

基于服务模板的制造资源封装方法

孔令军*, 徐文胜, 查建中

(北京交通大学 机械与电子控制工程学院, 北京 100044)

(* 通信作者电子邮箱 08116294@bjtu.edu.cn)

摘要:为规范和加快制造资源的封装过程,提出了一种新型的基于服务模板的制造资源封装方法。首先,根据制造资源的特点,给出了服务模板的概念、结构及类型;接着,提出了利用已有服务程序进行模板抽取的过程,定义了一种用于模板抽取的标记语言;然后,给出了基于服务模板的制造资源封装过程,将制造资源封装为基于面向服务体系架构的制造服务。最后,通过实例证明,提出的方法不仅可以规范制造服务的开发过程,而且可充分利用已有的制造资源服务程序,加快制造资源的封装进程。通过采用提出的方法,普通的产品开发人员就可以完成制造资源的服务化封装,而无需专业性的服务封装开发编程知识。该方法可为网络环境下的制造资源共享提供基础性支持。

关键词:资源封装;模板;网络化制造;抽取;面向服务的体系架构

中图分类号: TP391; TP311.5 **文献标志码:** A

Encapsulation method of manufacturing resources based on service templates

KONG Ling-jun*, XU Wen-sheng, CHA Jian-zhong

(School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: To standardize and accelerate the encapsulation process of manufacturing resources, a new encapsulation method of manufacturing resources was proposed based on service templates. The concept, structure and types of service templates were given according to the characteristics of manufacturing resources. Then the service template extraction procedure which utilized the existing programs of manufacturing services was proposed, and a service template extraction language was defined. The encapsulation procedure of manufacturing resources based on service templates was proposed, and the manufacturing resources were encapsulated to service-oriented architecture-based manufacturing services. At last the application case shows that the proposed method can standardize the development procedure of manufacturing services and accelerate the encapsulation process of manufacturing resources by fully utilizing the existing programs of manufacturing services, so ordinary product developers can encapsulate manufacturing resources without special knowledge of software programming. This method can provide fundamental support for the sharing of manufacturing resources in the network environment.

Key words: resource encapsulation; template; networked manufacturing; extraction; Service-Oriented Architecture (SOA)

0 引言

随着网络化制造的发展^[1-5],大量的制造服务程序被开发出来。如文献[6]提出了一种基于移动 Agent 的制造资源封装方法;文献[7]将制造资源封装为九大类 Web 制造服务;文献[8-10]研究了基于 Web 服务资源框架(Web Services Resource Framework, WSRF)的制造资源封装方法;文献[11]基于面向服务对象架构(Service Object-Oriented Architecture, SOOA)将制造资源封装为一种动态服务;文献[12]提出了面向云制造软资源的封装方法。为简化制造资源的封装工作,大部分的研究方法都是采用基于模板的制造资源封装方法,如文献[6-7,10,12]。但已有研究方法没有考虑到如何利用已有的制造服务程序,将其抽取为模板加以利用,以进一步加快制造资源的封装进程。

为制造资源建立服务模板是一项很有必要的工作。一般说来制造服务程序的开发成本较高,不仅要考虑传统服务程

序的开发方面,如技术选型、编码实现、单元测试等,还要考虑如何通过服务程序来调用制造资源提供的开放接口,以实现具体制造资源或制造功能的封装。但用于封装一类制造资源或其固定功能的服务程序基本上是一致的,只是在与具体服务类型信息、服务资源信息等方面有着些许的不同。故可以利用已有的制造服务程序,建立面向固定资源或固定功能的服务模板,以加快同种类制造资源或制造功能的封装。这样一方面加快了制造资源的封装进程,另一方面也规范了制造服务的开发过程,为制造资源的集成共享提供支持。

为解决上述问题,本文提出了一种基于服务模板的制造资源封装方法。该方法可根据已有制造服务程序的特点,将其抽取为不同类型的服务模板,再基于抽取得到的服务模板对其他制造资源进行封装,得到新的制造服务部署包。在封装实例中,通过对已有 ANSYS 服务程序进行分析,利用提出的服务模板抽取语言将其抽取为面向工程软件的制造接口服务模板,并基于该服务模板对 HyperMesh 软件资源进行了封

收稿日期:2012-07-30;修回日期:2012-09-26。 基金项目:国家自然科学基金资助项目(51175033)。

作者简介:孔令军(1986-),男,辽宁鞍山人,博士研究生,主要研究方向:网络化制造、SOA、多学科优化设计; 徐文胜(1970-),男,湖北鄂州人,副教授,博士,主要研究方向:并行工程、网络化制造、信息系统安全; 查建中(1947-),男,北京人,教授,博士,主要研究方向:智能工程及其应用、优化算法。

