维普资讯 http://www.cqvip.com

一种分布式信息系统的构件模型及其应用研究*)

胡金柱　王　锐　张昭理

(华中师范大学计算机系　武汉430079)　(软件工程国家重点实验室　武汉430072)

The Application and Research of CORBA Components in Distributed Information System

HU Jin-Zhu　WANG Rui　ZHAN Zhao-Li

(Department of Computer Science, Central China Normal University, Wuhan, 430079, China)

(State Key Laboratory of Software Engineering, Wuhan 430072, China)

Abstract　With the extension of enterprise business and the increasing requirement in distributed application, the implementation of information system becomes more difficult which may lead to larger failing possibility of project. How to develop an effective and strong system and how to enhance its reliability and maintainability are questions for discussion. As a result, the component-based software development approach is proposed. In this paper, we introduce one of those architectures and give a developing methods based on Corba components with a realistic example, and discuss the process of developing distributed information system. We raise a new information system-oriented component model and the component design methodology, which are analyzed with formal description.

Keywords　Distributed information system, Component technology, Component model

1. 引言

随着企、事业信息系统分布式应用需求的不断增长，大型分布式信息系统软件的开发难度在不断提高，项目失败的可能性也相应地增加。大型分布式信息软件系统的开发经常需要利用已有的成果，集成不同系统的标准软件包，通过集成来减少不必要的重复工作，提高软件的可用性，加快开发进度。但是传统的"算法＋数据结构"的开发方法主要着眼于从无到有的构造，对软件集成的支持不够，大量的开发时间被投入到底层编程中，这种编程既耗时又是大量的重复劳动，从长远看还会造成软件难于维护等一系列的问题。如何高效率地开发一个可靠性高和可维护性好的分布式信息系统，一直是软件开发者面临的一个新课题。

基于构件的软件开发技术[1](Component-Based Software Development Technology, CBSDT)改变了传统的"算法＋数据结构"的开发模式，成为"构件开发＋基于构件的组装"的开发模式。或者说，基于构件的软件开发技术使得软件开发从代码开发过程，转移到对已测试通过的、或者是已经成功使用的、并且互操作性好的一系列构件的集成过程。基于构件的软件开发技术可以提高软件系统的可靠性、灵活性和可维护性，可以缩短软件开发的周期、减少软件系统的维护费用以及快速地集成应用系统。它是目前用于提高大型分布式信息系统软件开发效率的最有效的技术。

2. 一种分布式信息系统的构件模型

分布式信息系统中的构件，大部分是执行应用逻辑的业务构件。一方面它是一种复杂的对象，具有对象所普遍具有的客观属性；另一方面它又是系统宏观功能的一部分。这就要求提供一种构件宏观逻辑描述来确定构件之间的逻辑联系，指导系统的功能运转[2]。因此基于分布式信息系统的构件模型描述可分为两部分：构件描述和逻辑描述。

2.1　一种分布式信息系统的构件描述

对于分布式信息系统而言，其构件描述主要是描述构件的类别、接口、功能和行为。类别描述(Category Description)定义了构件所属的类别，以便于分类和检索；接口是构件与外界联系的桥梁和纽带，具有统一和规范的形式，用户通过接口获取构件提供的服务，接口描述(Interface Description)则规定了构件向外提供的服务；功能描述(Function Description)表达了构件作为复杂对象所具有的功能服务等静态属性；行为描述(Action Description)是对构件动作的规范和标准的抽象描述，它不仅是一种静态的文本描述，也是一种可解释的动作集合，即描述构件的动态属性。

例如，一个分布式数据库访问构件 DatabaseAccess 是一个 Linux 下的 CORBA 构件，它是领域通用构件。该构件具有3个接口：DBConnection(数据连接接口)、DBExecute(数据操作执行接口)和 DBResultControl(数据结果集控制接口)。其中：

(1) DBConnection 的 getConnection 功能函数负责连接客户指定的数据库，并返回一个 DBConnection 的实例。

(2) DBExecute 接口中有 executeQuery (SQL) 和 DBResultControl 两个功能函数。executeQuery (SQL) 返回 DBResultControl 接口的实例，ApplyUpdate (SQL) 则是根据 SQL 语句对数据库进行更新。

