

基于 ERP 系统的消息中间件设计与实现

赵伟¹, 周兵²

(1. 河南省信息网络重点开放实验室, 郑州 450052; 2. 郑州大学信息工程学院, 郑州 450001)

摘要: 由于网络体系结构、操作系统以及数据库系统的异构性, 目前 ERP 系统与其他应用系统间稳定的数据共享仍是有待解决的问题。为提高 ERP 系统的耦合性, 该文提出一种基于消息中间件的设计和实现方案, 以消息中间件为基础, 能够把含有多个数据库、多种操作系统的企业应用结合成一个协同工作的有机整体, 实现企业的分布式应用。该方案在多个大型销售企业的实际应用表明, 它改进了数据传输安全性和传输效率以及代码的可移植性。

关键词: 消息中间件; EDI 技术; 接口

Design and Implementation of Message-oriented Middleware Based on ERP System

ZHAO Wei¹, ZHOU Bing²

(1. Henan Provincial Key Lab on Information Network, Zhengzhou 450052;

2. College of Information Engineering, Zhengzhou University, Zhengzhou 450001)

【Abstract】 At present the steadily data share between ERP systems and other applications is still a major issue to be solved, due to the heterogeneity of networks, operation systems and database systems. The paper presents a project solution based on message-oriented middleware. In order to improve the coupling of the ERP system, the solution combines the enterprise applications based on different operation systems and databases into a unified system, and realizes the real distributing applications. This solution has been used in several large sale enterprises and can improve the security and efficiency of data transferring and the portability of codes.

【Key words】 message-oriented middleware; EDI; interface

1 概述

随着计算机网络全球化的发展, 系统间的交互协作变得越来越富有挑战性, 而且系统间的信息共享、数据格式和语法多样化的需求也日益增长。但是每个信息系统都是相对独立和完整的, 在信息系统中, 系统开发的时间、所使用的技术与开发工具以及系统的运行平台都是不同的, 信息存在的形式和访问方式也各不相同, 这为信息的共享与集成带来多于预期的工作, 例如在网络数据传输、不同平台间的编码转换、事务协调、数据安全、数据转换与重组等。而且随着 IT 技术的不断发展, 越来越多的企业开发并实施自己的信息系统, 每个企业都有少至几个、多至几十个信息处理系统, 信息的共享和重复使用已成为当今系统构建的主要任务之一。

信息处理可分为同步信息处理(synchronize processing)和异步信息处理(asynchrone processing)两种方式, 高速的计算机和通信设备并不能解决异步信息处理问题。以物流业务为例, 当中国销售企业的销售人员白天处理一笔涉外的进口订单业务时, 需要从在美国的工厂获取有关信息, 而这时美国正处于午夜, 计算机应用系统已关闭, 此笔进口订单业务无法完成; 如果等到美国的计算机应用系统开通时, 中国又进入了夜晚。传统的方法只能通过传真方式解决, 但其不能纳入计算机自动处理, 而且在手工处理时也极易产生错误。这是一个典型的异步信息处理问题。

现在已经有一些计算机技术例如 EDI, CORBA, DCOM 和

Java RMI 等都在信息处理和信息的共享方面取得一些初步的成功^[1]。由于它们的复杂性(例如 CORBA, DCOM)和操作平台的特殊性(例如 RMI)等原因, 因此都没有得到企业的支持和接受。然而, 这些尝试已经紧密地连接了企业的系统, 尽管这种连接不适合企业间电子商务系统间的连接。消息中间件(Message Oriented Middleware, MOM)^[1]是一个能支持系统间异步消息交流的中间件平台, MOM 平台提供 API 接口和在不同类型的应用程序间的信息的异步传送。为了使分布式应用程序更容易结合为一个整体, MOM 通常被作为解决分布式应用程序间交互的一种方法^[2]。

本文以 IBM 的 MQ Series 的产品为基础, 提出一种 MOM 实现方案: 将销售部门的销售 PSI 计划以 EDI 的格式, 通过消息的形式发送到 MQ 的消息队中, 制造部门从 MQ 对列中取到该消息, 生成对应的生产计划, 通过在 EDI 中将重要数据加密, 然后以消息形式传送至消息队列, 保证了数据的安全性, 同时也保证了销售企业和工厂间信息的异步传输的及时性; 将 MOM 与 EDI 两种方式相结合, 把含有多个数据库、多种操作系统的企业应用结合成一个协同工作的有机整体, 真正实现了企业的分布式应用。通过多个大型销售企业的实际应用表明, 在数据传输安全性、效率以及可移植性方面都有明显的优势。