装,验证了本文方法的有效性。

1 基于服务模板的制造资源封装

基于服务模板的制造资源封装过程如图 1 所示。首先,对已有制造服务程序进行分析,通过服务模板抽取语言对其进行标记,替换制造服务程序在代码文件、配置文件、项目构建文件中与具体服务类型和服务资源有关的信息,形成可以重复利用的服务模板;然后,通过服务模板库对服务模板进行管理,实现服务模板的添加、查询等操作功能;最后,根据资源封装要求查询得到服务模板,提供服务模板中所需的与具体服务类型和服务资源有关的封装信息,利用模板引擎采用“变量替换”的方式生成新的制造服务程序,再利用项目构建文件对生成的制造服务程序进行编译、打包,生成新的制造服务部署包。下面分别详细介绍服务模板的概念、结构及类型,服务模板的抽取过程和基于服务模板的制造资源封装过程。

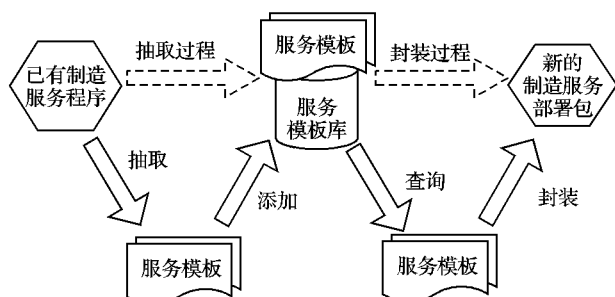


图 1 基于服务模板的制造资源封装模式

1.1 服务模板的概念、结构及类型

定义 1 服务模板。服务模板是对制造服务程序的一种抽象,其屏蔽掉了与具体服务类型和服务资源相关的信息。

从技术角度来说,服务模板一般来自于两个工程项目:一个项目提供服务提供方模板的相关内容;而另一个项目提供服务请求方模板的相关内容。基于两个项目中的内容,本文将服务模板的结构模型表示为:

ServiceTemplateModel =

⟨Code, Configuration, Dependency, Build⟩

其中:Code 指服务程序模板,包括服务提供方程序模板和服务请求方程序模板两部分,这样设计的好处是:当开发利用服务的程序时,可以直接利用服务请求方模板生成的服务请求方程序调用服务提供方程序,而无需考虑具体的服务实现技术,降低了分布式服务程序开发的难度;Configuration 指服务配置模板,抽象出与具体制造服务和制造资源相关的配置信息的属性项,利用封装信息中的配置属性值生成新的制造服务的配置文件;Dependency 指服务依赖代码包,是服务程序所

依赖的第三方类库,如 Java 中的 Jar 包、C 语言中的 lib 库等;Build 指项目构建模板,从程序的角度定义了项目的基本信息,为封装过程的编译、打包提供支持,如 Apache Maven 项目的 pom.xml、Java 构建工具 Ant 的 build.xml、Linux 下 Make 中的 Makefile 等。

简单来说,制造服务程序是采用服务技术基于制造资源提供的开放接口对其功能的一种封装。由于制造资源提供的开放接口特点不同,制造服务程序对制造资源的封装层次也有所不同,具体可分为面向功能型制造服务程序、面向资源型制造服务程序和面向接口型制造服务程序。面向功能型制造服务程序是指制造服务程序基于制造资源提供的开放接口,对制造资源的某项具体功能进行了服务化封装,如对疲劳强度分析机的分析能力进行封装的制造服务程序。面向资源型制造服务程序是指制造服务程序对某种制造资源提供的接口进行了服务化的封装,如对知识库资源的数据库接口进行了封装的制造服务程序。面向接口型服务程序是指制造服务程序对一类制造资源提供的接口进行了服务化封装,如对工程软件提供的命令行接口(Command-Line Interface, CLI)进行了封装的制造服务程序。

不同类型的制造服务程序可抽取为不同层次的服务模板。面向功能型制造服务程序可抽取为制造功能服务模板,它可用于同种制造资源的功能的封装。面向资源型制造服务程序可抽取为制造资源服务模板,它可用于同种制造资源的接口的封装。面向接口型制造服务程序可抽取为制造接口服务模板,它可用于对该接口支持的多种制造资源的封装。下面分别给出各类型服务模板的定义。

定义 2 制造功能服务模板 (Manufacturing Function Service Template, MST-F)。制造功能服务模板是对一种具体制造资源功能的服务程序的抽象,其与制造资源和制造功能是一一对应的关系,是服务模板的基本类型,对资源的抽象层次较低。

定义 3 制造资源服务模板 (Manufacturing Resource Service Template, MST-R)。制造资源服务模板是对一种制造资源的接口的服务程序的抽象,其与制造资源种类之间是一一对应的关系。

定义 4 制造接口服务模板 (Manufacturing Interface Service Template, MST-I)。制造接口服务模板是对一类制造资源的接口的服务程序的抽象,其与制造资源接口是一一对应的关系,与制造资源种类是一对多的关系,对资源的抽象层次较高。

