

一种基于 CCM 业务构件的可重构企业信息系统结构^{*}

郭银章 乔钢柱 徐玉斌

(太原重型机械学院系统仿真与计算机应用研究所 太原030024)

摘要 针对制造企业信息系统的软件体系结构和功能应用软件的可重用问题进行研究。基于软件体系结构描述语言 ADL 的 BNF 语法和 OMG/CCM 的构件技术,详细设计了企业信息系统的可重构软件体系结构模型、系统的可重用业务构件和系统的即插即用框架。通过扩充 CCM 构件的消息和通知服务,描述了企业信息系统软件体系结构和功能应用软件的重用。

关键词 CCM 构件,企业信息系统,可重构

An Reused Enterprise Information System Architecture Based on CCM Business Component

GUO Yin-Zhang QIAO Gang-Zhu XU Yu-Bin

(Division of System Simulation and Computer Application, Taiyuan Heavy Machinery Institute, Taiyuan 030024)

Abstract Research is made on the reuse of software architecture and function application software in manufacture enterprise information system. Based on BNF syntax of software architecture definition language ADL and OMG/CCM component technology, this article detailedly designs the reconfigurable software architecture model, the reused business component and plug and play framework of enterprise information system. Then, the reuse of software architecture and function application software on enterprise information system is described by extending CCM component messages and notification service.

Keywords CCM component, Enterprise information system, Reconfigurable

1 引言

为了应对日益激烈的全球化市场竞争,现代企业需要建立企业间的动态联盟,来实现以产品为中心的跨地区协同制造和组织管理。同时,为了应对瞬息变化的全球市场,企业必须随时调整自己的业务规则以适应市场的需求。因此,现代企业信息系统的构建,必须满足企业过程重组对信息系统的重构和保护企业在信息系统方面的投资需求,信息系统的可重构技术已经成为现代企业信息系统构建中满足上述需求的首选技术。信息系统软件的重构技术主要表现在对现有构件进行重用,对现有基于构件的软件体系结构进行重用以及对现有的构件库进行管理三个方面。软件体系结构的重用是指通过连接子的方式将现有的构件库中的构件无缝地连接成一个应用系统,是基于构件和构件库的更大程度复用。在对现有构件进行重用的实现技术和机制方面,主要表现在基于 OO 技术的可重构实现方式和基于分布对象技术的重构方案,这两种方法在系统重构时都需要了解实现细节,属于白盒机制。而系统重构的真正目的是使设计人员致力于领域工程的分析,而并

不关心组成系统的构件的实现细节,即实现黑盒机制的构件即插即用装配与重组策略。为此,我们提出了一种基于 CCM 业务构件的企业可重构信息系统,主要针对可重构企业信息系统的软件体系结构模型、CCM 业务构件系统以及即插即用的系统重构框架进行研究与设计。

2 基于 CCM 构件的可重构信息系统软件体系结构

2.1 软件体系结构模型

软件体系结构是软件更高层次的抽象,主要包括软件系统总体的组织和全局控制、通信协议、数据存储、设计元素特定功能的分配、设计元素的组织、规模和性能等。软件体系结构主要由处理元素,数据元素和连接元素构成,其中处理元素是用来处理数据的元素,数据元素则为处理元素提供数据信息来源,这两者统称为构件。连接元素是整个软件体系结构的核心,主要用于描述软件体系结构中各连接对象间的相互关系,在运行期实际控制各个对象的相互作用。软件体系结构的外在接口称为角色,它是软件体系结构和构件绑定时的唯一接口。软件体系结

^{*}基金项目:本文受山西省青年科学基金项目资助(项目编号20031029)。郭银章 副教授,主要研究方向:分布式网络计算与智能 Agent 技术。徐玉斌 硕士,教授。

构的3C定义模型为:SA = 构件(Component) + 连接器(Connector) + 绑定约束(Constraint)。

基于 CCM 业务构件模型的软件体系结构,定义了组成可重构系统的构件以及构件间的相互作用关系,由于 CCM 业务构件系统具有构件接口描述,

构件管理与组装的功能,因此我们提出了基于构件模型的层次化可重构信息系统体系结构,整个系统分为基础平台和支撑环境层,分布式对象计算平台层,可重用的业务构件层,基于领域的公共业务过程层以及企业可重构系统的应用层。如图1所示。

