

多 Agent 协作技术在电子商务中介平台中的应用

郭文生¹, 杜军平², 尹怡欣¹

(1. 北京科技大学信息工程学院, 北京 100083; 2. 北京工商大学计算机学院, 北京 100037)

摘要: 描述了一个 B2B 电子商务中介平台的设计与实现过程。在设计过程中广泛地借鉴了 J2EE 技术和 Mobile Agent 的特点和优势, 完成了二者的有效结合。并结合电子商务中介平台的特点完成了中介平台 Agent 的功能设计和 Agent 与 J2EE 互操作中间件的研制。

关键词: 移动 Agent; J2EE; 电子商务

Application of Multi-agent Corporation in E-business Intermediation Platform

GUO Wensheng¹, DU Junping², YIN Yixin¹

(1. School of Information Engineering, Beijing University of Science and Technology, Beijing 100083;

2. School of Computer, Beijing University of Technology and Business, Beijing 100037)

【Abstract】 This paper describes the realization of a B2B intermediation e-business platform. The main concern derives from the usage of two emerging technologies, mobile agent and J2EE. It also achieves the design of intermediation agent and the development of agent-J2EE interoperability middleware.

【Key words】 Mobile agent; J2EE; E-business

在电子商务的相关文献中,对中介平台的研究吸引了广泛的关注并成为了新的热点。在现实的电子商务活动中,必须建立一种机制使得参与交易的各实体(主要是消费者和供应商)之间能够彼此方便地发现对方并快速精确地完成交易,从而有效地减少花费在寻找有用信息上的时间。这种机制可以通过中介平台来提供。中介平台为电子商务交易活动提供了保障,从而缩小了消费者和供应商之间的距离。

本文描述了一个针对 B2B 交易的中介平台,该平台通过对 Mobile Agent 和 J2EE 技术的应用,可以适用于多种业务领域。我们在这一体系结构的基础上实现了国家旅游局的电子商务中介平台。

1 电子商务中介平台建模

1.1 中介平台的交易流程

我们可以把一个复杂的中介交易模型分解为 3 个基本域:用户域,服务提供者域和代理域。一个电子商务中介交易周期可以分为以下阶段:(1)对交易请求进行规范化处理并对交易方案进行预处理;(2)双方理解交易方案并就交易方案进行谈判;(3)对交易方案进行评价;(4)交易执行。

当消费者访问系统时,会通过浏览可使用服务的分级菜单来构造交易请求。平台通过 Ontology 技术所提供的知识来支持用户动态规划复杂的交易请求。当系统对用户请求进行确认之后,就会调用其内部的功能找到一个可行的交易方案。交易方案是服务提供者提出的一系列可以满足用户请求的报价。系统将对每个构造交易请求的简单服务分别进行理解。

当交易方案的理解过程结束以后,系统就会把交易方案提交给消费者。中介平台允许消费者进一步访问被代理资源(服务提供者)。如果平台的用户没有找到合适报价的话,可

以重新定义交易请求。

一旦系统接受了交易方案之后,接下来就进入交易执行阶段。在该阶段,平台将执行交易的服务并与相关的服务提供者域进行交互。

1.2 基于 J2EE 技术的中介平台体系结构

国家旅游局的电子商务中介平台采用了 J2EE 的多层体系结构,整个应用被分为用户界面层、业务逻辑层和数据访问层(如图 1 所示)。平台运行在 J2EE 应用服务器上,由一系列可重用的组件组成。

