

# 一种基于适配器的构件组合方法<sup>①</sup>

姜 宇, 苏中滨, 黄 芳

(东北农业大学 工程学院, 哈尔滨 150030)

**摘 要:** 基于构件的复用技术被认为是提高软件开发效率和服务质量的有效途径, 已在不同应用领域中得到了广泛应用。在面向服务的架构中, 如何利用已有粒度较小的构件构造新的大粒度构件成为当前软件复用领域中的一个研究热点。针对现有面向接口的构件组合方法的不足, 从可复用构件的获取标准与组合方法两个方面进行分析介绍, 并在此基础上提供了一种通用的基于适配器技术的构件组合方法。

**关键词:** 构件组合; 适配器; 软件复用

## Method Adapter-Based for Components Composition

JIANG Yu, SU Zhong-Bin, HUANG Fang

(Engineering College, Northeast Agricultural University, Harbin 150030, China)

**Abstract:** Component-based reuse technology is considered to improve software development efficiency and quality of service and effective way to have on different application areas have been widely used. In service-oriented architecture, how to use a smaller component size has been to construct a new large size as the current software component reuse in the field of a hotspot. Component-oriented interface for the existing deficiency combination of methods, this article from reusable components and combination of standard methods for analysis of both presentation and on this basis provides a universal adapter technology based on combination of components Methods.

**Keywords:** component composition; adapter; software reuse

### 1 引言

伴随计算机硬件和网络技术的不断发展, 在分布式计算系统框架下许多新技术被不断推出。构件(Component)技术因为能够有效增强服务的模块化, 缩短软件系统开发周期, 节约开发成本, 易于维护等突出特点, 以受到软件产业的广泛关注。由此触发了基于构件的软件开发(Component-Based Software Development, CBSD)技术在不同应用领域(多媒体、移动计算等)得到广泛应用。CBSD 不同于传统的软件开发方法, 它强调可复用构件的获取与组合<sup>[1]</sup>。如何生产和组合可复用构件是 CBSD 的重要研究内容。

当前, 随着构件产业的蓬勃发展, 出现了许多不同类型、不同层次的构件产品, 例如: COM+, EJB 和 CORBA 等。开发人员在系统开发的过程中, 经常面临构件因自身能力不足而难以完成任务需求, 在软

件的开发效率和复用性上达不到满意的效果, 从而产生了聚合构件功能来完成任务需求的愿望。针对这一问题的解决方案是扩充构件的服务能力。目前, 国内外针对构件组合方面的研究已取得阶段性成果, 这些研究工作在极大程度上推动了 CBSD 的发展, 为解决单一构件难以完成的复杂任务提供了有利支持。

由于构件产品缺少共同遵守的契约和标准, 特别是在接口组合问题(语义混乱, 接口冗余)上, 缺少一种有效的通用组合方法这样。构件所提供的服务质量要么被忽略要么难以实现, 要么实现方法与应用程序紧密耦合, 破坏了软件的重构和构件的复用性。针对上述问题, 本文提出了一种在源构件的基础上, 开发聚合构件组合完成任务需求的通用性方法。借助适配器的概念对源构件接口进行封装和扩展, 通过合理分配和共享任务、资源和中间结果等信息, 提高聚合构件

① 收稿时间:2010-08-03;收到修改稿时间:2010-09-12

自身能力,形成用户满意的任务解决方案,并给出构件组合的通用方法描述。

## 2 构件的标准化

### 2.1 构件的描述

目前,各构件提供商和研发领域尚未达成统一标准,CBSD领域对构件的定义尚未统一,本文引用文献[2]对构件的定义,即构件是一个数据单元或一个计算单元,由构件接口集合和构件实现模块组成。由于构件对系统环境具有不可预知性,针对不同的任务需求,构件的标准化方法也各不相同,这些因素都为构造聚合构件工作造成了困难。因此,为保证构件组合的正确性和一致性。本文对构件的形式化描述如下:

构件(Component)::={构件名称,结构描述,服务功能集,接口描述,状态集}

构件名称(Component Name)描述构件的名称,是区别于其他构件的唯一标志。

结构描述(Architecture Description)::={简单结构|复合结构}

简单结构(Simple Architecture)描述构件的单一职责,具有简单结构的构件被称为源构件。复合结构(Compound Architecture)描述具有多职责的构件,能为系统提供更多的功能,具有复合结构的构件被称为聚合构件。聚合构件一般由源构件组合构成,在CBSD中,应用程序是粒度最大的聚合构件。

