specification. Programmer has to manually implement the state  transition functionality. It may limits the efficiency of system at runtime.  
  
3. Approach  
  In order to automatically transformer the states of the older version to new version, we based on different stages in the  state mapping process, develop a methodology to deal with state mapping based on OSGi. According to two aspects:  Design Time Aspects and Runtime Aspects.  
Design Time Aspects:  
1. Design and develop a state transformation framework which provides state mapping function for the bundle replacement  
2. Modify the bundle which imported the package of the bundle prepare for state mapping.  
3. Develop the new version of bundle.  
4. Configure the description of the correspond relations between the old version bundle and new version. (XML format meta information configuration file or java annotation)  
Runtime Aspects:  
1. Stop old version of bundle  
2. Uninstall old version of bundle  
3. Install new version of bundle  
4. Start new version of bundle  
5. Refresh the bundle which imported the package of the bundle prepared for state mapping.  
  
4. Contribution  
  The main contribution of this paper are:   
Design a framework which support state transformation management according to the extension bundle of OSGi specification which provides the state mapping functionality.   
Implementation of the framework as far as possible little human intervention for state mapping.    
Empirical validation of the state mapping process.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
2. Runtime state mapping  
  2.1 Overview  
Generally software maintenance process consists of stopping the system to be updated,  performing the update of the code, and restarting the system.In some situation like air-traffic controllers and life-support software system, must run continuously  and interruption of the service is not an option in general[1]. The OSGi component model is a dynamic model. Bundles1 can be installed, started, stopped, updated, and uninstalled without bringing down the whole system.   
   
Programmer has to manually implement the state transition functionality. It may limits the efficiency of system at runtime.  
OSGi reduces complexity and thus development and maintenance expenses by providing a modular architecture for today's large-scale distributed systems as well as small, embedded applications. The OSGi programming model realizes the promise of component-based systems[3].