

# 异构分布环境下企业计算模型的研究与实现

詹 敏, 左 春, 孙玉芳

(中国科学院软件研究所, 北京 100080)

**摘 要:** 异构分布环境下企业关键应用的无缝集成是企业计算要解决的重点问题. 传统的企业计算模型采用自上而下的集成方法, 往往与自下而上的实际集成过程相矛盾. 本文提出的新的企业计算结构以对等计算技术为核心的基于松耦合消息协同的企业计算模型 UECM 可以完全适应这种自下而上的集成需求. UECM 目标在于为异构、分布、分散环境下的企业应用系统的互操作提供一个轻量的、低代价的、健壮的、可伸缩的基础平台, 同时克服传统方法在扩展性和服务器瓶颈问题上的不足.

**关键词:** 信息孤岛; 互操作; 对等计算; 对等实体; 服务; 消息; 通用企业计算模型

**中图分类号:** TP311 **文献标识码:** A **文章编号:** 0372-2112 (2002) 11-1681-04

## Research and Implementation of Enterprise Computing Model Under Distributed and Heterogeneous Environments

ZHAN Min, ZUO Chun, SUN Yur-fang

(Institute of Software, Chinese Academy of Sciences, Beijing 100080, China)

**Abstract:** Enterprise computing focuses on seamless integration of critical applications under distributed and heterogeneous environments. Traditional enterprise computing models are “top down” ones that are inconsistent with “bottom up” interoperation requirements. This paper brings forward a novel model named UECM for enterprise computing which is based on the coordination of loose coupled messages and peer to peer technology. It aims to provide a light weighted, low cost, robust and scalable infrastructure for interoperations among enterprise applications under heterogeneous, distributed and decentralized environment and better solve the problems of extensibility and server bottle neck caused by traditional models.

**Key words:** information island; interoperation; peer to peer computing; peer; service; message; UECM

### 1 引言

伴随着计算技术的发展, 企业应用系统的运行环境和计算方式也在不断改变. 从最初的大型机集中应用到客户机/服务器计算再到现在的浏览器/服务器计算和  $N$  层计算, 每个阶段都对企业应用系统产生重大的影响.

我们与中国人寿保险公司合作多年, 双方在各种平台上共同开发了一系列的应用软件. 同时, 其他公司也在参与开发中国人寿保险公司的其他业务系统. 这些由不同人员在不同平台上实现的业务处理系统需要彼此协同, 相互交换数据, 以保证公司业务流程的整体性和连续性. 另外, 网络技术的兴起也使保险公司感受到了通过 Internet 向其他企业或个人提供服务的可行性和必要性.

对中国人寿保险公司这样的大型企业而言, 多种计算技术混成的企业应用系统犹如一柄双刃剑. 一方面它能使新开发的企业应用程序跟上计算技术的最新发展, 并能根据不同的业务系统选择使用最能满足需求(如效率、可靠性等)的开发、运行环境和合作伙伴. 另一方面, 混成技术使企业的应用系统异构化, 从而导致信息孤岛的产生, 使企业应用系统之间无法自动地、顺利地进行交互.

企业是一个统一的实体, 企业的资源需要统一规划, 企业的业务流程需要协同运作. 而且, 企业之间的协同也是现代社会的一个特征和趋势. 企业与企业通过供应链连接而成一个

巨大的虚拟企业, 彼此共享经营信息和非经营信息. 如何将高度异构分布的企业应用系统集成化, 实现关键应用的无缝连接, 消弭信息孤岛, 是企业计算关注的焦点.

企业计算的研究对象是企业级的软件体系结构, 目的是使异构分布的企业应用系统能在该体系结构下协同运作, 实现企业内部以及企业之间的关键应用的互操作.

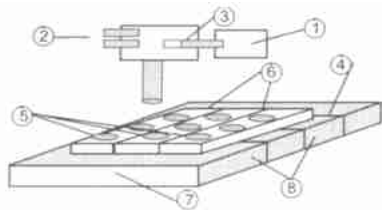
本文的后续部分首先回顾了现有企业计算的软件体系结构, 同时给出每一种结构的典型代表, 进一步论述它们的不足, 然后在此基础上提出 UECM (Universal Enterprise Computing Model, 通用企业计算模型), 并详细阐述 UECM 体系结构的组成和实现, 最后给出结论.

### 2 企业计算的软件体系结构

根据各应用系统与基础结构耦合的紧密程度, 可以将企业计算的软件体系结构分成两类: 紧耦合结构和松耦合结构.