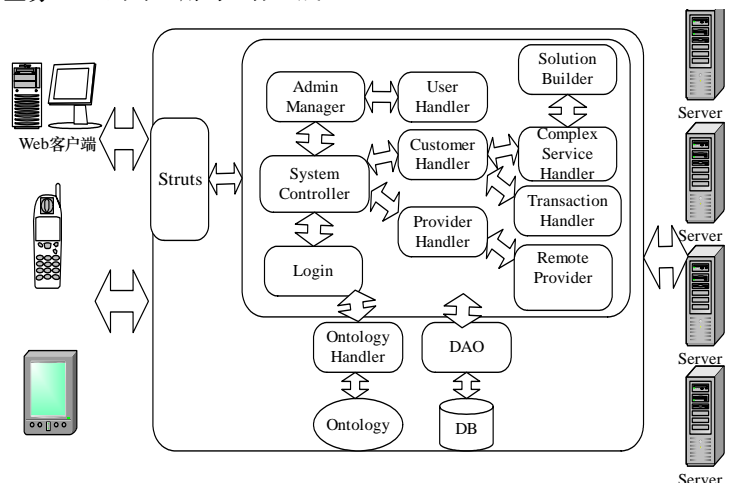


图 1 基于 J2EE 技术的中介平台体系结构

中介平台可以支持 Web 浏览器或手持设备等多种异构的客户端。这些客户端都支持 HTML 或 WML 的页面,中间层

作者简介: 郭文生(1977—),男,硕士,主研方向:人工智能;杜军平、尹怡欣,博士、教授

收稿日期: 2005-10-08 **E-mail:** gwsice@126.com

的 Servlet 可以对这些页面进行处理。

表示层是客户端访问中间层业务逻辑的入口，采用基于 MVC 模式的 Struts 框架实现。框架负责分析 Servlet 接收到的 XML 流并把对应的 Action 分配给相关的 Bean (UserHandler、ProviderHandler、LoginBean) 执行。

系统的数据访问层采用了 DAO 设计模式，由两类 Bean 完成不同类型的数据访问任务：Ontology Handler 负责与 Ontology 知识库进行交互，Data Controller 封装了数据库访问的细节。中介交易流程的处理由下面 3 类 Bean 来完成：(1)Complex Service Handler 可以将用户请求分解成一系列的原子请求，这些原子请求可以由不同的服务提供者完成，然后再被组合起来完整地提供给用户。(2)Solution Builder 负责根据用户请求寻找合适的提供者。(3)Transaction Handler 可以完成整个的交易循环。

Remote Provider Handler (RPH) 负责本系统与外部系统的互联，它可以创建并指定 Mobile Provider Agent (MPA) 完成平台与外部提供者之间的交互。在服务提供者域里有两类 Agent，它们是静止提供者 (SAPA) 和移动提供者 (MAPA)。

2 Agent 技术在电子商务中介平台中的应用

平台与外部服务的交互是一种双向的过程。也就是说，外部服务提供者可以主动与代理域进行联系；同时，代理域也可以同外部服务进行交互。

2.1 Agent 功能设计

平台允许每个外部提供者与平台进行联系以便及时更新报价信息数据库。这些更新操作包括 add、update 和 delete 等。在这一过程中，SAPA 首先创建一个 MAPA 并赋予它对数据库进行访问的权限。随后，MAPA 将移动到代理域，经过认证后在数据库上执行对报价信息的增、删、改操作。

在中介平台中，管理员负有对系统进行全局监控的责任。它可以请求一个特定的报价或者一个服务提供者的完整报价目录。而且，在发现数据库中的报价已经陈旧之后，还有权对报价进行更新。AdminManager 首先调用 RPH 创建并初始化一个 MPA。RPH 为 MPA 给出了对外部提供者的访问授权和报价信息名字。随后，MPA 将移动到提供者域与 SAPA 进行交互得到授权，并从提供者的本地数据库获取相关信息。最后，MPA 移动返回中介平台并更新平台的报价数据库。其具体操作流程如图 2。

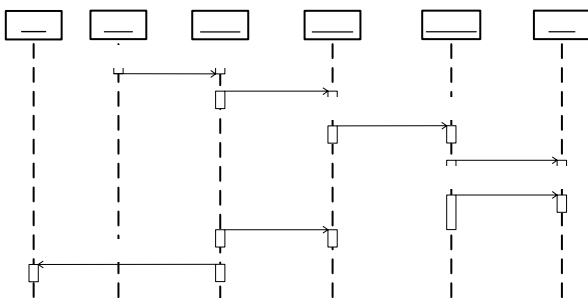


图 2 管理员对平台报价的更新过程

在根据用户请求寻找合适报价的过程中，Solution Builder (SB) 可能会发现一些存储在平台数据库中的报价信息是不完整的。在这种情况下，SB 就可以同报价的提供者进行联系来更新报价，最后再完成用户请求的匹配工作。上述过程所执行的具体操作如下：SB 首先通过调用 RPH 来创建一个 MPA。MPA 携带不完整的报价的编码移动到报价的提供者，在经过必须的授权之后从 SAPA 那里得到完整的报价信