(3) DBResultControl 接口对返回的记录进行控制，如定位、移动记录，获取指定行或列的数据等。

2.2　一种分布式信息系统的逻辑描述

逻辑描述是对某个构件集合完成的逻辑功能进行描述，如果说构件描述侧重于构件"是什么"，则逻辑描述侧重于构件"做什么"和"怎么做"。逻辑描述可以定义为一个五元组：

$$LogicDescription=(C_a, E, P, C_t, \beta)$$

其中：C_a 是完成一定业务功能的构件的集合，E 是构件产生的事件的集合，P 是构件动作集合，C_t 是构件控制集合，由业务规则集和控制集组成，β 则是一种映射关系：

$$\beta: E \rightarrow p(p \in P)$$

构件控制是控制构件的动作，即对一个构件的引用和消息监听。其形式定义如下：

*)本文受软件工程国家重点实验室开放基金(SKLSE04-18)和湖北省重点科技攻关项目的支持(2001AA101C31)。

维普资讯 http://www.cqvip.com

构件动作 P=::〈控制语句〉

控制语句::=〈构件名〉〈构件入口〉

控制语句是控制构件动作的基本单位。用逻辑描述五元组对应用逻辑进行抽象，使构件、构件接口和系统动作对应用逻辑唯一标识，在构件控制集中抽象描述了业务处理的规则，将每一个动作产生的事件序列映射成业务处理的流程动作，从而实现应用逻辑功能。例如一个带权限的查询模块，由三个构件协作完成，它们分别为：Identification（身份验证构件），InfoQry（信息查询构件），DbAccess（数据访问构件）。Infoqry 首先调用 Identification 构件的接口方法 SetUserType 鉴别用户的身份，设定查询的类型，然后调用自身的 GetSqlStr 方法根据用户身份和查询规则，转换为查询的 Sql 语句，再把 Sql 语句交给 DbAccess 的 ExecuteSql 方法实施具体的查询。

面向大型分布式信息系统的构件模型，其基本思想就是要用系统宏观的逻辑描述来连接实现系统具体功能的构件，从而组成完整的构件系统，改善构件模型的抽象性和逻辑可扩充性之间的特征冲突问题。因此，关于分布式信息系统的构件模型描述，包括构件描述和逻辑描述具有一定的实用性。

3. 一种分布式信息系统的软件体系结构

在数据库服务器和客户端之间增加中间层应用服务器，把企业的业务逻辑以构件的形态封装起来，再把这些能执行业务逻辑的构件“包装”到应用服务器分发到企业的内部网络中，由这些构件相互协作完成企业的业务处理；如果需要提供 Internet/Intranet 服务则可以通过 Web 服务器直接调用应用服务器中的构件来实现。这就是一种典型的 C/S 计算模式和 Web 计算模式相结合的多层分布式信息系统计算模式。这种模式与传统的 C/S 模式和 B/S 模式相比，具有更好的伸缩性、柔韧性和健壮性，因此它在大型企业分布式信息系统开发中得到了广泛的应用。

图1给出了一种典型的分布式信息系统软件体系结构的多层模型。该模型由客户表示层、Web 服务层、业务逻辑层和数据服务层等四层结构组成。其中：

客户表示层的作用是向用户提供应用接口，它可以是图形界面的应用程序，也可以是 Web 浏览器。但它不需要完成任何重要的业务逻辑，也不直接和数据库交互，同时也不保存本地的状态信息，是一个真正意义上的“瘦”客户机。

Web 服务层的功能是在多层结构中起着代理和缓存作用，用来存储应用需要的 Java Applet 程序和静态的 HTML 网页。另外，当用户向 Web 服务器请求处理企业业务时，Web 服务器代理访问其它服务对象实现业务处理功能。