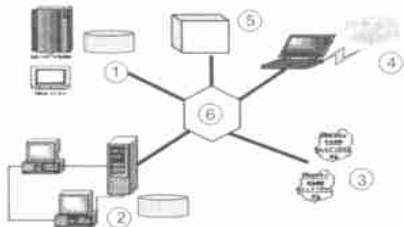
紧耦合结构如图 1 所示<sup>[1]</sup>. 实现了一个或多个接口(②)的异构分布的组件/程序(①)以约定的方式(⑤)嵌入到组件模型(⑥)中, 而组件框架(⑦)则为支持和增强组件模型提供了一系列的运行时服务(⑧). 不同的组件框架间可以根据约定的方式进行通讯. 这种结构的优点在于系统的基础结构(infrastructure)部分的同质, 因而能够很好地保证系统的安全、效率等. 缺点在于每个组件/程序以及它们之间的通讯都必须遵循特殊的约定, 对应用程序的改造很大, 不利于遗留系统

(legacy system)的集成和将来的扩展. CORBA、EJB、COM+ 等传统的  $N$  层计算技术是紧耦合结构的代表.



- ①组件实现(Component Implementation); ②组件接口(Component Interface)  
③接口规约(Interface Contract); ④独立配置(Independent Deployment)  
⑤组件类型和规约(Types and Contracts) ⑥组件模型(Component Model)  
⑦组件框架(Component Framework); ⑧协同服务(Coordination Services)

图 1 紧耦合结构示意图



- ①遗留系统(Legacy System); ②客户/服务器系统(C/S-based System)  
③组件对象(Objects); ④电子商务应用(E Business System)  
⑤商业软件包(Package); ⑥消息代理(Message Brokering)

图 2 松耦合结构示意图

松耦合结构如图 2 所示<sup>[2]</sup>. 松耦合方式指导异构分布的组件/程序(①...⑤)松散地分布在系统之中, 相互之间通过公共的(或约定的)报文(包括数据包和消息包)进行通讯. 松耦合结构的优点在于各异构分布的组件/程序能够保持原有的形态, 不需要进行大的或结构上的变动, 有利于当前以及将来各种应用程序的集成. 缺点是在松散环境下效率、安全、事务的管理比较复杂. 同时, 松耦合结构的中央服务器往往是系统的瓶颈, 单点失败是这种结构所要经常面临的问题. 消息代理(message brokering)是松耦合结构的代表.

由于无法保证所有的企业应用系统都采用同一平台, 更由于无法预见将来的计算技术, 因此, 平台独立的松耦合结构能够很好地保证企业计算的开放性和扩展性. 为了更好地解决松耦合结构所面临的问题, 为企业应用系统提供一个开放的、平台独立的、高效的互操作环境, 我们提出并实现了基于对等计算(peer to peer computing)技术的 UECM 结构.

### 3 UECM 的体系结构

UECM 基本上是一个松耦合的结构. 它有效地解决了传统松耦合结构的服务器瓶颈和单点失败问题, 核心思想是企业服务之间的对等计算. 服务是由应用程序提供的、可以被外部对象访问的、用于实现一定目的的功能性操作, 它是企业应用程序之间互操作的基本单位.

#### 3.1 对等计算及其在企业计算中的作用

企业应用系统的计算技术发展经历了单机计算、客户机/服务器计算、浏览器/服务器计算和  $N$  层计算几个阶段. 这些

计算技术在越来越强调高性能服务器的同时, 严重地忽略了日益强大的桌面计算能力, 在某种程度上造成了现有系统的服务器瓶颈和计算资源的浪费.

Napster<sup>[3]</sup>, Gnutella<sup>[4]</sup> 和 Freenet 等应用系统的出现代表着对等计算技术的兴起<sup>[5]</sup>. 对等计算是指通过系统间的直接交互实现计算机资源和服务的共享<sup>[6]</sup>. 参与对等计算的系统称之为对等实体. 与以往技术不同的是, 传统意义下的纯客户机在对等计算环境下既可以继续作为客户机也可以作为服务器存在. 也就是说, 对等计算能够充分利用现有的桌面计算能力和网络连接, 使计算机资源和服务的共享可以不通过传统意义下的服务器进行, 从而大大减轻了服务器的负载, 有效地避免了服务器瓶颈. 因此, 对等计算将是下一个计算技术浪潮, 是影响 Internet 未来的几大技术之一<sup>[7]</sup>.