服务功能集(Service Function)::={原子操作|复合操作}

原子操作(Atom Manipulate)是构件在接到服务请求时的一个动作。复合操作(Complex Manipulate)是构件在接到服务请求时的一系列动作。系统中建立了一条服务链,提供从第一个操作到最后一个操作的路径,并且按序执行该路径上的所有服务,最后返回一个终止状态。当接到一个服务请求时,服务功能集只提供一种操作方式。当所有操作条件都为真,则执行服务链上的服务,否则等待操作条件改变为真时,再执行服务链中是服务。服务链中条件表达式的初始状态缺省值为真,表示无条件执行服务链中的操作。

接口描述(Interface Description)::={内部接口,外部接口}

接口是对服务中的参数个数,参数类型,参数名称等信息的抽象描述,每个接口都与一个动作存在映

射关系。内部接口(Internal Interface)描述执行服务需要的内部条件。外部接口(External Interface)描述构件对外提供的功能描述<sup>[3]</sup>。

状态集(State Set)::={初始状态,中间状态,终止状态}

初始状态(Initial State),中间状态(Middle State)和终止状态(End State)共同构成状态集,彼此不存在重复状态。当构件获得初始状态时,构件开始执行服务动作,当服务动作结束时,返回一个终止状态<sup>[4]</sup>。

### 2.2 构件的规范化

在设计和选择构件时,应从系统需求出发,组合方案中的源构件应具有以下条件,以便为聚合构件提供有利支持。

(1) 单一职责。在任务请求中把所需构件的职责逐层分离,由单一职责的源构件完成功能实现。这样不但可以避免这些职责因为耦合在一起,而降低构件的移植性和复用性,而且面对任务需求改变时,系统仍能保持功能的稳定。因此,构件具有单一职责在构件组合和软件复用方面是很有必要的。

(2) 接口规范化。构件组合是在面向接口连接的基础上实现的。接口是构件与外部环境交互的唯一通道,在设计和选择构件时,应尽量满足接口定义的一致性。一个接口定义了一个契约,任何实现该接口的类或结构必须遵守这个契约。在聚合构件的构造过程中,构件内部的源代码往往是不可见的,可能同时具有不同的开发语言,支持不同的开发平台,分布在不同的网络站点,各构件之间只能通过接口进行通信,所以接口规范化在构件组合的过程中是极其重要的<sup>[5]</sup>。

(3) 平台支持。运行平台仅支持某种特殊编程语言所开发的构件,不同厂商的构件产品很难相互组合。应尽量适应开发环境和用户需求的变化,目前的解决方法是基于体系结构<sup>[6]</sup>的平台支持。

## 3 构件的组合

在CBSD中,可复用构件组合方法普遍采用“面向实现”的方式,这种方法是在构件中添加增值服务接口,在另一些构件中引用这一接口的实现。这种组合方式存在明显的不足,不但破坏了构件结构的稳定性和重用性,而且对构件的使用仅限于单个的、孤立的调用,缺乏把多个构件组合起来的通用性方法。如图1所示,畜牧疫病信息管理系统(ADIMS)主要有

应用界面 (AppInterface)、数据库访问 (DBAccess)、日志管理 (LogManager) 三个大粒度的构件在适配器模式下组合构成。本文通过 ADIMS 组合实例来分析和讨论适配器在构件组合中的应用。

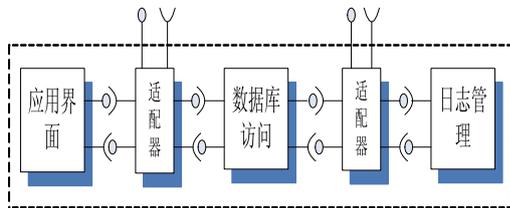


图 1 基于适配器的 ADIMS 组合结构图

### 3.1 适配器的接口组合

本文引用适配器 (Adapter) 的概念来解决构件组合中的接口匹配问题, 它能很好的解决接口信息 (类型、参数、签名、返回值等) 不匹配而导致的构件难以组合的问题。因为源构件中的服务是独立实现的, 组合在一起的前一个服务的接口的信息并不一定与下一个服务的接口匹配, 所以需要在适配器中将数据从前一个服务的接口中得到, 并转化成下一个服务接口识别的形式<sup>[7]</sup>。以 ADIMS 中用户界面构件与数据库访问构件的组合为例, 如图 2 所示:

