

基于 CCM 的构件组装技术的研究与应用

刘丽君, 钟军凯, 曹宝香

(曲阜师范大学 计算机科学学院, 山东 日照 276826)

摘要:随着计算机网络技术的应用发展, 分布式构件技术成为分布式计算领域的热点, CCM 就是主流的分布式构件技术之一。分析了 CORBA 构件模型(CCM)的特点: 即通过端口定义构件对外提供的功能、通过端口定义构件所需要的外部功能, 构件通过端口进行组装。在产品生命周期管理(PLM)中的供应链管理系统(SCM)中充分应用了 CCM 及其端口组装方法, 实现在分布式环境下软件构件的即插即用功能, 使 SCM 系统具有良好的可移植性和平台无关性。

关键词:构件; CORBA 构件模型; 构件组装; 供应链管理系统

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2008)07-0215-03

Research of Component Composition Technology Based on CCM and Its Application

LIU Li-jun, ZHONG Jun-kai, CAO Bao-xiang

(College of Computer Science of Qufu Normal University, Rizhao 276826, China)

Abstract: Along with the development of computer networks and applications, distributed component technology became the hotspot of distributed computing. CCM is one of the popular distributed component technologies. Analyzes CORBA component model (CCM) which defines not only the services it provides but also the services it required. So CORBA components can be assembled via their ports. It applies the mechanism of the component assembly based on CCM in the supply chain management of the PLM. This new kind of connective component realizes the distributed component's plus and play, so it builds the SCM which has platform-independence and easy to re-plant.

Key words: component; CORBA component model; component assembly; SCM

0 引言

CORBA 构件模型 (CCM, CORBA Component Model) 是 OMG 在 CORBA 3.0 中的显著特征, 是支持构件和系统集成的主要贡献。CCM 标准化了利用 CORBA 作为中间件的开发, 通过定义一组特性和服务, CCM 构件扩展和包容 CORBA 对象模型。CCM 构件技术使开发者只需关注业务逻辑的实现, 而且具有高度的可移植性和可重用性。

构件组装技术是基于构件的软件开发的研究重点, 目前构件组装技术有多种, 如基于框架的构件组装方法和基于灰盒编程的连接器方法等。文中所研究的是基于构件模型的组装机制, 构件模型指明构件接口

的结构, 构件与环境之间的交互机制, 构件内部的结构等。CCM 在面向对象的基础上将面向对象进行了扩充, 把构件技术与面向对象技术融合。文中将对 CCM 和构件组装技术进行研究, 分析基于 CCM 的组装机制, 同时在产品生命周期管理(PLM)的供应链系统中进行应用。

1 CCM 构件定义与构件组装技术

1.1 CCM 构件

CCM 构件^[1]可分为: 服务、会话、实体和进程四类。

(1) 服务构件: 在每次调用时实例化, 因此不能在调用之间保持状态。它对应于 EJB 中的无状态会话构件。

(2) 会话构件: 构件实例在事务型会话过程中保留状态信息, 并在一次会话中允许多次调用。会话构件实例跨越会话边界将丢失状态信息, 对应于 EJB 中的有状态会话构件。

收稿日期: 2007-10-29

基金项目: 国家自然科学基金资助项目(60072014); 山东省科技厅项目(2006CC2301001)

作者简介: 刘丽君(1980-), 女, 山东泰安人, 硕士研究生, 研究方向为中间件、网络数据库; 曹宝香, 教授, 研究方向为中间件、网络数据库、CAD。

(3)进程构件:构件实例有持久状态,它们的生命周期与所服务进程的生命周期相同,因此是不确定的。

(4)实体构件:最终拥有持久的实例。该实例与数据库中的实体相对应,通过数据库中的实体关键字来访问对应的实例,对应于 EJB 中的实体构件。

1.2 CCM 构件模型

CCM 抽象构件模型^[2]定义了 CCM 构件外观特征,一个 CCM 构件具有如下特征:

(1)端口:CORBA 构件模型共支持四种基本的端口(见图 1)。

a. 刻面(Facet):是构件向外提供的接口,构件相互独立的命名接口。CCM 中的一个特殊的刻面是“等价”接口,该接口提供在 CCM 构件不同的刻面之间导航的功能。

b. 插接口(Receptacle):是构件的需求接口。一个构件实例的插接口可以和另一个构件的刻面接口相连接。插接口提供了连接和断开操作,并在内部协调对象引用与指代对象类型之间的一致性。

c. 事件源(Event Source)和事件槽(Event Sink):事件源负责发布事件,事件槽用于接收外界向其推入的事件。它们并不直接连接,而是同时连接到事件通道上。

(2)主关键字:该值由实体构件提供,供客户标志该构件。

(3)属性与配置:即一组可读写的命名值,实现对构件的初始配置。它们被描述成 IDL 的属性,可进行 get()和 set()操作。

(4)本地接口:提供了工厂功能来创建新的实例,一个构件可以提供一个或多个本地接口。

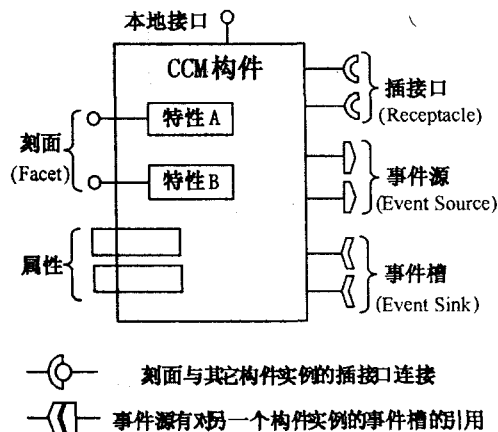


图 1 CCM 构件模型

1.3 基于 CCM 的构件组装技术

(1)在 CCM 构件中使用端口对外交互信息,端口分别代表构件向外提供的功能和构件要求的外部功能。CCM 中构件基于端口进行组装,使构件要求的功

能和所提供的功能进行匹配。构件端口为构件组装提供了两种交互模式:同步模式和异步模式。构件的插接口提供了连接构件需要的接口的机制,构件刻面与插接口两个端口支持同步调用的交互模式。事件源与事件槽这两个端口支持异步通知的交互模式。

(2)在 CCM 中构件提供和请求的服务在构件端口中进行声明,端口是构件服务的契约化规范,构件的组装要满足构件端口的匹配^[3],为进行构件端口匹配的检查定义构件组装配置算法:

```

Configs(i,S,Sv)
// i 是构件计数,S 表示产生的配置,Sv 表示产生的有效配置
for(j=1; j<=size(Ci,P);)
//检查构件 Ci 提供的所有服务
if ({Ci, Pj} ∩ S.p = NULL)
{ S = SU {Ci, Pj};
if(A.p ⊆ S.p) //检查 S 是否构成配置
{Sv = Sv ∪ S;} //若构成,将 S 并入 Sv 中
else
{cofigs(i,S,Sv);} //否则存在服务断层,继续配置
S = S - {Ci, Pj} //存在服务交叠,对交叠服务进行隐藏操作
}
Configs(i+1,S,Sv;) //检查下一个构件

```

2 应用系统设计与实现

2.1 SCM 销售子系统设计

供应链系统(SCM)是 PLM 中的重要部分,SCM 中的销售管理子系统以订单为核心,对企业销售业务的执行过程跟踪和管理,为客户和供应商之间提供了一个平台。销售管理子系统的功能主要由三个构件实现^[4]:客户构件、数据访问构件、供应商构件。

①客户构件定义了两个端口^[5]:一个发布 RequestInfo 事件类型(客户需求某货物订单事件)的“事件源”;另一个是连接 AccessDB 接口(访问数据库接口)的“插接口”。

②数据访问构件定义了一个提供 AccessDB 接口(访问数据库接口)的“刻面”。

③供应商构件定义了两个端口:一个接受 RequestInfo 事件类型(客户需求某货物订单事件)的“事件槽”;另一个是连接 AccessDB 接口(访问数据接口)的“插接口”。

2.2 SCM 销售子系统实现

构件的定义:使用 OMG IDL3 文件 WService idl 定义以上三个构件:

```

Import Components;