企业应用系统是由异构、分散、分布的应用子系统组成的. 这种分散、分布的体系结构能够充分保证各子系统的自治权和控制权, 从而大大简化系统的开发工作以及后续的维护代价. 随着企业的发展, 分散分布子系统之间的互操作需求也在不断地增长, 企业往往要求能用最快的速度 and 最低的代价实现异构应用系统之间的互操作. 不幸的是, 目前的集中式企业计算模型并不能很好地满足这种需求.

从根本上来说, 当前的集中式企业计算模型是一种自上而下的应用系统集成方法, 它们使用的软件工程技术将整个企业应用系统看作是一个分布在不同节点上的单一程序(与之相反的是, 分散分布的企业应用系统的集成实际上是一个自下而上的过程, 它是由分散独立的应用子系统连接而成一个互操作的虚拟系统, 并保留各子系统的自治权). 为了更好、更有效地实现企业应用系统的互操作, 需要一个可行的、充分分布的并且基于网络的企业计算体系结构, 它能够满足企业内部以及企业之间集成的技术需求, 为以最小代价实现企业关键应用之间的互操作提供一个安全的、受控的、具有动态可访问性和灵活性的软件工程方法. 基于 Internet 发展起来的对等计算技术能够很好地满足我们的需要.

基于对等计算技术的企业计算模型能够为应用系统的互操作提供一个充分分布的框架. 该框架允许应用系统自动地发现其他应用系统的存在, 并根据企业业务流程的需要协调应用系统之间的关系. 同时, 由于该框架放弃了传统的集中式模型, 代之以与需要集成的应用系统相关联的软件组件, 因此, 对等计算技术将为企业应用集成带来很好的容错性、伸缩性和易维护性, 并使集成后的企业应用系统能很快适应外界的变化. 由此, 可以看出对等计算技术对企业应用系统的互操作有着特殊的意义, 利用对等计算技术能够有效地实现企业内部以及企业之间应用系统的信息共享和流程衔接, 是企业计算模型的未来和发展方向.

#### 3.2 UECM 的组成

UECM 是一个以对等计算为基础的松耦合结构, 目标是提供一个开放的、平台独立的、互操作的、具有良好的可扩展性和可维护性的企业计算环境, 能够安全、可靠、有效地进行数据传输和应用互操作. 利用一系列协议完成了 UECM 的描述, 也就是说, UECM 是基于协议而不是基于 API 的, 这样能

够最大程度地保证异构环境下的企业计算: 所有实现了该协议的企业应用系统都能够顺利地实现互操作, 而不管它们所处的平台环境和所具有的功能特性. 图 3 是 UECM 的逻辑结构示意图.

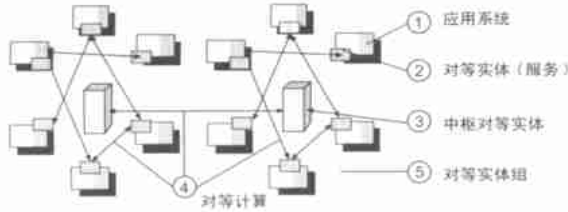


图 3 UECM 的逻辑结构示意图

任何参加对等计算的企业应用系统都应该根据 UECM 协议的规定将它愿意向外界提供的服务封装成对等实体, 企业计算将在对等实体之间进行(一个对等实体允许拥有多个服务). 相关的对等实体(领域相近或兴趣相近)可以组成对等实体组, 每个对等实体组都拥有一个中枢对等实体, 该对等实体提供对等实体组的信息以及组内每个成员的概要描述, 主要用于服务发现和成员管理. 同一组内以及组间的对等实体相互之间都可以进行对等计算.

UECM 的对等计算可以表示成如下的六元组:

$\langle \text{SourcePoint}, \text{Task}, \text{ServiceDiscovery}, \text{DestPoint}, \text{Execute}, \text{Message} \rangle$ , 其中:

SourcePoint: 对等计算的发起者

Task: 对等计算的内容(即发起者要求的服务)

ServiceDiscovery: 服务的发现机制

DestPoint: 对等计算的接受者

Execute: 执行服务

Message: UECM 消息

该六元组的含义是: 对等计算的发起者根据自己的意愿(所需要的服务), 通过服务发现机制寻找符合要求的服务, 然后向满足要求的对等实体(即对等计算的接受者)发出服务请求, 对等计算的提供者接受该请求, 执行服务, 并向发起者返回服务结果.