作者简介: 赵伟(1976-), 男, 硕士, 主研方向: 分布式系统; 周兵, 教授、博士

收稿日期: 2007-05-25 **E-mail:** iezhaowei@zzu.edu.cn

2 MOM 的功能

2.1 MOM 的概念

MOM提供了可靠的数据传输机制,尽管它使系统间松散地耦合在一起,即并不总是以同一速度运作,有时系统间还是处于不连接状态。但是MOM在分布式应用程序间传输消息时能够处理所有网络之间的传输,使应用程序的开发者不需要考虑网络底层的传输,并在客户端和服务端建立一个共同传输媒介^[3]。

为了使销售企业的ERP系统与供应商之间形成完善的物流链环节、提高各自的经济效益,本文提出利用MOM构建消息的传输通道,将Oracle Ebs11i 电子商务套件和供应商之间不同操作系统的企业应用系统结合成一个协同工作的有机整体,真正实现了企业的分布式应用。

2.2 ERP Services Message Queue 的功能

ERP Services Message Queue 核心消息队列能够被系统管理员动态地创建,用于保存数据并具有消息处理功能,如图1所示。无论何时,当一个消息进入到ERP Services Message Queue,它将会经历几个处理阶段,直到一个合法的响应被正常传回给发送消息的一方。这些处理阶段包括消息备份、消息解密以及消息队列管理,如果消息被加密发送,则通过 Decryptor 解密还原消息。

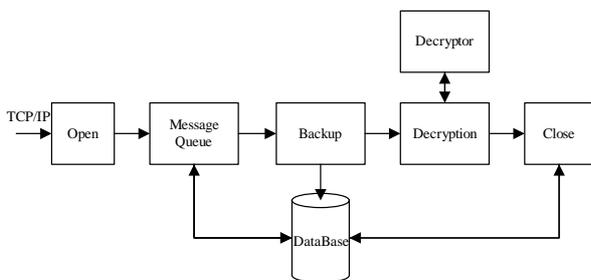


图1 ERP Services Message Queue 功能

2.3 系统结构设计

ERP Services Message Queue 系统体系结构如图2所示。

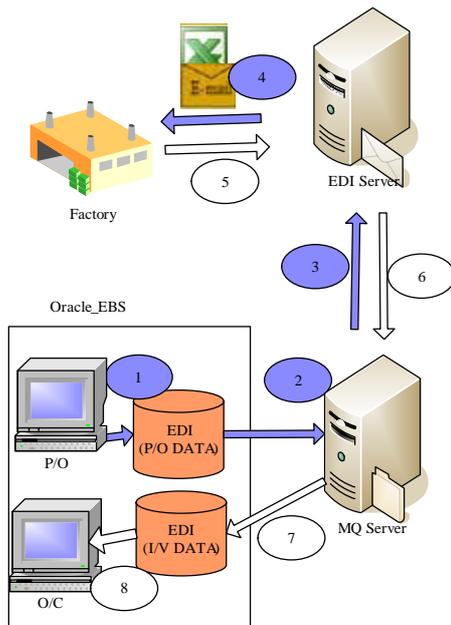


图2 系统体系结构

整个系统分为3个部分：

(1)Oracle_EBS11i 系统中的消息形成部分,通过 Data

Output 处理程序将 EBS Add-on 表中抽出需要发往制造企业 Factor 的订单信息文件,该信息文件以一定的格式存储采购订单的重要信息,它生成后定时自动调用 Data 备份和启动 Java 组包发送程序,即先将这些文件先备份,然后调用 Shell 传送程序将该文件按一定消息的格式组织成消息数据包,将其发送到 MQ Sever;

(2)消息处理,MQ Sever 得到消息后,保存 MQ 消息,并得到相关表和字段信息,即采购订单的 P/O 数据;