表 1 制造服务模板类型

| 类型 | 名称 | 说明 | 用途 | 抽象层次 |
|-------|----------|-----------------|---------------------|------|
| MST-F | 制造功能服务模板 | 对某种制造资源的具体功能的抽象 | 用于同种制造资源的功能的封装 | 低 |
| MST-R | 制造资源服务模板 | 对某种制造资源的接口的抽象 | 用于同种制造资源的接口的封装 | 中 |
| MST-I | 制造接口服务模板 | 对某类制造资源的接口的抽象 | 用于对该类接口支持的多种制造资源的封装 | 高 |

1.2 抽取过程

定义 5 服务抽取。服务抽取是指利用已有的制造服务程序,将程序中与具体服务类型和服务资源相关的耦合信息使用模板变量进行替代,从而得到服务模板的过程。

服务模板抽取语言 (Service Template Extraction Language, STEL) 是指一组具有特定语义的服务模板变量,可用于对制

造服务程序中的耦合信息进行替代,以实现服务抽取。由于制造服务的开发技术较多,提出一种适用于所有开发技术的 STEL 是不现实的。本文基于 Velocity 模板语言 (Velocity Template Language, VTL)^[13] 提出一种面向 SOOA 技术^[14] 的 STEL。对于其他技术开发的制造服务程序,可基于提出的 STEL 作进一步地扩展或修改,以实现其他技术类型的服务抽取。

本文提出的 STEL 的目标是给出一组模板变量,用于对已有 SOOA 制造服务程序中与服务类型和服务资源的相关信息进行替代,以形成制造功能服务模板、制造资源服务模板或制造接口服务模板。STEL 包括两部分:一部分是面向服务类型的服务模板抽取语言(STEL Type, STEL-T);另一部分是面

向服务资源的服务模板抽取语言(STEL Service, STEL-S)。表 2 列出了 STEL-T 的基本语素、类型及说明。表 3 列出了 STEL-S 的基本语素、类型及说明。图 2 为某 SOOA 制造服务的基于 Jini Configuration^[15]形式的制造服务配置模板片段,该模板采用提出 STEL 进行了模板化标记。

表 2 STEL-T 基本语素说明

| VTL | 名称 | 类型 | 说明 |
|------------------------|--------|-------------|-------------|
| \${ RequestorPackage} | 请求方包名 | Java 包命名规则 | 作为服务请求方的包名 |
| \${ ServiceInterface} | 服务接口名 | Java 类命名规则 | 作为服务的接口名 |
| \${ RequestorClass} | 请求方类名 | Java 类命名规则 | 作为服务请求方的类名 |
| \${ RequestorMethod} | 请求方方法名 | Java 方法命名规则 | 作为服务请求方的方法名 |
| \${ ProviderPackage} | 提供方包名 | Java 包命名规则 | 作为服务提供方的包名 |
| \${ ProviderClass} | 提供方类名 | Java 类命名规则 | 作为服务提供方的类名 |
| \${ ProviderMethod} | 提供方方法名 | Java 方法命名规则 | 作为服务提供方的方法名 |
| \${ InterfaceTemplate} | 接口模板 | String | 资源的接口模板 |

表 3 STEL-S 基本语素说明

| VTL | 名称 | 类型 | 说明 |
|-----------------------|-------|--------------------------------------|--------------------------|
| \${ ResourceID} | 资源 ID | String | 资源的 ID |
| \${ ServiceName} | 服务名称 | String | 服务的名称 |
| \${ ResourceType} | 资源类型 | Enum(Intelligence, Knowledge, Tool) | 资源类型,为智力资源、知识资源和工具资源三者之一 |
| \${ HostID} | 主机 ID | String | 用于描述资源所在的主机 ID |
| \${ ResourcePath} | 资源路径 | String | 资源所在路径的具体信息 |
| \${ ResourceVersion} | 资源版本 | String | 用于标记资源的版本号 |
| \${ InstanceNo} | 实例数量 | Integer | 支持同时服务的最大实例数 |
| \${ ResourceFunction} | 资源功能 | String | 资源提供功能的详细描述 |
| \${ LookupGroup} | 查找组 | String | 服务注册时的查找组信息 |
| \${ LeaseTime} | 租约时间 | Integer | 服务的租约时间,以秒为单位 |