业务逻辑层是多层结构中最重要的一层，所有的业务逻辑和业务处理都由这一层提供，对数据库的操作也由这一层来完成，它主要由满足企业业务需要的分布式构件组成。

数据服务层主要负责数据存取服务，它还可以包括以前遗留系统的数据库服务器或文件服务器。

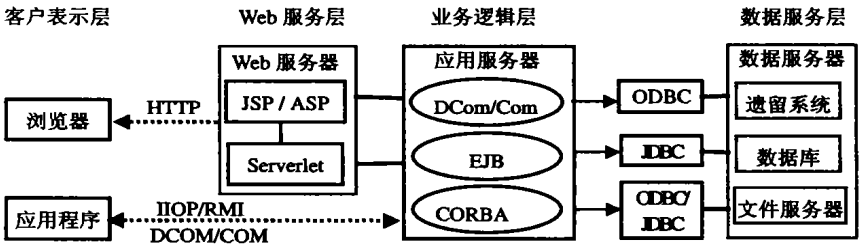

图1

4. CBSDT 在一个法院分布式信息系统中的应用

4.1 法院分布式信息系统中 CORBA 构件模型的实用性

CORBA 提供了独立于编程语言的接口定义语言（IDL）来统一描述构件的接口与方法，通过具体的 IDL 编译器映射为不同程序语言编写的构件对象，构件只要遵循 CORBA 规范即可实现相互调用。CORBA 还提供了各种丰富的系统服务，如命名服务、事件服务、事务处理服务、安全服务等等，开发者可以将原始构件与 CORBA 服务的任意组合相混合产生所需的功能。另外，CORBA 的核心 ORB 还为构件提供了在异构平台和网络环境中互操作的对象总线。由此可见，采用企业级的 CORBA 构件模型开发一个大型分布式信息系统，具有较强的实用性。

例如，在一个人民法院分布式信息系统的诉讼信息管理系统中，其查询子系统涉及到许多类型的信息查询，如立案人员根据当事人信息查询案件审理情况，各业务庭庭长或院长查询超审限案件情况和案件判决情况，各级法院之间通过 Internet/Intranet 互相进行案件业务的查询统计等等。在设计中我们考虑到法院有大量此类分布式信息的处理，另外，法院分布式信息系统对整个系统的安全性、可靠性以及可维护性的要求较高，因此我们采用了 CORBA 构件模型。

4.2 法院分布式信息系统中的 CORBA 构件设计

法院分布式信息系统中的 CORBA 构件大致可以分为三大类：(1)利用 CORBA 的基础结构提供的对象总线（ORB）使构件可以跨地址空间、操作系统和网络实现互操作的“互操作构件”；(2)在这个基础结构的基础上利用附加的系统级服务定制满足特殊需要的“服务构件”；(3)在某些应用领域中创建能够模拟真实世界业务处理的“业务构件”。基于 CORBA 构件的法院分布式信息系统的整体结构如图2所示。

用于查询子系统的构件可分为四类：表现构件、身份验证构件、查询业务构件和数据访问构件。其中表现构件属于系统构件范畴主要包括窗体、按钮、列表框等，它组成用户的图形界面，并调用 CORBA 服务器中的各分布式构件进行业务处理。为保证系统的安全，在查询前先进行身份验证，验证通过后设定用户的查询类型。如院长可以查询所有类型案件的信息，而刑事庭人员只能查询刑事案件的情况。查询业务构件是实现查询业务的专用构件，主要完成查询规则的转换，把查询的内容以 SQL 语句的形式交给数据访问构件，由数据访问构件实施查询并把结果反馈给表现构件。数据访问构件属于有较大复用价值的通用构件，最好独立划分，以保持各构件间的独立性，提高可重用性。

4.3 法院分布式信息系统中 CORBA 构件的开发过程

用于法院分布式信息系统中 CORBA 构件的开发过程大

维普资讯 http://www.cqvip.com

致描述如下：

(1) 采用接口定义语言 IDL 为每一构件对象编写规格说明，指明对象调用接口和提供的操作。

(2) 使用 IDL 编译器生成客户的桩代码(Stub)和服务器的框架(Skeleton)对象代码。

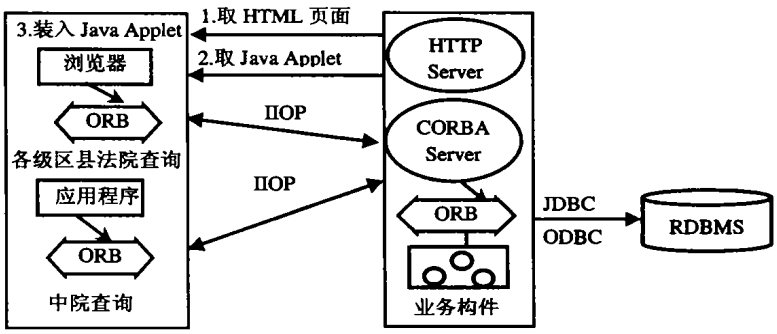

图2

(3) 编写客户端构件对象代码。

(4) 编写服务器构件对象代码。

(5) 用具体语言编译器编译产生客户构件和服务构件程序代码。

例如，为查询业务构件定义的 IDL 文件部分内容如下：

```
module LawInfoQry{
    interface CaseInfoQry
        {
            string GetSqlStr (in string UserType, in string
            QryCondition);
        };
};
```

