

# 基于OSGi的知识服务应用示范系统 体系架构设计与实践\*

□ 鲜国建 赵瑞雪 寇远涛 朱亮 / 中国农业科学院信息研究所 北京 100081

**摘要:** 文章在简要介绍开放服务网关 (OSGi) 规范、Bundle模型及实现框架基础上, 设计了基于OSGi的知识服务应用示范系统体系架构, 共包括资源层、数据交互层、功能组件层、服务组件层、组件管理层和应用服务层等六层, 最后基于Felix框架实现了一个简单的知识服务原型系统, 初步完成对科技监测、领域知识结构及其演化分析、学术关系网络分析、领域科研信息环境和关联数据服务等知识服务子系统的模块化、组件式开发, 验证了各服务功能实现即插即用、动态化管理的可行性和先进性, 初步为各知识服务示范子系统的集成部署奠定了实践基础。

**关键词:** OSGi, STKOS, 知识服务, 应用示范系统, 架构设计

DOI: 10.3772/j.issn.1673—2286.2012.12.007

## 1 引言

为适应新形势下科研环境和科研方式的转变, 满足科研用户对专业化、个性化、深层次知识服务的需求, 完善和拓展国家科技图书文献中心 (NSTL) 的服务功能和服务方式, 显著提升面向专业领域的知识服务能力, NSTL部分成员单位承担了国家“十二五”科技支撑计划课题“基于STKOS的知识服务应用示范”。该课题将基于NSTL强大的文献资源体系, 充分发挥科技知识组织系统 (STKOS) 在知识组织、知识关联、语义推理、知识挖掘等方面的优势, 开展科技监测、领域知识结构及其演化分析、领域学术关系网络分析、领域科研信息环境构建和关联数据服务等不同类型知识服务应用示范研究。

在课题研究过程中, 各类型知识服务应用示范研究由不同参与单位承担, 并各自负责设计开发相应的应用示范子系统。然而, 这些示范子系统彼此并不是孤立的, 需要进行相关信息资源和服务功能的广泛共享和交互。因此, 为实现各子系统间的无缝集成以及协同互操作<sup>[1]</sup>, 最大程度地提高应用示范的整体效果, 有必要基于类似OSGi这样先进的系统设计规范和框架, 对各应用示范子系统的数据交互接口和服务功能以规范化、标准化的方式进行设计、实现、整合和集成。

本文接下来简要介绍了OSGi规范及其实现框架, 对基于STKOS的知识服务应用示范系统的功能需求进行了概要分析, 并基于OSGi规范设计了应用示范系统的整体体系架构, 最后基于Felix框架开发了简易的原型系统, 验证了将各类型知识服务以模块化、组件式设计及开发, 并实现服务功能“热插拔”动态化管理的可行性和先进性<sup>[2]</sup>。

## 2 OSGi简介

开放服务网关协议 (Open Service Gateway Initiative, OSGi) 是OSGi联盟制定的基于Java语言的服务规范, 支持软件开发人员构建动态化、模块化的Java应用系统, 目标是实现组件级的复用<sup>[3]</sup>。基于中间件的OSGi技术提供了一个面向服务、基于组件的开发环境, 并提供标准化的方式来管理整个软件生命周期。基于OSGi开发的系统具有复杂度低、可重用、动态部署、可持续运行等特点。OSGi规范最早于2000年发布, 最新版本是2012年6月刚发布的R5<sup>[4]</sup>。

### 2.1 OSGi框架

OSGi规范包括OSGi标准和OSGi框架两部分。其中OSGi标准定义主要的服务工作及具体功能和职责;

\* 本文系国家“十二五”科技支撑计划项目课题“基于STKOS的知识服务应用示范” (编号: 2011BAH10B06) 的研究成果之一。

OSGi框架则提供平台无关性、动态代码装入和开发应用程序的上下文环境,实现在有限的资源设备上应用程序的开放和动态部署。

OSGi框架作为该规范的核心,为应用程序模块(Bundle)提供了一个标准环境,共划分为四个层次:执行环境层(Execution Environment)是Bundle应用运行所依赖的Java执行环境;组件模块层(Modules)定义Bundle应用的装载机制;生命周期管理层(Life Cycle)实现对Bundle的动态安装、启动、停止、升级和卸载;服务注册及服务层(Service Registry)定义各个Bundle在生命周期中的动态协作模型,可以注册发布服务和动态绑定服务,并且可以在服务注册状态改变时接受事件消息<sup>[5]</sup>。