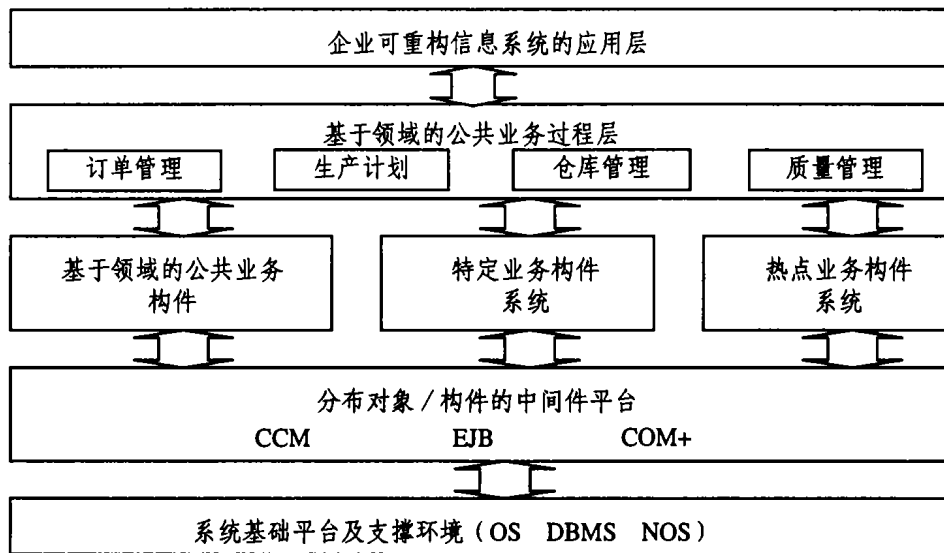


图1 基于 CCM 构件的企业可重构信息系统体系结构

(1)基础平台及支撑环境层:主要为整个系统运行的客户端和服务端提供最基础的环境,主要包括操作系统、数据库管理系统及底层的网络环境。

(2)分布对象网络计算平台层:是企业可重构信息系统分布网络计算的基础支撑体系,是一种中间件的集成框架,它屏蔽了企业可重构信息系统的处理元素和数据元素的业务构件与基础平台以及业务构件之间诸如通讯协议、操作系统、数据异构和构件位置等交互细节,为其上层提供一种透明的互操作机制。

(3)业务构件系统层:软件体系结构是企业可重构信息系统的构造蓝图,业务构件就是可复用的信息系统构造块,我们将为业务领域中绝大多数应用提供公共功能和将公共数据的业务构件划分为公共业务构件群;将为特定业务应用提供服务的业务构件划分为特定业务构件群;将那些经常进行重用的热点构件划分为热点业务构件群。这样针对不同的应用,业务构件层与分布网络计算平台向上层提供集成服务。

(4)公共业务过程层:为特定业务域的应用提供基本功能和结构,它不仅要应用业务构件系统层提供的公共构件,而且还要应用其特定业务构件,为上层提供公共的业务过程和特有的领域专用业务构件。

(5)企业可重构信息系统应用层:应用层主要通过创建 GUI 构件来使用并扩展公共业务过程层提供的公共业务过程,来形成具体的功能应用系统。可

重构信息系统应用层是企业信息系统进行系统应用重构的主要界面层。

2.2 软件体系结构 ADL 描述及实现

基于 CCM 业务构件的软件体系结构,是整个信息系统重构的控制蓝图。它描述了整个信息系统中的外在接口,各业务构件间的相互作用关系以及构件与连接器的绑定。软件体系结构一般采用其描述语言 ADL 进行实现,其实现语法如下:

```

Architecture:: //软件体系结构描述
Architecture architecture name {
Roles component list
Component_instances component_instances_list
Connectors connector_list
[architecture_topology topology]
component_list:: //外部接口业务构件集合
top_most; {component_name} //上端口扩充的业务构件
internal; {component_name} //内置的业务构件
bottom_most; {component_name} //下端口扩充的业务构件
component_instances_list:: //绑定构件集合
instance_name instantiates component_name
[with {parameter_instantiates}];
connector_list:: //连接器
{connector;}
Connector:: =
Connector connector_name {Message_filter message_filter_type;}
message_filter_type:: =
no_filtering | no_filtering | prioritized | mgs_sink
    
```

3 基于 CCM 构件模型的信息系统业务构件设计

3.1 CCM 构件模型

CCM 构件模型是国际对象管理组织 OMG 于 2002年7月在公共对象请求代理体系结构 CORBA3.0中提出的一种构件模型标准。CCM 构件模型

为企业可重构信息系统的业务构件提供了标准化的管理、接口描述和运行机制,CCM 定义了四种模型标准:抽象模型:提供了用于描述构件的接口和特性的方法,即构件实现的定义语言 CIDL。构件实现的框架模型:主要完成构件实现部分的自动化。容器模型:提供构件运行的服务端环境,管理和创建构件的实例,并为构件运行加载服务。构建配置及打包模型:与抽象模型、容器模型相互协作,共同构成完整的分布式企业服务体系结构,完成构件的连接和实现构件的动态集成。