首先定义 LawInfoQry 模块，其中包含了接口 CaseInfoQry，用于执行案件信息查询，接口中定义了方法 GetSqlStr，它根据输入的用户类型和查询条件转换为相应的 Sql 语句，交由数据访问构件实施查询。使用 VisiBroker 的 IDL 编译器编译该 IDL 文件，输入命令 Prompt＞idl2java LawInfoQry 得到客户端的桩和服务器框架代码。这里，采用了 Visibroke for Java 开发 CORBA 构件，C++Builder 开发图形用户界面程序。由于 Java 规定每个文件只能有一个公有的接口或类，编译 IDL 文件时会生成几个. Java 文件。这些文件存储在一个新建的子目录 LawInfoQry 中，为生成文件所属的包。生成的文件包括：

CaseInfoQryStub. java—CaseInfoQry 构件对象在客户端的桩代码。

CaseInfoQry. java —CaseInfoQry 接口声明。

CaseInfoQryHelper. java —声明 CaseInfoQryHelper 类，该类为 CaseInfoQry 接口定义各种支持功能方法。

CaseInfoQryHolder. java —声明 CaseInfoQryHolder 类，该类为传递参数提供支持。

CaseInfoQryOperation. java —定义 CaseInfoQry 接口的方法，同时在客户端和服务器端使用。

CaseInfoQryPOA. java —包含服务端 CaseInfoQry 构件对象的框架代码。

客户端代码—Client. java，服务器端代码—Server. java 和 CaseInfoQry 类及方法 GetSqlStr 的实现代码—CaseInfoQryImpl. java。最后用 Java 编译器编译客户和服务器代码，产生所需的 CORBA 构件。在运行 VisiBroker 客户程序或服务程序之前，必须在网络中至少有一台主机启动智能代理(Smart Agent)。智能代理是一种动态分布式目录服务，客户程序调用构件方法时会自动查询智能代理，由智能代理指定具体服务的位置，从而建立客户程序和服务程序之间的连接。智能代理与客户之间的通信完全透明，它还具有一定的负载平衡和容错功能。对于各法院之间基于 Internet/Intranet 的查询是从 Web 服务器下载包含构件方法存根的 Java applet，由 applet 通过 IIOP 协议直接调用业务构件实现。由于 CORBA 技术支持构件对象的位置透明性、实现透明性、执行状态透明性和通讯机制透明性，这样可以让开发者专注于构件业务逻辑的实现，而不用过多考虑复杂的底层，极大地提高了构件开发的效率。在中院信息系统实施成功后，把得到的系统模型和已经实现的业务构件分别保留下来，在基层法院开发时，可在已有构件基础上根据基层法院的业务需求修改部分业务构件即可快速组装新的应用系统。

结束语 随着企业(事业)信息系统分布式应用需求的不断增长，对开发方法提出了新的要求。基于 CORBA 的构件模型结合了构件技术和多级分布式计算模式的优点，为开发一个大型分布式信息系统提供了解决方案。本文结合我们采用 CORBA 构件模型开发法院分布式综合信息系统的实践，讨论了一种构件模型的形式化描述以及 CORBA 构件开发过程。

20世纪90年代以来，构件技术在软件开发中得到了日益广泛的应用，虽然在具体的应用中还有许多问题需要做进一步深入的研究，但它正在逐步形成一套完备的研究体系。目前构件技术及其应用的研究涉及到软件工程、操作系统、网络、数据库等多个计算机学科领域。构件技术必将渗透到各个应用领域的软件系统中，将成为软件开发的重要技术之一。

参 考 文 献

1 李刚，金茂忠. 一种可重用的构件设计方法. 计算机研究与发展，2000(5)：599～615

2 徐征，陈雪飞，刘晓铭，等. 一种构件化的动态软件系统模型. 小型微型计算机系统，1999(2)：98～101

3 刘晓铭，刘积仁，李天华. 构件化领域框架设计与实现. 计算机研究与发展，1999(2)：166～169

4 张晓东，柴跃廷. 一种形式化的构件模型框架. 清华大学学报(自然科学版)，2000(3)：64～67

5 Kim S-G. Designing a Domain Framework with Component Management Model. Journal of Software，2002，13(3)：335～341

6 Yang Fu-qing，Mei Hong，Li Ke-qin. Software reuse and software component technology. Acta Electronica Sinica，1999，27(2)：69～75

7 Nierstarasz. Component-Oriented Software Development. ACM，35(9)：160～165

8 Bellinzona R，Fugini M G. Reuse of Specification and Designs in a Development Information System. http：// www. resear-chindex. org