(3)EDI 处理,将取得的 P/O 数据插入 EDI Server 中对应的 P/O 表中,EDI Server 立即按照一定的格式约定将其转换为可读的 EXCEL 文件,并将其以 E-mail 的方式发给工厂。

在整个系统中 MQ Server 是桥梁和关键点,它通过 ERP Services Message Queue 异步地接收来自 Oracle EBS 和 EDI Server 的 message,并进行相应的数据库操作。

2.4 消息的状态转换

通过以消息状态转换图(图3)来说明消息在传送过程中的状态,图中的状态代码及说明见表1。

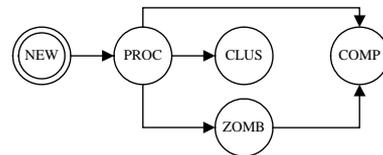


图3 消息的状态转换图

表1 状态代码及说明

状态代码	说明
NEW	EDI 数据文件按消息的格式封装成消息数据包
PROC	消息被发送至本地消息队列,保持该状态直到有命令让它传输
CLUS	当目标队列不存在时,队列管理器寻找到一个与目标队列地址相近的队列,并发送至该队列
ZOMB	当消息发送目标队列不成功时,消息状态为“ZOMB”
COMP	目的队列已成功收到消息,发一个应答给原队列

当消息在“NEW”状态后,将其发送到本地队列,准备传送到目标对列,此时消息的状态更新为“PROC”。如果状态为“PROC”的消息通过通道,成功地发送至目标对列后,目标对列发一个应答给原对列,并将消息状态更新为“COMP”。如果状态为“PROC”的消息发送到目标对列,当目标对列不存在时,发送到与目标对列地址相近的队列,接受对列接收成功,发送一个响应给原对列,并更新状态为“CLUS”。如果状态为“PROC”的消息发送目标对列不成功,此时更新状态为“ZOMB”,队列管理器定期收集此状态的消息,重新发送。

2.5 MOM 的安全性

为了保证企业内部的网络安全,企业分为外网和内网两个部分。外网和内网之间通过 MOM 进行数据传递。MOM 的安全性从以下3点进行设计:

(1)认证方面,采用用户名/密码对。基于独立数据库的用户名/密码对技术能够用于不同MOM管理器之间的认证,以及 MOM管理器与使用其服务的应用程序之间的认证^[3]。

(2)对 MOM 产品所拥有对象的访问控制机制。严格控制并只允许一部分用户访问一个对象而阻止其他用户访问相同的对象。对于文件的安全性,粒度通常就是指读、写、执行、删除、创建等访问权限。

(3)用唯一的且与发送方相关联的代码标识每条消息,并封装每条消息。

(4)对于重要的数据，进行加密封装发送。

3 MOM 的实现

3.1 ERP Services Message Queue 的实现

本系统中 MOM 采用 C/S 结构，MQ 服务器端和 EDI Server 数据库服务器分别置于不同的服务器上，EDI Server 中也有 MQ 客户端。在 MQ 服务器端建立一个 Queue Manager，管理应用所需的所有队列，保证放在队列中的消息的完整性和可恢复性，消息放入队列后，即使断电或硬盘故障等引起系统崩溃，重启后，恢复机制仍能保证消息完整。

根据业务和系统的需要，队列管理器和各队队列按规律定义名称如下^[4]：

队列管理器名：VQM_GW_AYBQ74_ED1

队列名称：Q_ED1_LCL_001, Q_ED1_LCL_002, Q_ED1_LCL_003, ...