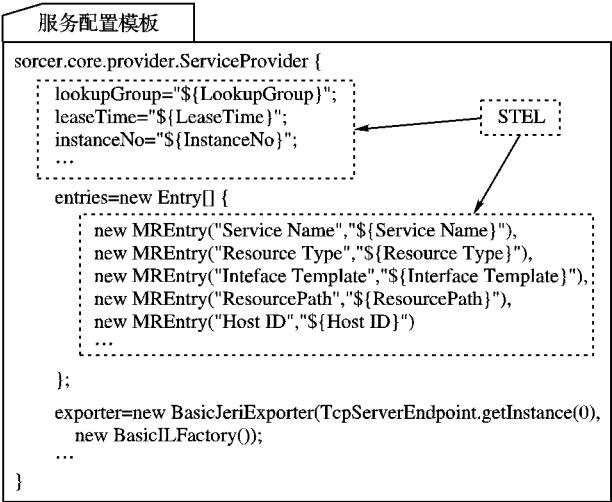


图 2 基于 STEL 的服务配置模板片段

1.3 封装过程

定义 6 服务封装。服务封装是指基于服务抽取得到的服务模板,将具体服务类型和服务资源的相关信息与模板变量进行替换,进而经过编译、打包,得到制造服务部署包的过程。

基于服务模板的制造资源封装过程,如图 3 所示,包括如下步骤:

- 1) 根据要封装的制造资源信息和制造功能信息,从服务模板库中查询、下载服务模板;
- 2) 基于具体服务类型和服务资源的封装信息,利用 Velocity 模板引擎对服务模板中的 STEL 模板变量进行替换,得到新的服务程序;

3) 基于新得到的服务程序中的项目构建文件,对服务程序进行编译、打包,得到制造资源的服务部署包,完成制造资源逻辑上的封装,待进一步部署成为物理上的制造服务。



图 3 基于服务模板的制造资源封装过程

2 封装实例

制造资源的模板化封装是一项十分复杂的工作。其一,建立服务模板的过程,或是从已有服务程序中抽取,或是新建开发服务模板,这都要求对服务的开发及资源的调用过程有着深入的了解;其二,得到服务模板后需要进一步地分析,确定服务模板的使用范围,从而得到不同类型的服务模板,这需要模板建立人员对制造资源及其接口的功能有着深入的了解;其三,当模板需求者利用服务模板封装新的制造资源时,需要对新资源有着深入的了解,以提供服务模板所需的封装信息,从而生成新的制造服务部署包。下面以一个封装实例详细介绍制造资源的模板化封装过程。

2.1 已有 ANSYS 服务分析

可以利用已开发的服务程序,应用 STEL 标记得到的服务模板。在文献[11]中已开发了基于 SOOA 技术的 ANSYS 服务,对该服务程序进行分析,为服务模板的抽取工作做准备。该 ANSYS 服务的执行过程如图 4 所示。首先,ANSYS 服务从工作空间中下载作业文件,包括基于 ANSYS 参数化设计语言(ANSYS Parametric Design Language, APDL)的作业分析文件和 mesh 网格文件,准备调用资源软件;然后,ANSYS 服务通

过 ANSYS 软件提供的 CLI 调用资源, ANSYS 的 CLI 支持 APDL 格式的作业请求提交;最后,在 ANSYS 完成有限元分析后,将作业结果文件上传到工作空间中,包括 jpeg 应变分析云图、avi 应变动画、计算日志等。

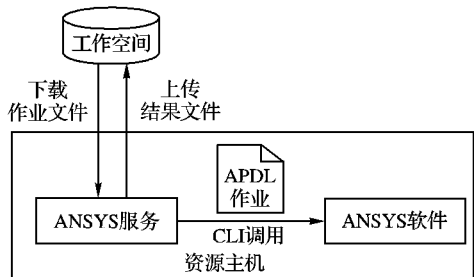


图 4 ANSYS 服务的执行过程

不难发现,以上描述的服务执行过程不只适用于 ANSYS 软件,对大部分的工程软件都是适用的。另外 CLI 是一种在没有计算机操作系统前普遍存在的资源应用接口,大部分的商用工程软件和自编工程软件都对其提供了良好的支持,并提供了面向 CLI 的作业文件规范,如上述的 APDL 文件、ProENGINEER 软件中的 trail 文件、HyperMesh 软件中的 cmf 文件、Matlab 软件中的 m 文件等,可以基于 CLI 编写、提交不同的作业任务。

经过分析可知,已有的 ANSYS 服务程序可以抽象成为一种面向工程软件 CLI 的制造接口服务模板,应用该模板不仅可以封装其他的 ANSYS 软件资源,还可以封装如 ProENGINEER、HyperMesh、Matlab 等其他种类的工程软件资源。

2.2 服务模板抽取

ANSYS 服务程序中需要进行模板化标记的文件包括服务请求方代码文件、服务提供方代码文件和服务配置文件,应用 STEL 对上述文件进行标记,屏蔽掉与具体 ANSYS 服务类型和服务资源相关的信息,得到面向工程软件 CLI 的制造接口服务模板。其结构如图 5 所示,包括代码模板、配置模板和接口模板。