服务是应用程序提供的用于实现一定目的的功能性操作, 可以被外部对象所访问. 按照作用范围, 服务可以分成组内服务和组间服务. 组内服务只能被服务所属对等实体所在的对等实体组内的其他对等实体访问, 而组间服务则可以被所有的对等实体访问. 就其使用方式而言, 服务又可以分成推入式服务(push)和拉入式服务(pull)两种. 推入式服务属于主动服务, 可以主动寻找服务的预约者, 并将服务提供给它们. 拉入式服务则属于被动服务, 只有在被请求的时候才提供.

对等实体由处于同一应用程序内的一系列服务和若干属性组成, 并由对等实体生命周期管理工具负责创建、消除. 按照所拥有服务的性质, 对等实体可以分成服务实体、请求实体、预约服务实体或三者的任意组合.

对等计算的发起者和响应者是对等计算的参与者, 由对等实体和服务唯一确定.

消息是 UECM 对等实体之间的通讯手段, 对等实体之间

的对等计算就是通过 UECM 规定的一系列消息实现的. 消息包含命令和数据两部分, 用 XML 表示. 消息中 XML 标记的含义应该是对等计算的双方都能够一致理解的, 从而保证对等计算的顺利进行. 这种理解上的一致性可以来自以下几个方面:

- (1) 行业标准(如 OFX 等);
- (2) 服务提供者规定并被服务请求者了解;
- (3) 服务提供者和服务请求者协商的结果;
- (4) UECM 的规范.

图 4(a)和 4(b)是服务、对等实体和消息的 DTD 片段.

```

<! DOCTYPE Message[
<! ENTITY % peer SYSTEM "peer.ent" >
<! ELEMENT Message( sender, receiver, route* , content ) >
  <! ATTLIST Message message_id CDAT
# REQUIRED
message_type CDATA# REQUIRED
... >
<! ELEMENT sender( Peer, Service ) >
<! ELEMENT receiver( Peer, Service ) >
<! ELEMENT route( Peer ) >
<! ELEMENT content( # PCDATA ) >
% peer;

```

图 4 (a) 消息、服务和对等实体的 DTD 片段

```

peer.ent
<! ELEMENT Peer( Service+ ) >
  < ATTLIST Peer_ id CDATA # REQUIRED
Peer_ NAME CDATA # REQUIRED
... >
<! ELEMENT Service( service_ id, function, Parameter?,
Return? ) >
  <! ELEMENT service_ id( # PCDATA ) >
  <! ELEMENT function( # PCDATA ) >
  <! ELEMENT Parameter( para_ id, para_ type ) >
  ...
  <! ELEMENT Return( # PCDATA ) >

```

图 4 (b) 消息、服务和对等实体的 DTD 片段

服务发现机制是对等实体寻找所需要服务的方法. 当一个对等实体需要其他对等实体提供服务时, 它将向其他对等实体广播该信息. 所有接收到该信息的对等实体检查是否有满足条件的服务, 如果不满足, 则丢弃该信息, 如果满足, 则将满足条件的服务描述以应答消息的方式返回. UECM 提供了以下几种机制用于服务发现: 历史优先、组内广播、中枢实体和请求传递. 服务发现机制是 UECM 克服传统结构的服务器瓶颈和单点失败问题的关键.

### 3.3 UECM 的结构

图 5 是 UECM 的层次结构图.

• 网络协议指现有的 Internet 协议, 包括 TCP/ IP, UDP 等, 它们在进行对等计算的通讯基础.

• 实体、实体组和服务是 UECM 定义的基础对象.

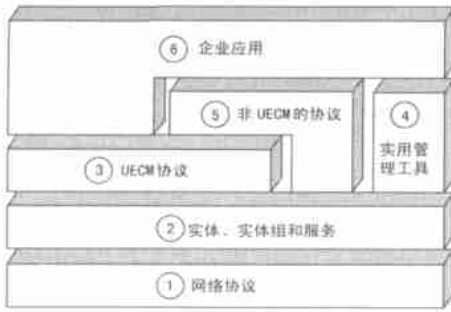


图 5 UECM 的层次结构

• UECM 协议规定了进行对等计算所必须的消息类型、消息格式以及完成各种对等计算所需要的消息交互过程。UECM 的协议包括：服务管理、对等实体/对等实体组管理、服务发现和服务请求等。其中：

①服务管理 规定服务注册、服务撤消、服务预约、预约撤消和服务信息查询等操作对应的消息格式和消息交互。

②对等实体/对等实体组管理 规定对等实体创建、撤消、变更、加入对等实体组、退出对等实体组、对等实体组创建、对等实体组撤消、指定/变更中枢对等实体和对等实体/对等实体组信息查询等操作对应的消息格式和消息交互。