## 2.2 Bundle模型

OSGi规范是基于插件式的软件架构,组件Bundle(亦称为模块或插件)是OSGi中重要元素之一,也是其基本组成单元,可通过动态安装、更新或停止Bundle来实现系统功能的热插拔。OSGi框架为每个Bundle提供了独立的类加载器(Classloader),这种机制使得Bundle支持热插拔,可实现Bundle的动态化管理、即插即用和即删即无。一般而言,一个Bundle通常包含三部分内容:Java类文件(源代码、Class文件)、非Java类文件(HTML、图片等)和MANIFEST.MF文件(用于向OSGi框架描述Bundle的有关信息和声明)。

在OSGi规范中,Bundle与上述四个层次有着密切联系,Bundle运行在Java环境之上,由模块层提供的类加载机制进行加载,它的启、停等状态由生命周期层进行管理,Bundle可以基于服务注册层来注册、获取和卸载服务.Bundle之间则可以通过Import Package、Export Package、Require Bundle和Bundle Activator等来共享Java类并管理模块的生命周期。

## 2.3 OSGi实现框架

目前基于OSGi规范实现的框架主要包括Apache Felix、Equinox、Spring DM和ApacheCXF-DOSGi。其中:

Apache Felix是Apache旗下的OSGi框架<sup>[6]</sup>,也是目前最全面的框架,提供的服务也最为全面,几乎涵

盖了OSGi R 4.2全部标准,只需要部署几个文件就可以运行,无论是开发还是调试都非常简便。

Equinox是Eclipse旗下的OSGi框架<sup>[7]</sup>,最大的优势在于它和Eclipse结合紧密,可在Eclipse开发环境中快速开发、部署和管理插件,被用于开发Eclipse插件的应用较多,开发Web程序则略有不足。

Spring DM是Spring旗下的OSGi框架<sup>[8]</sup>,最大特点是结合了Spring框架,但在系统运行过程中会出现过多的依赖性。Spring本身无论是类加载机制还是配置文件都与OSGi格格不入,若没有基于Spring项目的移植需求,不推荐使用。

此外,还有ApacheCXF-DOSGi<sup>[9]</sup>等日益成熟的分布式OSGi框架,也为应用系统的分布式开发和集成部署提供了基础环境和技术支持。上述几种框架都基于OSGi规范,实现的核心功能和使用方式基本一致。一般而言,OSGi组件(Bundle)可轻易从一个框架迁移到另一个框架。框架的不同主要体现在它们本身的设计和额外的服务功能上,因此,可根据应用系统的实际情况酌情选用。

## 3 基于OSGi的应用示范系统体系架构设计

### 3.1 系统需求分析

“基于STKOS的知识服务应用示范”课题,是基于NSTL海量科技文献资源和科技知识组织系统STKOS中的超级科技词表、领域本体和科研本体,在研究科技监测、领域知识结构及其演化、领域学术关系网络、领域科研信息环境和关联数据等相关模型基础上,设计开发不同类型但又密切相关的知识服务应用示范系统。尽管各应用示范系统相对独立,但仍基于共同的信息资源基础,存在大量信息资源和服务功能的共享和交互。因此,为实现各示范子系统面向同一领域进行集成部署的目标,拟将各项知识服务示范子系统以组件式、插件式开发,支持服务组件的即插即用和动态管理。

### 3.2 基于OSGi的体系架构

通过深入调研分析,结合本课题实际情况和应用需求,初步设计了基于OSGi规范的体系架构(如图1

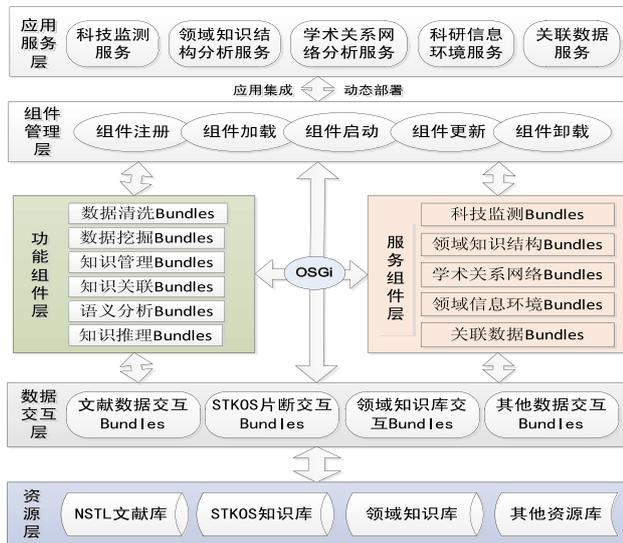


图1 基于OSGi的知识服务应用示范系统体系架构

所示), 以此来指导设计、实现和集成各应用示范系统的知识服务功能。

由图1可知, 为实现各应用示范子系统开发的自主性、灵活性和扩展性, 支持对各示范子系统插件式、松耦合地动态管理和集成部署, 体系架构共包括六层:

资源层由各示范系统要应用或产生的各类型、异构信息资源组成, 是整个体系架构的资源基础, 主要包括NSTL海量的科技文献资源数据库, 构建的STKOS超级科技词表、领域本体和科研本体等科技知识组织体系的知识库, 通过数据清洗、数据挖掘和知识关联等形成的领域知识库, 以及网络信息资源等其他资源。这些资源一般存储在关系数据库或XML、RDF等标准的结构化文档中。

数据交互层屏蔽了异构数据源的差异性, 为上层功能组件和服务组件访问资源层的各类型数据提供了标准统一的数据访问和交互接口, 支持对Oracle、MySQL、SQL Server等主流数据库, SKOS、RDF、OWL等格式化文档, 以及第三方开放数据接口的数据访问, 可提供多种途径和结果形式的数据访问方法。该层提供多类型数据访问和交互组件集(Bundles), 集成部署后可供其他层直接调用。

功能组件层是整个系统架构的核心层, 是在对各应用示范系统服务功能进行深入分析和抽象概括基础上, 设计并实现的各示范系统核心功能和公用功能组件集(Bundles)。一般包括面向领域的知识清洗、主题聚类、数据挖掘、知识管理、知识关联、语义分析、知识推理和可视化展示等组件集(Bundles)。该

层通过调用数据交互层的相关组件, 既能实现对资源层各类型信息资源的访问, 也可完成领域信息资源的清洗、加工和管理, 形成领域知识库。通过对这些核心或公共功能模块的组装, 即可形成提供特定服务的组件。

服务组件层是面向专业领域开展知识服务的核心组成部分, 是在集成整合数据访问层和功能组件层的基础上, 设计并实现的可直接面向用户提供知识服务的应用子系统, 实质上是由若干服务组件集(Bundles)组成, 主要包括科技监测、领域知识结构及其演化分析、学术关系网络分析、领域科研信息环境和关联数据服务等核心知识服务。该层的各个组件相对独立, 在系统功能和配套资源(图片和文档等方面)与一般应用系统没有区别, 主要是基于OSGi规范和框架来设计及实现, 各服务组件既可独立部署, 也可以面向特定领域选择性地组合集成部署。

组件管理层是整个体系架构的管理和调度中心, 全面负责数据交互层、功能组件层和服务组件层的协调管理, 实质上是对各应用示范系统中各类型组件(Bundle)进行集中管理, 提供组件的注册、安装、加载、启动、停止、更新、卸载等整个生命周期的管理功能。该层还提供对组件的存放、安装路径和初始化等配置参数, 插件描述、版本、版权等基本信息进行管理维护, 支持组件的动态管理, 最好还能对远程的组件进行分布式的加载和管理。

应用服务层面向最终用户提供各类型知识服务, 主要是对基于OSGi规范和框架实现的各类型组件进行应用集成和动态部署, 支持面向特定领域开展科技监测、领域知识结构及演化分析、学术关系网络分析、领域科研信息环境和关联数据等集成化、动态化、多类型、全方位的知识服务。该层可对各知识服务应用示范子系统的状态分别进行控制。

#### 4 基于OSGi的应用示范原型系统构建

在上述体系架构基础上, 本文还基于Apache旗下的Felix框架, 开发了基于OSGi的知识服务示范原型系统, 初步实现了对科技监测、领域知识结构演化分析、学术关系网络分析、领域科研信息环境和关联数据服务等知识服务子系统简易功能的模块化、组件式开发, 验证了将各应用示范系统进行模块化开发、对各服务组件的“热插拔”和动态化管理具有可行性和先进性。

表1 原型系统开发及运行环境列表

参数类型	技术参数
开发平台	MyEclipse 8.6
OSGi内核	标准的OSGi框架+Felix
Web容器	经Felix改写后的Jetty
插件管理	Apache Felix Web Console