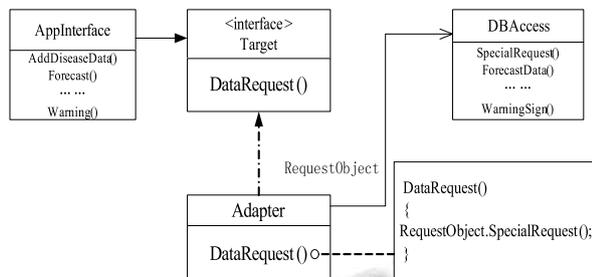


图 2 基于适配器的构件连接 UML 图

Target 是面向适配器编程的抽象接口, Adapter 继承它并提供具体实现。DBAccess 构件中的 SpecialRequest 等是需要适配的方法, 也是 AppInterface 构件请求的目标服务。在 Adapter 内部包装一个 RequestObject 对象, 通过本文给出的接口转化方法将被调用服务的接口转化成目标接口。在服务调用的过程中, 表面上使用的是 Adapter 中的 DataRequest 方法, 但实际上使用的是 RequestObject 对象的 SpecialRequest 方法; 其他方法的调用方式也基本相

同。在数据交互的过程中, 在适配器中可以对接接口信息做一些调整以满足服务调用要求。适配器作为构件连接的重要组成部分同时具备接口转换的重要职责。通过在适配器中修改或插入附加的数据来构造粒度更大的聚合构件来完成需求。适配器的这些显著特点为构件组合方法中的接口匹配问题提供了很好的解决方案。

### 3.2 构件的分层组合

系统开发人员首先依据任务和实现环境的要求, 在体系结构<sup>[8]</sup>思想的指导下, 自顶向下定义各层构件的组合格约, 然后由底层具有单一职责的源构件提供具体实现, 这里主要依据编程人员的经验来获得。在分解业务流程及确定构件组合格约时, 应着重关注大粒度构件中的功能分离, 尽量解除服务之间的耦合, 分析构件中服务的执行顺序和数据流情况, 借助适配器技术将分散的服务单元组织在一起, 通过服务上的交互协作来完成复杂的业务流程。聚合构件的服务集合是源构件功能行为的集合, 但同时受到适配器的协调和制约。经过对复杂的 ADIMS 业务流程的研究和分析, ADIMS 的分层结构树状图如图 3 所示:

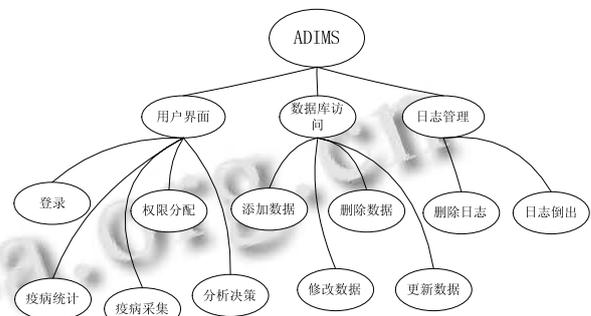


图 3 构件功能分层结构树

构件组合的实现是通过绑定需求的分层结构来分解复杂的业务流程。从逻辑上看, 各层实体间是相互独立的, 层与层之间关注系统的不同方面。ADIMS 由三个大粒度的构件 AppInterface、DBAccess 和 LogManager 组成, 这些大粒度构件分别由单独的小粒度源构件组合实现。这种对系统逐层定义和实现的方式是通过将相同或类似成分的服务进行归纳, 并遵循一致的抽象数据格式和语义, 对构件的多方面特征 (属性、方法、事件、状态) 进行抽象和提炼, 借助适配器来隐藏构件间的交互细节, 最终实现构件间的服务