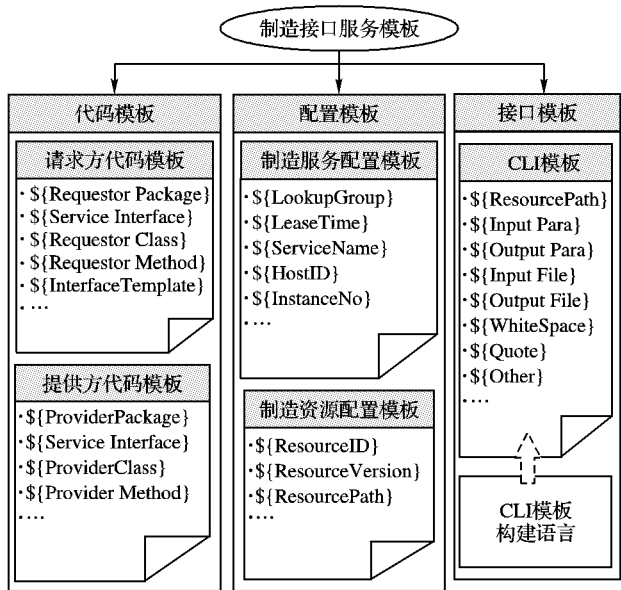


图 5 面向工程软件 CLI 的制造接口服务模板结构

代码模板用于生成基于 SOOA 的面向工程软件资源的 CLI 调用功能的服务代码,包括请求方代码模板和提供方代码模板两部分,代码模板中实现了 CLI 调用功能和 SOOA 制

造服务的相关功能,如服务动态注册、服务自动续约、服务远程调用、作业文件下载和作业结果文件上传等,主要的模板信息包括请求方代码和提供方代码的包名、服务接口名、类名、方法名以及接口模板,服务提供方代码利用接口模板动态组建资源接口,调用所封装的资源功能;配置模板用于生成新的工程软件服务的配置文件,包括制造服务配置模板和制造资源配置模板,将配置信息进行分类组织,实现制造服务与制造资源之间配置信息的解耦,制造服务配置模板用来设置与制造服务的相关信息,如服务动态注册组、服务租约时间、服务名称、服务主机 ID、服务实例数等,制造资源配置模板用来设置制造资源的属性信息,如资源 ID、资源版本号、资源路径等;接口模板这里采用的是 CLI 模板,为具体工程软件的 CLI 调用提供支持。本文基于 VTL 提出一套 CLI 模板构建语言 (CLI Template Construct Language, CLI-TCL),为实现工程软件资源的自动化封装及调用提供支持,CLI-TCL 的基本语素如表 4 所示。

表 4 CLI-TCL 基本语素说明

| VTL | 名称 | 说明 |
|-------------------|------|-----------------|
| \${ InputFile} | 输入文件 | CLI 的作业输入文件名 |
| \$(OutputFile) | 输出文件 | 指定 CLI 的作业输出文件名 |
| \$(ResourcePath) | 作业路径 | 工程软件资源的具体路径 |
| \${ InputPara} | 输入参数 | CLI 中的输入参数 |
| \$(OutputPara) | 输出参数 | CLI 中的输出参数 |
| \$(WhiteSpace) | 空格 | CLI 中的空格占位符 |
| \$(Quote) | 引号 | CLI 中的引号符 |
| \$(Other) | 其他 | 用于语言的进一步扩展 |

应用 CLI-TCL 可以构建面向不同工程软件的命令行接口模板。表 5 所示为应用 CLI-TCL 构建的 ANSYS、HyperMesh、ProENGINEER 工程软件的 CLI 模板。

表 5 ANSYS、HyperMesh、ProENGINEER 的 CLI 模板

| 工程软件 | CLI 模板 |
|-------------|---|
| ANSYS | <pre>\$(Quote) \$(ResourcePath) \$(Quote) \$(WhiteSpace) \$(InputPara) \$(WhiteSpace) \$(InputPara) \$(WhiteSpace) \$(InputFile) \$(WhiteSpace) \$(OutputPara) \$(WhiteSpace) \$(OutputFile)</pre> |
| HyperMesh | <pre>\$(Quote) \$(Quote) \$(ResourcePath) \$(Quote) \$(Quote) \$(WhiteSpace) \$(InputPara) \$(InputFile) \$(WhiteSpace) \$(InputPara)</pre> |
| ProENGINEER | <pre>\$(Quote) \$(ResourcePath) \$(Quote) \$(WhiteSpace) \$(InputPara) \$(WhiteSpace) \$(InputFile)</pre> |

图 6 为基于已有 ANSYS 服务抽取得到的服务接口模板,图 7 为抽取的服务配置模板。服务提供方代码模板和服务请求方代码模板的抽取原理同上,本文就不一一赘述。

2.3 基于服务模板的 HyperMesh 封装

2.2 节抽取得到的面向工程软件 CLI 的制造接口服务模板,可以用来封装 HyperMesh 工程软件。HyperMesh 的封装过程本质上是在一定的封装环境下将与具体服务类型和服务资源相关的信息不断地与制造接口服务模板中的 STEL 进行替换,以得到新的制造服务程序,并对其进行编译、打包生成制造服务部署包的过程,如图 8 所示,包括两个层次的封装。