CCM 构件模型是一种语言中性的软件构件模型,CCM 构件是一种新的 CORBA 基本元类型,构件封装了一个设计实体,对外提供一致的标准接口,通过 CIDL 语言描述的构件接口可以映射到具体的实际可执行语言环境(比如 JAVA,C++),供系统重构和集成使用。构件宿主用于为提供管理构件生命周期的工厂服务,CCM 引入一个新的关键词 Home 来支持构件宿主。基于 CCM 构件模型构建的业务构件,可以实现平台的异构性,互操作性,可重构性,对于企业信息系统的重构,提供很好的技术支持。

3.2 企业业务构件集的结构与设计

企业业务构件集是针对企业信息系统的分析模型划分的业务构件集合,它是一组通过协作来发布

系统功能的业务构件群。业务构件的设计要求构件间的耦合度要低,充分降低构件间的交叉关联,以适应其重用和集成。业务构件是信息系统集成重构的基本元素,其必须具有关联的独立性;位置的透明性;支持异构连接性和较强的演化能力,所以业务构件集应采取层次化结构进行组织。其逻辑层次可分为业务过程构件层,业务实体构件层和热点构件层。我们以制造业信息系统中典型的公共业务过程销售订单处理与执行为例介绍业务构件的设计。根据订单管理业务流程,构建订单管理的业务构件系统结构如图2所示。

- 业务过程构件层:代表领域本身的业务活动,如订单管理、发票管理、收发货管理等。
- 业务实体构件层:代表业务过程需要调用的业务数据,存储对象以及相关服务。
- 热点构件层:将易变化的功能分析形成的构件集,是信息可重构时常变化的构件。

当业务构件设计完成之后,采用 CCM 所提供的接口定义语言 CIDL 对其进行封装,对外提供透明的标准接口,由 CIDL 描述的接口是一种中性语言接口,通过 CIDL 编译器生成服务端的构件,并在命名服务中注册以及在相关的接口仓库中添加其相关接口及服务端和客户端的框架,供系统重构调用。

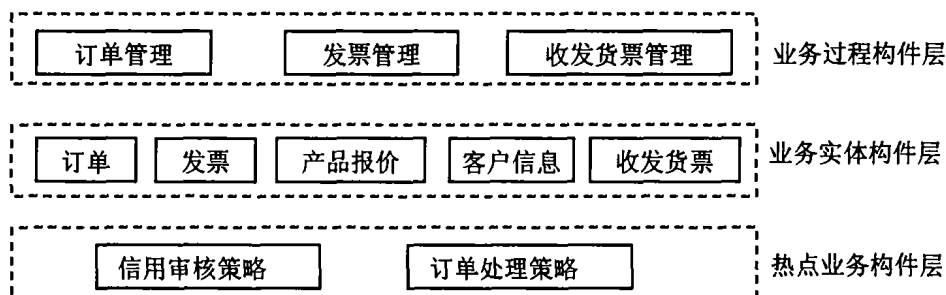


图2 以订单管理系统为例的业务构件集的结构

3.3 CCM 业务构件中消息和通知服务的扩展

CCM 业务构件经 CIDL 描述并实现之后,为了满足整个软件体系结构的外部接口绑定需求,必须对已定义好的业务构件进行自身功能的扩展。其扩展的内容主要是对构件中消息和通知服务的处理,即在已由 CIDL 定义好的构件外层新封装一个层面,该层面主要包括顶端域和下端域以及相应的端口,顶端域是发送请求的集合和接收消息的集合,下端域是接收请求的集合和发送消息的集合。上端域用 TOP_DOMAIN 表示,下端域用 BOTTOM_DOMAIN 表示,其语法结构如下。

```

Component ::=
ADL component_name {
Interface component_message_interface
Parameters component_parameters
Methods component_methods
[behavior component_behavior]
[context component_context]};
    
```

```

component_message_interface ::=
top_domain_interface
bottom_domain_interface
top_domain_interface ::=
top_domain is
out_interface_requests
in_interface_notifications
bottom_domain_interface ::=
bottom_domain is
out_interface_notifications
in_interface_requests
Interface_request ::=
{requests; } | null;
interface_notification ::=
{notifications; } | null;
request ::=
message_name {request_parameters}
request_parameter ::=
[to component_name || parameters_list]
notification ::=
message_name {parameter_list}
component_behavior ::=
startup
cleanup
{internal_state_change | message_transition}
    
```

```

message_transition ::=
received_message
[invoked_methods | generated_message]
received_message ::=
received_message
[notification_sequence | request_sequence]
invoked_methods ::=
invoked_methods
method_name { , method_name };
generated_message ::=
[message_generation_frequency
request_sequence]
message_generation_frequency ::=
always_generate | may_generate
Request_sequence ::=
Request_name { logical_operator
request_name };
Notification_sequence ::=
Notification_name { logice_operator
notification_name };