```

```

Module WrService
{
  Interface AccessDB//访问数据库接口
  {
    Void ExecuteSQL (in String strsql)
    //执行传入的 SQL 语句
  }
  eventtype RequestInfo//客户需求货物的订单数据信息
  {
    String dwnm();//客户单位名称
    String hwxx();//需求货物型号
    String hwnm();//需求货物名称
    String num();//需求货物数量
    String time();//交货时间
    String address();//交货地点
  };
  Component Client//客户构件
  {
    Uses AccessDB Idb;
    //声明该构件要求使用 AccessDB 类型的接口
    Publishes RequestInfo info;
    //声明该构件发布 RequestInfo 类型的事件
  }
  home ClientHome manages Client{};
  //声明管理客户构件的宿主
  Component DBServe//数据库访问构件
  {
    Provide AccessDB Idb;
    //声明构件提供一个 AccessDB 类型的接口
  }
  home DBServe Home manages DBServe{};
  //定义管理数据库访问构件的宿主
  Component Supplier//供应商构件
  {
    Receives RequestInfo info;
    //声明该构件接收 RequestInfo 类型的事件
    Uses AccessDB Idb;
    //声明该构件要求使用 AccessDB 类型的接口
  }
  home SupplierHome manages Supplier;
  //定义管理供应商构件的宿主
};

```

构件实现:IDL 3 文件经过 IDL 3 编译器产生了与 IDL 3 文件对应的本地(Local)接口,实现构件也就是在实现文件中这些本地接口。将以数据库访问构件为例说明构件实现方法:

接口 AccessDB 的本地接口映射:

```

Local interface CCM - AccessDB: AccessDB
{
  Void ExecuteSQL(in String strsql)
};

```

在 AccessDBImpl.java 中实现 CCM - AccessDB 接口:

```

Public class AccessDBImpl extends org.omg.CORBA.LocalObject
Implements CCM - AccessDB
{
  Void ExecuteSQL (in String strsql)
  {
    .....//方法实现
  }
};

```

构件 DBserve 映射到本地接口:

```

Local interface CCM - DBServe:: Components:: EnterpriseC-

```

```

omponent
{
  CCM - AccessDB get - Idb ();
};

```

在 DBServeImpl.java 中实现 CCM - DBserve 接口:

```

Public class DBServeImpl extends AccessDBImpl
implements CCM - DBServe
{
  public CCM - AccessDB get - Idb()
  {
    .....//方法实现
  }
};

```

2.3 构件组装

输入客户构件软件包、数据访问构件软件包和供应商构件软件包,利用组装工具按照设计的方案将它们组装起来^[6]。SCM 中构件组装见图 2。

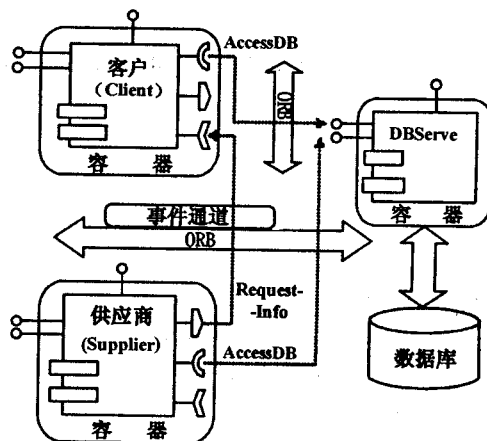


图 2 SCM 中构件的组装

组装工具把供应商构件接收 RequestInfo 类型事件的“事件槽”与客户发布 RequestInfo 类型事件的“事件源”连接,把数据访问构件提供 AccessDB 接口(访问数据库接口)的“刻画”与客户、供应商访问数据库的“插接口”连接。完成上述操作以后,组装工具将输出扩展名为“.arr”的构件集文件。此构件集文件中包含的构件集描述器描述组装信息。构件集描述器包含构件集中用到的所有构件,描述创建构件和构件宿主的实例及部署信息和连接信息。

组装工具应完成以下三个主要任务:

- 1) 载入所需构件包,读取相关描述信息并把用于构件连接的端口信息以图化的方式显示给用户。
- 2) 用户根据系统体系结构与组装工具进行交互。用组装工具提供的描述方法描述构件间的连接关系和构件实例的创建信息。
- 3) 最后组装工具根据与用户的交互信息生成一个构件集描述器,并把相关文件压缩成为一个构件集文件(.aar)作为组装工具的输出。

(下转第 221 页)