绑定。

## 4 构件的组合方法

### 4.1 构件接口的类型转换

系统创建了一个接口类型转换器(Conversion Manager)用来完成接口数据类型之间转换的管理。类型转换器提供预定义和自定义两种类型的转换。对预定义类型转换如图 4 所示。适配器继承并实现 ITypeConverter 接口, 此时应在代码中进行显式转换, 告诉编译器你知道这会有丢失数据的危险, 因此编写代码时要把这种可能性考虑在内。当源构件接口数据类型转换失败, 系统会抛出异常, 根据提示采用自定义的转换方式实现接口数据类型的转换。系统允许自定义不同构件接口或类之间的数据类型转换。可以继承 ITypeConverter 接口中的 ChangeType 方法, 手动将原始类型转化为目标类型, 进而实现自定义类型的数据转换。同时构造一个继承类的实例, 将这种自定义的类型转换方法添加到预定义类型转化方法列表中, 这种方式能很好解决构件接口中类型转换问题。

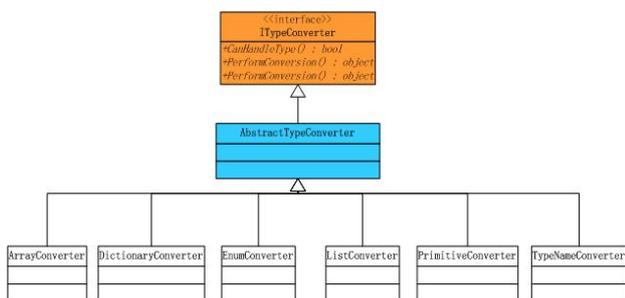


图 4 预定义的接口类型转化

### 4.2 构件组合方法的推导

本文对常见的构件组合方式进行分析 and 总结, 借助文献[9]中提出的 6 种构件组合机制来构造大粒度的聚合构件。任一组合方案都被描述成一个 key/value 键值对, 系统通过迟绑定的方式读取目标方案, 在源构件的接口处实现服务组合, 最后从聚合构件的外部接口输出组合结果标志。方法的操作步骤如下。

输入: 构件库(源构件集合), 初始状态, 适配器, 预定义组合方案。

输出: 组合构件的终止状态或者组合失败的标志。

Step1. 从构件库中选择源构件, 建立备选构件集 RCS, 并对所有 RCS 中的源构件进行分级排序(level1,

level2, ..., leveln) 和同级排序(sort1, sort2, ..., sortn)。

Step2. 读取 XML 格式的预定义组合方案。

Step3. 从 RCS 中提取 level1 下的 sort1 和 sort2 目标源构件, 并解析它们的接口信息, 包括: 参数个数, 参数类型, 可见性等, 初始化目标源构件的状态。

Step4. 建立适配器 Adapter1。通过已有方法对目标接口类型进行必要的转换操作。

Step5. 在满足复合操作的构造条件时, 以预定义的组合方案为指导思想, 进行基于接口匹配的服务功能集组合操作<sup>[10]</sup>。否则, 等待操作条件为真时, 再执行服务链中的服务。

Step6. 维护构件的中间状态, 采用隐藏的方式处理冗余状态, 并对聚合构件进行功能扩展。

Step7. 对组合构件的结构重新描述。为组合构件命名, 本文采用目标源构件名称叠加的方式。

Step8. 服务动作结束时返回一个终止状态或组合失败标志。

Step9. 若 RCS 不为空, 重复 Step2 - Step8 的步骤直到 RCS 为空并返回一个终止状态或者组合失败的标志。

首先分析系统中构件与适配器的连接关系, 然后在构件库中查找与相匹配的结构简单的小粒度构件, 得到一个候选构件集合, 根据预定义的组合方案描述, 提取需要的服务, 构造新的服务链, 最后根据系统中控制流和数据流的描述, 结合分层结构树状图和构件组合方法共同实现构件协作的目的。聚合构件通过暴露外部接口的方式把服务传递给请求者。这种基于适配器的构件组合方法描述了聚合功能的组合关系和制约关系。

## 5 结束语

本文结合基于构件的软件开发技术。首先针对可复用构件的标准化和规范化问题进行了分析和总结; 其次借助适配器的概念实现构件内部接口细节的完整封装, 并根据系统需求构造出系统分层结构树状图, 这种分层剥离业务需求的方法使得构件复用过程中的灵活性和鲁棒性得到提高; 最后, 在基于构件的软件开发和适配器技术的指导下, 本文提出一种具有通用性的构件组合方法。随着软件复用和构件技术在深度和广度的不断拓展, 构件组合方法被越来越多的企业和科研机构研究使

(下转第 193 页)