系统间消息传递如图 4 所示。

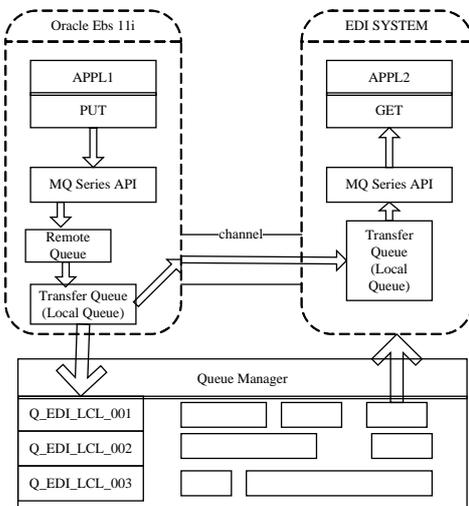


图 4 系统间消息传递

3.2 实验及分析

笔者在某项目中使用了 ERP Services Message Queue，通过分析比较使用消息中间件前后的数据结果，发现采用 MQ 优点如下：

(1)提高了响应时间。图 5 是本项目的系统测试数据，其中，Original 表示没有应用 MQ 前的测试数据；Via ERP Services MQ 表示应用 MQ 后的测试数据；纵轴表示每个客户机发送 50 个消息至目标队列加收到目标队列发回应答的平均时间。实验结果表明应用 MQ 传递消息，响应时间得到了明显的提高。

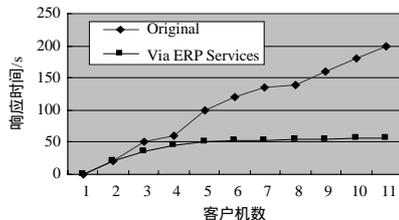


图 5 ERP Services MQ 性能测试

而且 ERP Services Message Queue 可以保证 EBS 系统和 EDI 系统之间的采购订单信息不丢失且不重复，只传送一次。

(2)支持异步处理。ERP Services Message Queue 不仅支持传统的同步的通信模式，也支持异步的工作方式。消息创建和发送时，消息的发送方到接收方的网络连接不需要建立，两端的通信程序可以运行在不同的时间，如果逻辑允许，它们不必等待其他程序的应答而继续工作，利用这种异步处理功能，可以使得采购订单的传输处理可以是独立的，并行的。

(3)缩短了应用开发周期。两个完全相同功能的模块分别采用 MQ 和 RMI 实现系统接口间开发，开发进度如图 6 所示。项目实际结果证明，ERP Services Message Queue 负责处理所有的网络通信问题，包括不同的网络协议以及计费数据传输过程中的故障诊断和恢复，开发人员不需要掌握系统和网络通信编程，极大地提高了开发效率，加快了项目的进度，降低了项目的风险，进而节省了项目经费，具有较强实用性。

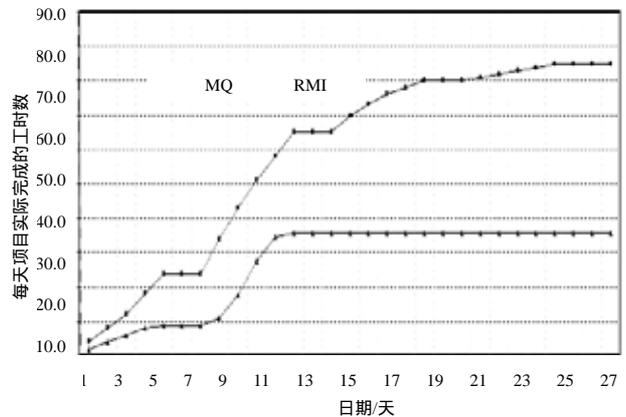


图 6 MQ 和 RMI 开发进度

4 结束语

构建安全、及时、可靠、稳定的物流链具有一定实际意义，MOM 技术的快速发展为构建现代物流链带来了新的机遇。本文提出的基于 MOM 的数据传输技术，极大提高响应时间和项目的进度，同时又具有很高的安全性和可靠性，并能很好地解决跨平台问题，并已在多个大型的销售企业付诸实施。

参考文献

- [1] Kien T N, Erradi A, aheshwari P M. WSMB: A Middleware for Enhanced Web Services Interoperability[C]//Proc. of Interoperability of Enterprise Software and Applications Conference. Geneva, Switzerland: [s. n.], 2005.
- [2] Zbib R, Anjum F, Ghosh A. Intrusion Tolerance in Distributed Middleware[J]. Information Systems Frontiers, 2004, 6(1): 102-105.
- [3] Maheshwari P. Benchmarking Message Oriented Middleware TIB/RV Versus SonicMQ[J]. Concurrency and Computation: Practice and Experience, 2005, 17(12): 200-205.
- [4] Maheshwari P, Tang Hua. Enhancing Web Services with Message-oriented Middleware[C]//Proceedings of the IEEE International Conference on Web Services. [S. l.]: IEEE Press, 2000.