## 4.1 系统环境

本原型系统基于的开发及运行环境,如表1所示。

在表1中,MyEclipse 8.6为创建图2中各组件(Bundle)提供了标准的OSGi框架,为各Bundle的代码编写、程序编译和调试提供了工作平台。Felix为遵循OSGi规范的各种Bundle提供了动态解析和运行的环境。轻量级的Web容器Jetty,经改写后更加符合OSGi要求,可以运用JSP、Servlet等包含有Web程序的Bundles。Apache Felix Web Console则是Felix子项目的成果,可基于网页浏览器用来动态监测和管理各个Bundle<sup>[10]</sup>。

## 4.2 知识服务组件的设计与实现

本原型系统共设计并简易实现了11个Bundles,如表2所示。

其中,KnowledgeServicesBasedOnSTKOS是整个原型系统的门户,其余是各类型知识服务定义的服务接口及其实现。由于只是原型系统,因此没有实现具体的知识服务功能,也没有定义更具体的数据访问组件和功能组件的接口及其实现。

在创建各知识服务接口定义及实现组件时,可配置文件MANIFEST.MF来定义各组件的自描述信息,如Bundle-Name设置组件的名称,Bundle-Vendor指定组件开发机构,Bundle-SymbolicName指定生成Jar文件的名称,Bundle-Version可指定组件的版本信息,Import-Package可指定该组件需要依赖的其他组件,Export-Package参数可将该Bundle要提供的服务暴露给其他组件。领域科研信息环境接口的配置文件如图2所示。

在具体实现各服务组件接口时,可在各Bundle的激活器Activator类的start启动和stop停止方法中进行服务的注册和卸载。

每个组件的具体功能可根据各类型知识服务的

表2 原型系统组件列表

组件名称	组件实例
基于STKOS的知识服务应用示范原型系统	NSTL_KnowledgeServicesBasedOnSTKOS Plug-in
科技监测服务接口	NSTL_SciTechMonitoringService Plug-in
科技监测服务接口实现	NSTL_SciTechMonitoringServiceImpl Plug-in
知识结构及演化分析服务接口	NSTL_KnowledgeStructureService Plug-in
知识结构及演化分析服务接口实现	NSTL_KnowledgeStructureServiceImpl Plug-in
学术关系网络服务接口	NSTL_AcademicNetworkService Plug-in
学术关系网络服务接口实现	NSTL_AcademicNetworkServiceImpl Plug-in
领域科研信息环境服务接口	NSTL_DomainInfoEnvironmentService Plug-in
领域科研信息环境服务接口实现	NSTL_DomainInfoEnvironmentServiceImpl Plug-in
关联数据服务接口	NSTL_LinkedDataService Plug-in
关联数据服务接口实现	NSTL_LinkedDataServiceImpl Plug-in

```

Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: NSTL_DomainInfoEnvironmentService Plug-in
Bundle-SymbolicName: nstl.stkos.knowledgeservice.domaininfoenvironment
Bundle-Version: 1.0.0
Bundle-Vendor: NSTL_农科
Bundle-RequiredExecutionEnvironment: JavaSE-1.6
Import-Package: org.osgi.framework;version="1.3.0"
Export-Package: nstl.stkos.knowledgeservice.domaininfoenvironment

```

图2 领域科研信息环境接口组件的自描述信息

具体需求进行业务逻辑的封装和方法的实现。完成程序开发后,可通过MyEclipse的Export功能将上述各组件导出“Deployable plug-ins and fragments”Jar文件,以便进行部署或供其他组件调用。该文件以Bundle的Bundle-SymbolicName和Bundle-Version进行组合命名,如“nstl.stkos.knowledgeservice.domaininfoenvironment\_1.0.0.jar”。

## 4.3 原型系统的试运行与分析

完成上述各服务接口和实现组件的简易开发后,通过MyEclipse的Export功能,生成了11个Jar文件。直接在MyEclipse中启动Felix项目后,通过网页浏览器访问“http://localhost:8080/system/console/bundles”,即可实现对这些组件的动态装载、启动、停止、更新和卸载等操作,如图3、4所示。