第一层封装,将制造接口服务模板标准化为制造资源服务模板,封装信息为与服务类型相关的信息,主要包括服务提供方和服务请求方程序的包名、接口名、方法名和 CLI 模板等

信息,基于该模板利用 Velocity 模板引擎结合封装信息替换得到面向 HyperMesh 工程软件的制造资源服务模板,该模板

具有面向 HyperMesh 资源的标准化服务接口,可用于进一步封装生成面向具体 HyperMesh 软件的制造服务部署包。

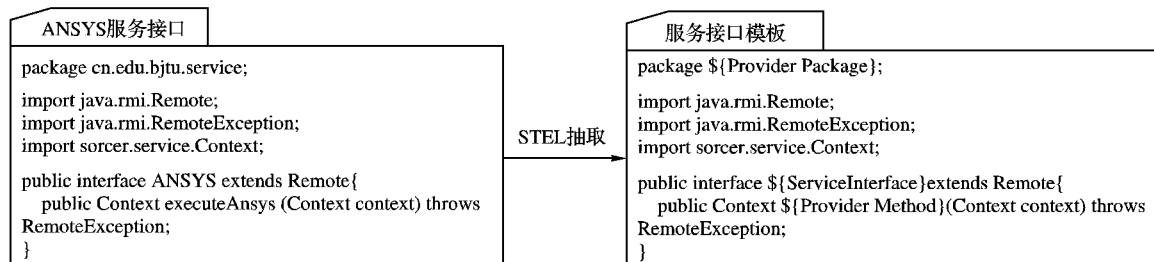


图6 服务接口模板抽取

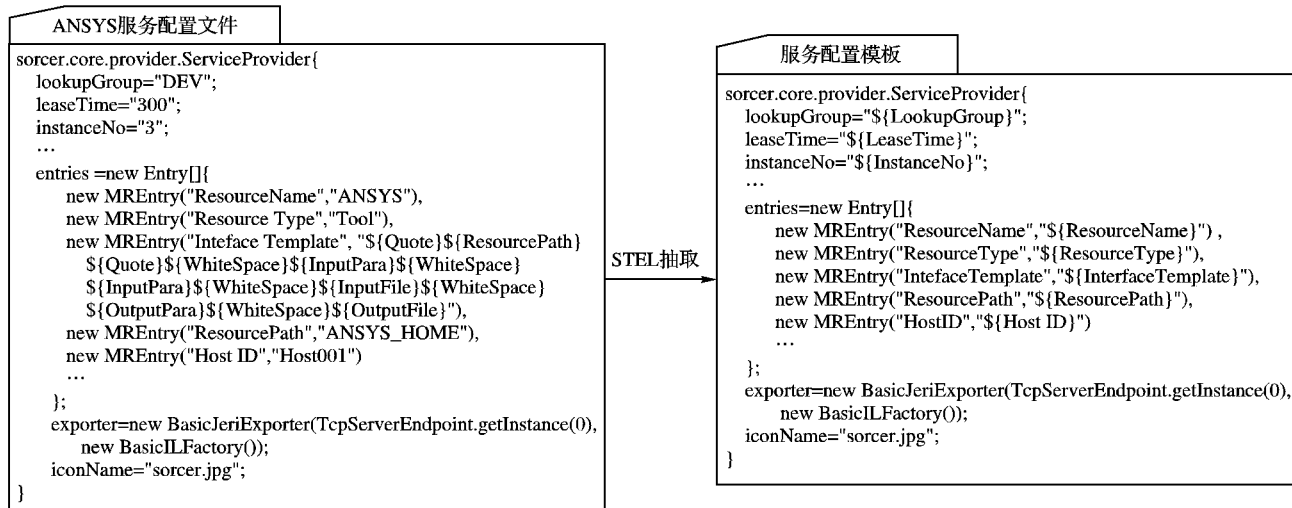


图7 服务配置模板抽取

第二层封装,将制造资源服务模板实例化为制造服务部署包,封装信息为与服务资源相关的信息,主要包括资源名称、资源路径、资源描述、实例数、注册组信息、租约时间、服务实例数等,制造服务部署包是与主机上的具体制造资源相对应的。

先利用 Velocity 模板引擎替换制造资源服务模板中的封

装信息得到新的制造服务程序,再基于 Ant 调用项目的构建文件,完成制造服务程序的编译、打包,得到 HyperMesh 软件制造服务部署包。当制造服务部署包被部署到服务容器后,会读取服务配置文件中的信息以动态构建制造服务,在制造服务接受请求后会基于 CLI 模板动态生成制造资源的调用接口,以响应资源的服务请求。