```

4 基于 CCM 的可重构企业信息系统的框架

框架是一组互相交互的、可重用的、可定制的应用程序和软件构件的集合。基于 CCM 业务构件的，基于 CCM 的可重构企业信息系统的框架应当具有可重用、可定制和可扩展的特性，它提供信息系统即插即用的集成机制。这种机制实现的难点在于已标准化封装的构件的组装/集成。如何实现软件体系结构框架的外部接口角色与构件的绑定，是实现可重构系统集成的主要任务。可重构信息系统的框架是面向特定领域的即插即用集成框架，框架上的重构集成元素是业务构件基于我们对业务构件的消息和通知服务的扩展，在基于 CCM 的业务构件外层添加了一个上端口和下端口，上端口 TOP_Interface 包含所连接的构件和连接器，与 TOP_domain 相连。下端口 bottom_interface 也是包含了业务构件和连接器。其语法结构的描述如下。

```

Topology ::=
{connector_name connections
top_interface connection_sequence

```

```

bottom_interface connection_sequence}
connection_sequence ::=
[where [not] connection_constraint]
[message_filter interface_message_filter_type]
connection_block_name ::=
component_instance_name | connector_name
connection_constraint ::=
boolcan_expr | msg_sequence | environment_command

```

经过框架体系结构配置语言编译器编译后，生成重构后的体系结构，完成角色与构件的绑定，即实现业务构件的组装和集成。同时完成连接器与构件以及连接器与连接器之间的绑定。

结束语 为了应对日益激烈的全球化市场竞争，企业信息系统必须具有可重构能力。本文基于软件体系结构描述语言 ADL 的 BNF 语法和 OMG/CCM 的构件技术，提出了一种基于 CCM 构件的企业可重构信息系统结构，针对企业信息系统的可重构软件体系结构模型，对系统的可重用业务构件和系统的即插即用框架进行研究和设计。通过对 CCM 业务构件的消息和通知服务扩充，实现了企业信息系统软件体系结构和功能应用软件的重用。

参考文献

- 肖亚军,张玉平. 基于 CORBA 构件的软件体系结构模型. 计算机工程, 2002, 28(10)
- 赵会群, 王国仁, 高远. 软件系统结构抽象模型. 计算机学报, 2002, 25(7)
- 李绪蓉, 丁秋林. 基于业务构件的快速可重构信息系统研究. 计算机科学, 2003, 30(3)
- 基于 CORBA 和多代理技术的可重构企业信息系统. 计算机集成制造系统, 2000, 6(3)
- 郭银章, 徐玉斌. 基于 CORBA 分布式应用技术的 Web 数据库集成体系结构研究. 计算机应用研究, 2003(10)
- 郭银章, 徐玉斌, 曾建潮. 基于 CORBA/WEB 技术的信息系统集成结构与模型研究. 航空计算技术, 2003(1)
- Chan F T S, Zhang J, Li P. Agent-And CORBA-Based Application Integration Platform For An Agile Manufacturing Environment. Int J Adv Manuf Technology, 2003, 21

(上接第170页)

量,使系统高速高效运行,尽量加快系统整体的时间响应,满足实时性要求。

参考文献

- Wang Dong sheng, Zheng Wei min, Wang Ding xing, Shen Meiming. Checkpointing and rollback recovery for network of workstations [J]. Science in CHINA Series E-Technological Sciences, 1999, 42(2): 207~214
- Park H S, Kim Y H, Kim D S, Kwon W H. A Scheduling Method

for Network-Based Control System. IEEE Trans. On Cont. Syst., 2002, 10(3)

- Linux Virtual Server Project. <http://www.linuxvirtualserver.org>
- LinuxParallelProcessingUsingClustersPro. HankDietz. <http://yara.ecn.purdue.edu/ppcluster>
- Hill J M D, Skillicorn D. Lessons learned from implementing BSP. Journal of future Generation Computer Systems. April 1998
- 毛羽刚, 金士尧, 张拥军. 与并行与分布硬实时系统的调度. 计算机科学, 1999, 26: 51~54