图3 基于Apache Felix Web Console的Bundle动态管理



图4 各服务组件的运行状态监控

通过试运行不难发现,OSGi规范及其实现框架为应用系统插件式的体系设计和系统实现提供了标准规范和支撑环境。在该系统中,每个Bundle对应一个独立的应用功能模块,每个Bundle可视为一个组件。应用系统的开发与部署都可以从OSGi所提供的标准规范中受益,从而给系统带来诸多优势,如:每个组件都是独立的,应用代码结构更加规范与合理;每个组件都可以在运行期动态加载与卸载,应用部署更加灵活;每个组件都具有独立的版本号,版本升级、应用规划与管理更加方便;每个组件都具有完善的安全策略,应用系统的整体更加安全。

#### 参考文献

- [1] 史纪强,何兴曙,王志琼,等.基于插件技术的企业应用集成架构研究[J].计算机与应用化学,2012,29(2):191-194.
- [2] 梁淑英,丁红海,饶若楠.一种基于OSGi的医疗设备集成框架[J].计算机应用与软件,2010,27(2):90-93,117.
- [3] 林昊,曾宪杰.OSGi原理与最佳实践[M].北京:电子工业出版社,2009.
- [4] OSGi Alliance | Release5 / HomePage [EB/OL]. (2011-12-19) [2012-06-24]. <http://www.osgi.org/Release5/HomePage>.
- [5] 张辉,刘万军,吕欢欢.一种基于OSGi的服务动态化模型应用[J].计算机系统应用,2011,20(12):144-147,140.
- [6] Apache Felix [EB/OL].(2011-03-04)[2012-06-25]. <http://felix.apache.org/site/index.html>.
- [7] Equinox [EB/OL].(2012-01-01)[2012-06-25]. <http://eclipse.org/equinox/>.
- [8] Spring Dynamic Modules [EB/OL].(2012-01-01)[2012-06-25].<http://www.springsource.org/osgi>.
- [9] Apache CXF -- DOSGi Releases[EB/OL].(2012-04-10)[2012-06-25].<http://xf.apache.org/dosgi-releases.html>.
- [10] Apache Felix Web Console[EB/OL].(2011-12-19)[2012-06-24]. <http://felix.apache.org/site/apache-felix-web-console.html>.

#### 作者简介

鲜国建 (1982-), 男, 硕士学历。研究方向: 知识组织、关联数据、数字资源加工、信息系统开发。E-mail: xgj@mail.caas.net.cn

赵瑞雪 (1968-), 博士, 研究员, 博士生导师, 研究方向: 信息管理与信息系统、信息资源管理、知识组织及数字图书馆。E-mail: zhaorx@mail.caas.net.cn

#### Architecture Design and Implementation of the Application and Demonstration System of Knowledge Services Based on OSGi

Xian Guojian, Zhao Ruixue, Kou Yuantao, Zhu Liang / Agricultural Information Institution of CAAS, Beijing, 100081

Abstract: This paper gives brief introduction about the OSGi, and its bundle module and several realized frameworks. The architecture of the application and demonstration system of knowledge services based on OSGi was designed, including information resource layer, data interoperation layer, function modules layer, service modules layer, modules management layer and application layer. Finally, a knowledge service prototype system was realized, and all modules are developed along with the OSGi specifications, which proves that it is feasible and advanced to manage the modules dynamically and plug and play based on OSGi.

Keywords: OSGi, STKOS, Knowledge services, Application and demonstration of system, Architecture design

(收稿日期: 2012-08-01)

## 5 结语

为实现基于STKOS的各类型知识服务应用示范子系统以模块化、组件式方式进行设计与开发,支持各类型功能组件和服务组件的即插即用和动态管理,最终将各示范子系统面向同一领域进行集成部署,本文引入OSGi规范,对此开展了初步研究,初步设计了基于OSGi的知识服务应用示范系统的体系架构,并基于Felix框架实现了一个简单的知识服务原型系统,初步完成对科技监测、领域知识结构演化分析、学术关系网络分析、领域科研信息环境和关联数据服务等知识服务示范子系统的模块化、组件式开发,验证了对各服务功能实现即插即用、动态化管理的可行性和先进性。

本原型系统离真实的应用系统还有很大差距,接下来还需进一步细化各知识服务应用示范子系统的功能需求,详细设计数据交互组件、功能组件和服务组件的数据交互和功能调用关系及接口标准,并结合Java应用系统开发技术,开展各类型Bundles的具体功能实现,并在Bundles的仓储管理、分布式部署等方面开展进一步研究与实践,为面向专业领域形成可扩展、可分布、可动态管理部署的组件集群和知识服务应用示范门户奠定实践基础。