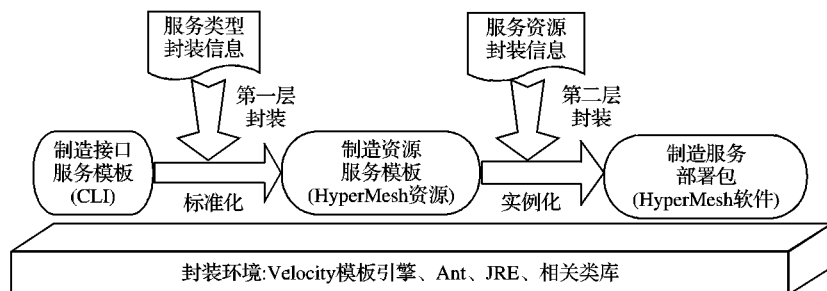


图8 基于服务模板的 HyperMesh 软件封装

2.4 与其他方法比较

与已有基于模板的制造资源封装方法^[6-7,10,12]相比,本文方法具有如下特点:1)模板多层次性。根据制造资源特点,对服务模板进行了分类,并构建了不同类型模板间的抽象层次关系,使服务模板不再只与单一的资源功能相对应,而可以与制造资源的类型或接口相对应,扩大了服务模板的应用范围;2)服务可重用性。可通过 STEL 对已有的制造服务程序进行服务模板抽取,使其得到最大限度的重复利用,省去了很多重复的模板开发工作;3)资源适应性。采用 SOOA 作为服务模板的底层服务实现技术,利用 SOOA 的协同定位中立、访问协议中立和服务执行中立三大特点^[16],可很好地适应制造资源的高度动态性和协议独立性,为网络环境下的制造资

源高效共享提供支持。

在原开发的资源集成平台^[17]中,开发制造服务需要专业的软件工程师和制造工程师相配合才能完成,所需专业性知识强。在通过本文方法进行改进的资源集成平台^[18]中,普通的产品开发人员就可以基于服务模板完成制造资源的服务化封装,而无需与资源封装相关的专业知识,进一步规范和加快了制造资源的封装过程。

3 结语

网络化是制造业的主要发展方向之一,它的目标是实现制造资源在全球范围内的优化配置。本文针对网络化制造中的资源封装问题,提出了一种基于服务模板的制造资源封装

方法,该方法具有模板多层次性、服务可重用性和资源适应性的特点,不仅可以加快、规范制造资源的封装工作,还可以将已有制造服务程序抽取为服务模板加以利用,为实现网络环境下的制造资源的集成共享提供支持。该方法对其他领域的资源封装也具有一定的借鉴意义。

参考文献:

- [1] 范玉顺, 张立晴, 刘博. 网络化制造与制造网络[J]. 中国机械工程, 2004, 15(19): 1733 - 1738.
- [2] HUANG G Q, WRIGHT P K, NEWMAN S T. Wireless manufacturing: a literature review, recent developments, and case studies [J]. International Journal of Computer Integrated Manufacturing, 2009, 22(7): 579 - 594.
- [3] 谭伟, 姚锡凡, 李勇. 制造网格 ASP 模式的无缝资源发现结构研究[J]. 计算机应用, 2009, 29(7): 1966 - 1969.
- [4] CAMARINHA-MATOS L M. Collaborative networked organizations: status and trends in manufacturing[J]. Annual Reviews in Control, 2009, 33(2): 199 - 208.
- [5] TAO F, ZHANG L, NEE A Y C. A review of the application of grid technology in manufacturing[J]. International Journal of Production Research, 2010, 49(13): 4119 - 4155.
- [6] 周光辉, 江平宇. 基于移动 Agent 的网络化制造资源的封装和集成[J]. 计算机集成制造系统, 2002, 8(9): 728 - 732.
- [7] TAO F, HU Y F, ZHOU Z D. Study on manufacturing grid and its resource service optimal-selection system[J]. International Journal of Advanced Manufacturing Technology, 2008, 37(9/10): 1022 - 1041.
- [8] KUMAR R, YANG Z, ZHANG J, *et al.* Virtualization for manufacturing Web services: a WS-RF approach[J]. International Journal of Information Technology, 2005, 11(3): 40 - 51.
- [9] 井浩, 张璟, 李军怀. 基于 WSRF 的网络化制造资源 Web 服务封装的研究[J]. 计算机工程与应用, 2007, 43(15): 22 - 25.
- [10] 武蕾, 孟祥旭, 刘士军. 制造网格中资源服务化封装方法研究[J]. 计算机集成制造系统, 2008, 14(9): 1837 - 1844.
- [11] 于加晴, 查建中, 陆一平, 等. 面向复杂产品的分布式协同设计系统[J]. 中南大学学报: 自然科学版, 2010, 41(2): 539 - 546.
- [12] 陈琨, 王东勃, 王颖慧, 等. 云制造软资源封装研究[J]. 中国制造业信息化: 学术版, 2012, 41(5): 58 - 63.
- [13] GLASS-HUSAIN W. The Apache velocity project [EB/OL]. [2012-07-25]. <http://velocity.apache.org>.
- [14] SOBOLEWSKI M. Exertion oriented programming[J]. International Journal on Computer Science and Information Systems, 2008, 3(1): 86 - 109.
- [15] NEWMARCH J. Foundations of Jini 2 Programming[M]. Berkeley: Apress, 2006.
- [16] SOBOLEWSKI M. SORCER: computing and metacomputing inter-grid[C]// Proceedings of the 10th International Conference on Enterprise Information Systems. Berlin: Springer-Verlag, 2008: 74 - 85.
- [17] YU JIAQING, CHA JIANZHONG, LU YIPING, *et al.* A CAE-integrated distributed collaborative design system for finite element analysis of complex product based on SOOA[J]. Advances in Engineering Software, 2010, 41(4): 590 - 603.
- [18] KONG LINGJUN, XU WENSHENG, CHA JIANZHONG, *et al.* A resource integration platform for manufacturing grid based on SOOA [C]// Proceedings of the 1st International Conference on Electronic and Mechanical Engineering and Information Technology. Piscataway: IEEE, 2011: 1466 - 1469.