网络通信协议支持,处理异构系统中电子政务数据报文传送、管理以及报文转换等过程中各种需求不同的报文。

(2)路由支持:数据交换平台将接受到的文件解包后提取目的地信息,查询路由表,将文件按要求格式重新打包路由到下一级交换平台或与其连接的系统。

3.6 监控中心

监控中心由数据监控、安全支撑、错误处理、日志管理四个部分组成。

(1)数据监控:实现对数据交换流量和数据文件交换状态(创建、激活、挂起、终止等)的控制、管理、查询、统计与审计等。

(2)安全支撑:在利用防火墙、入侵检测、漏洞扫描等网络安全技术的基础上,保障交换平台的网络通讯与传输安全;在提供统一的 CA 身份认证前提下,控制用户的数据请求、访问权限;在提供加密、认证、数字签名等技术的前提下,实现数据的保密性、完整性和不可否认性。

(3)错误处理:使数据交换过程成为一个事务性过程,保证数据交换中数据的准确性和传输过程的完整性,提高通信的效率、健壮性和可靠性。

(4)日志管理:完成交换数据备份、记录系统异常和网络异常信息等。

(上接第 214 页)

参考文献:

- [1] Eirinaki M, Vazirgiannis M. Web Mining for Web Personalization[J]. ACM Transactions on Internet Technology, 2003, 3(1):1-27.
- [2] Baglioni M, Ferrara U, Romei A. Preprocessing and Mining Web Log Data for Web Personalization[J]. Advances in Arti-

(上接第 217 页)

3 结束语

CCM 构件是一个自包含的构件实体。文中对 CORBA 构件模型和基于 CCM 构件组装技术进行了研究,并在 PLM 中供应链系统(SCM)的销售管理子系统中应用。基于构件的开发遵循构件开发-应用组装-应用部署的过程,同时下一步要考虑系统的演化,以及自动组装过程的研究。

参考文献:

- [1] 韩跃科.基于 Java/CORBA 的政务信息发布系统研究[J].

4 结束语

数据交换不仅仅局限于电子政务系统方面,在其它系统的集成过程中也是经常要面临的问题。文中提出的数据交换平台方案通过数据交换拓扑网络,将各个物理位置分散的异构系统连接为一个统一的整体,不仅可以有效地解决电子政务系统中各应用中数据交换与信息共享问题,同时也适用于其他领域。通过这种方案构建的体系结构,无论是新子系统,还是旧的业务系统都有其相对独立性、灵活性,为系统进一步扩展提供良好的扩展性。

参考文献:

- [1] 王 琰,徐 玲.电子政务理论与实务[M].北京:清华大学出版社,2004:2-17.
- [2] 向 真,吴秋云,陈 华.电子政务三网模式下的数据交换[J].计算机工程与科学,2004,26(8):11-13.
- [3] 何国辉.基于 XML 的电子政务系统设计[J].微计算机信息,2006,22(3):151-154.
- [4] 张 繁,蔡家楣.电子政务系统中的数据交换和共享服务平台设计[J].计算机工程与应用,2003,39(7):226-229.
- [5] 梁 娟,熊桂喜,李 静.电子政务中基于 XML 的关系异构交换技术[J].计算机与数学工程,2006,34(12):60-63.

cial Intelligence, 2003(2):237-249.

- [3] 杨小兵.聚类分析中若干关键技术的研究[D].杭州:浙江大学,2005.
- [4] 吕亚兵.WEB 站点日志数据挖掘的研究与实现[D].武汉:武汉理工大学,2006.
- [5] 梁 伟,张慧颖.电子商务推荐系统中推荐模型的研究[J].计算机工程与应用,2006,42(36):183-186.

计算机技术与发展,2007,17(6):203-205.

- [2] Wallnau K C, Carney D. Building systems from commercial components[M]. [s.l.]: Addison Wesley, 2002.
- [3] 张 驰.基于接口匹配的构件组装[J].计算机应用,2007,27(6):1420-1422.
- [4] 李 睿,徐 红,曾应员.构件化技术在学生成绩查询统计系统中的应用[J].计算机技术与发展,2007,17(6):214-216.
- [5] 徐东升,张 驰.CORBA 构件接口扩展技术与描述[J].微电子学与计算机,2007,24(4):12-14.
- [6] 任洪敏,钱乐秋.构件组装及其形式化推导研究[J].软件学报,2003,14(6):1066-1074.