③服务发现 规定对等实体用于发现其他对等实体组、对等实体和服务的消息格式和消息交互。服务发现是 UECM 的基础。

④服务请求 规定对等实体请求其他实体提供服务时的消息格式和消息交互。

• 实用管理工具是 UECM 建议的管理工具,用于为对等实体、对等实体组、服务等 UECM 对象的生命周期管理提供操作界面。

• 非 UECM 的协议是由 UECM 的实现者根据需要自行规定的协议。UECM 允许对等计算的最终实现者根据需要添加有关协议,最大程度地保证结构的灵活性。

### 3.4 UECM 的实现及特点

UECM 最早是作为中国人寿保险公司的企业计算解决方案而提出的,它快速、经济地实现了异构分布环境下企业关键应用的无缝连接,并且为将来的 B2B、B2C 之间的互动计算提供了一个良好的基础结构。

我们利用平台独立语言 Java 实现了 UECM 的互操作协议、对等实体包装器(wrapper)以及对等实体管理工具(UecmClient)和对等实体组管理工具(UecmServer),并成功地将有互操作需求的应用系统包装成对等实体,通过 UECM 基础结构实现与其他对等实体的互操作。

UECM 具有如下特点:

• 体系结构开放性 UECM 的协议、规范都是基于已有的、开放的工业标准(XML, TCP/IP 等),只要遵循 UECM 的协议规范,任何应用系统都可以与其他系统实现互操作。

• 平台独立性 UECM 的实现与目的环境的平台无关,实现者可以根据自己的实际需要任何平台上实现 UECM。

• 易用性 企业应用系统只需要经过简单的包装就可以

向其他应用系统提供服务并获得服务。

• 安全性 UECM 定义了企业服务的访问权限。通过对企业服务的权限管理,可以保证企业服务只被授权用户访问。

• 灵活性 UECM 允许实现者根据实际需要定义自己的协议,充分保证了系统最终实现的灵活性。同时,不同应用系统之间的数据交互方式既可以是实时的,也可以是成批的。

• 健壮性 UECM 通过对等实体和服务发现机制保证企业计算的健壮性,有效地防止了单点失败和服务器瓶颈问题,为企业计算提供了一个可靠的基础结构。

## 4 结束语

本文详细论述了基于对等计算的 UECM 体系结构。UECM 致力于解决异构分布环境下企业计算所面临的问题。以 CORBA 为代表的紧耦合结构和以 Message Brokering 为代表的松耦合结构在处理企业计算时所面临的问题使它们无法成为理想的解决方案。对等计算能够充分利用传统意义下的客户端计算能力,有效地避免服务器瓶颈和单点失败问题,保证系统的开放性和易扩展性,因而成为企业计算的理想方案。

UECM 为企业应用系统互操作提供了一个公共的基础结构,使异构分布的企业应用在该基础结构上能够安全、有效地实现信息和流程的共享。

目前我们已经用 Java 实现了 UECM 规定的协议以及相关的管理工具(如服务管理、对等实体管理等),我们将继续在其他环境下实现 UECM 的协议,进一步验证 UECM 对异构环境下企业计算的有效性。

参考文献:

- [ 1 ] Felix Bachman, Len Bass, Charles Buhman, Santiago Comella Dorda, Fred Long, John Robert, Robert Seacord, Kurt Wallnau. Volume II: Technical Concepts of Component-based Software Engineering[ R]. CMU, Pittsburgh, 2000.
- [ 2 ] David S Linthicum. Enterprise Application Integration[M]. Boston: Addison Wesley Pub, 1999.
- [ 3 ] Napster[ EB/OL]. <http://www.napster.com>.
- [ 4 ] The Gnutella protocol specification[ EB/OL]. <http://dss.clip2.com/GnutellaProtocol104.pdf>, 2000.
- [ 5 ] Andy Barnhart. Share and share alike[ EB/OL]. Software Development Online, February 01, 2001.
- [ 6 ] <http://www.peer-to-peerwg.org/what-is/index.html>[ EB/OL].
- [ 7 ] Amy Kover. Napster: The hot idea of the year[ J]. FORTUNE, Monday, June 26, 2000.

作者简介:



詹 敏 男, 1974 年 4 月生, 福建省闽清县人, 中国科学院软件研究所博士研究生, 研究方向为大型数据库与网络工程。

左 春 男, 1959 年 4 月生, 辽宁省人, 中国科学院软件研究所研究员, 研究方向为大型数据库与网络工程。