(上接第 3528 页)

了 SPARDL 模型的有效性。

| Element Name | Total | Auto | Manual | Reviewed | Undischarged |
|--------------|-------|------|--------|----------|--------------|
| spardl | 84 | 84 | 0 | 0 | 0 |
| c0 | 0 | 0 | 0 | 0 | 0 |
| c1 | 0 | 0 | 0 | 0 | 0 |
| c2 | 0 | 0 | 0 | 0 | 0 |
| c3 | 0 | 0 | 0 | 0 | 0 |
| c5 | 0 | 0 | 0 | 0 | 0 |
| m0 | 6 | 6 | 0 | 0 | 0 |
| m1 | 10 | 10 | 0 | 0 | 0 |
| m2 | 12 | 12 | 0 | 0 | 0 |
| m3 | 12 | 12 | 0 | 0 | 0 |
| m4 | 12 | 12 | 0 | 0 | 0 |
| m5 | 16 | 16 | 0 | 0 | 0 |
| m6 | 16 | 16 | 0 | 0 | 0 |

图 7 实验结果

参考文献:

- [1] WANG ZHENG, LI JIANWEN, ZHAO YONGXIN, *et al.* SPARDL: a requirement modeling language for periodic control system [C]// ISOIA'10: Proceedings of the 5th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation. Berlin: Springer, 2010: 594 - 608.
- [2] CIMATTI A, CLARKE E, GIUNCHIGLIA F, *et al.* NUSMV: a new symbolic model checker[J]. International Journal on Software Tools for Technology Transfer, 2000, 2(4): 2000.
- [3] CLARKE E M, JR GRUMBERG O, PELED D A. Model checking [M]. Cambridge, Massachusetts: MIT Press, 1999.
- [4] HOLZMANN G J. The model checker SPIN[J]. IEEE Transactions on Software Engineering, 1997, 23(5): 279 - 295.
- [5] BENGTSSON J, LARSEN K G, LARSSON F, *et al.* UPPAAL in 1995[C]// Proceedings of the Second International Workshop on Tools and Algorithms for Construction and Analysis of Systems. London: Springer-Verlag, 1996: 431 - 434.
- [6] BALL T, RAJAMANI S K. The SLAM project: debugging system software via static analysis[C]// Proceedings of the 29th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. New York: ACM, 2002: 1 - 3.
- [7] BEYER D, HENZINGER T A, JHALA R, *et al.* The software model checker blast: applications to software engineering[J]. International Journal on Software Tools for Technology Transfer, 2007, 9(2): 505 - 525.
- [8] CLARKE E, KROENING D, FLAVIO L. A tool for checking ANSI-C programs[C]// TACAS'04: Tools and Algorithms for the Construction and Analysis of Systems. London: Springer-Verlag, 2004: 168 - 176.
- [9] HENZINGER T A, HOROWITZ B, KIRSCH C M. Giotto: a time-triggered language for embedded programming[C]// EMSOFT'01: Proceedings of the 1st International Workshop on Embedded Software. London: Springer-Verlag, 2001: 166 - 184.
- [10] ABRIAL J R, BUTLER M J, HALLERSTEDE S, *et al.* Rodin: an open toolset for modelling and reasoning in Event-B[J]. International Journal on Software Tools for Technology Transfer, 2010, 12(6): 447 - 466.
- [11] ABRIAL J R. The B-book: assigning programs to meanings[M]. New York: Cambridge University Press, 1996.
- [12] ABRIAL J R. Modeling in Event-B: system and software engineering[M]. New York: Cambridge University Press, 2010.