术, 处理 SQL Server 数据表中的数据。LINQ to SQL 是 .NET Framework 3.5 版的一个组件, 提供了用于将关系数据作为对象管理的基础结构。本文中运用 LINQ to SQL 技术的部分 C# 程序代码如下所示:

```
using System.Linq;
//...代码省略
int monthcount = monthList.Count();
for (var i = 0; i < monthcount; i++)
{ //查询用户几个月的通话时间
var CallTime = from n in db.PhoneCallSum
where n.PhoneNumber == phoneNumberDropDown
List.Selectedvalue && n.查询周期==monthList[i] select n;
//...代码省略
}
```

#### 4 结束语

本文针对中国移动网上话费详单页面中的数据难以进行提取处理的问题, 介绍了一种在 Microsoft 的 .NET 3.5 框架下, 使用 ASP.NET、SgmlReader、LINQ 和 XML 等关键技术, 对移动话费详单 HTML 页面进行自动数据提取的方案。该方案具有简单、易行、高效的特点。通过这一方案能够实现对话费详单页面数据信息进行自动搜集的功能, 并让用户能进一步完成对话费详单信息进行统计、计算等开发处理工作,

从而可使用户准确、全面地了解话费详单情况。该方案已经成功应用于浙江移动即将推出的“移动套餐专家”系统中。

#### 参考文献

- 1 Bjorg S, Touch M. Browse MindTouch (frmly deki wiki) Files on SourceForge.net (SgmlReader Download Page) [2010-03-01]. <http://sourceforge.net/projects/dekiwiki/files/SgmlReader, 2010>
- 2 Altova Company. XML Editor, Data Management, UML, and Web Services Tools from Altova [2010-04-01]. <http://www.altova.com/>
- 3 Jackson J, Myllymaki J. Web-Based Data Mining (from IBM developerWorks)[2010-03-15]. <http://www.myllymaki.org/jussi/webdata/dworks2001.pdf>
- 4 Wahlin D. Parse HTML Pages to Extract Data [2010-02-15]. [http://www.ftponline.com/xmlmag/2002\\_12/online/xml\\_wahlin\\_12\\_18\\_02/default\\_pf.aspx](http://www.ftponline.com/xmlmag/2002_12/online/xml_wahlin_12_18_02/default_pf.aspx)
- 5 耿建勇, 鲁士文. 微软 .NET 框架下提取在线 Web 数据的方法. 计算机系统应用, 2004, 13(4): 53-56.
- 6 Wahlin D. 王宝良译. 基于 XML 的 ASP.NET 开发. 北京: 清华大学出版社, 2002.
- 7 龚赤兵. Visual Studio 2008 中的 LINQ 开发技术. 北京: 机械工业出版社, 2008.

(上接第 222 页)

用, 相关研究还在不断深入, 这些都为构建高质量大规模的应用系统提供了强有力的支持。

#### 参考文献

- 1 张驰. 基于接口匹配的构件组装. 计算机应用, 2007, 27(6): 1420-1422.
- 2 张世琨, 张文娟, 常欣, 王立福, 杨英清. 基于软件体系结构的可复用构件制作和组装. 软件学报, 2001, 12(9): 1351-1359.
- 3 孙莹, 陈松乔. 接口连接式构件组装的一种形式化方法. 计算机科学, 2006, 33(7): 253-256.
- 4 任洪敏, 钱乐秋. 构件组装及形式化推导研究. 软件学报, 2003, 14(6): 1066-1074.
- 5 毛莺池, 梁奕, 王志坚. 异构软件构件组装模型设计与实现. 计算机工程, 2005, 31(7): 56-58.
- 6 陈建勋, 张子鹤, 马于涛, 夏学知. 基于软件体系结构的组件集成框架. 计算机工程, 2004, 30(22): 68-71.
- 7 张景山, 廖华明, 侯紫峰, 徐志伟. 普及计算中基于接口语义描述的动态服务组合方法. 计算机研究与发展, 2004, 41(7): 1124-1134.
- 8 孟凡超, 初佃辉, 战德臣. 基于行为的构件组合及存在性检查. 计算机工程, 2009, 35(15): 41-44.
- 9 黄万良, 陈松乔. 基于消息的构件组合运算与构件演化. 小型微型计算机系统, 2007, 28(7): 1216-1220.
- 10 黄靖, 卢炎生, 徐丽萍. 适时构件合成的语义研究. 计算机科学, 2007, 34(2): 10-16.