息，最后返回。在系统构建解决方案的过程中，如果顾客选择了对报价的有效性进行在线检测的话，系统就须同远程的服务提供者进行通信。这样可以避免向用户给出无效的解决方案。如图 3。

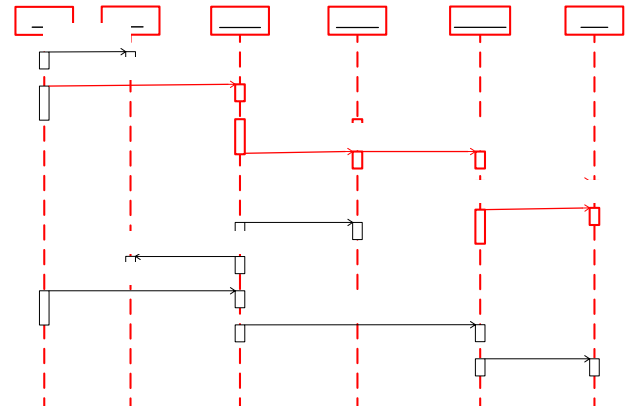


图 3 有供应商参与的解决方案构建

当消费者确定了解决方案之后，平台就开始了交易处理过程。整个过程由以下任务组成：

- (1)与各个服务提供者联系，对构成解决方案的所有原子报价进行预订。
- (2)批准所有预订成功原子报价。
- (3)如果任何一个原子报价没有预订成功或者没有得到批准的话就取消整个交易过程。

在交易处理过程中，Transaction Handler (TH) 由消费者主动触发并处理整个交易细节，然后调用 RPH 并与远程的 SAPA 进行交互以执行每一项任务。

2.2 Agent 组件的实现

在基于多 Agent 的电子商务中介平台的研制过程中，我们采用了通用的 Mobile Agent 开发平台 Grasshopper 来实现 Agent 组件的开发。Grasshopper 可以在一定程度上提供 Mobile Agent 技术与 J2EE 体系结构的集成，从而进一步提升系统的性能。

Mobile Agent 在运行时需要在分布式计算环境的所有节点上都提供 MA 的运行环境。这种运行环境可以支持 Agent 的创建、执行、定位、迁移、通信和安全控制等功能。Grasshopper 是一种构建于分布式处理环境之上的 Mobile Agent 平台中间件，它体现了一种从普通的分布式软件对象向 Mobile Agent 发展的趋势。Grasshopper 提供了所有上述的服务，同时也可与兼容 MASIF 标准的其它 MA 平台交互。

2.3 Agent-J2EE 交互中间件

由于 Grasshopper 平台为电子商务中介平台的建立提供了许多优势，因此，Agent 技术在电子商务领域的应用前景也毋庸置疑。尽管如此，在 Grasshopper 和 J2EE 技术的互操作问题上还是存在着一些问题。

Grasshopper 使用了一种特殊的通信服务来处理分布式组件之间的远程通信活动。这些活动包括 Agent 之间的位置无关通信、Agent 传输和 Agent 的定位等。此外，这一通信服务还使得 EJB 等外部应用对象也可以同 Grasshopper 的 Agent 进行交互。这些外部应用对象可以作为客户端去调用 Agent 提供的方法。另一方面，它们也可以作为服务器去接收其它请求，这时它们需要遵守通信服务给出的规则。这些应用组件需要实现特定的服务器接口，然后启动一个 receiver 在特定的地址范围和端口上对 Grasshopper Agent 的请求进行

监听。在这种情况下，基本的 Grasshopper 通信服务架构既不允许 Grasshopper Agent 创建新的 EJB 实例，也不允许 EJB 把自己作为一个服务器对象注册到通信服务接收器上。在这种条件下，如果 Agent 想要访问中介平台数据库的话，就必须自己实现 JDBC 和 SQL 的数据访问代码。这种方法虽然能够实现，但是也有相当明显的缺点。

实际上，中介平台为了优化数据访问机制实现了一个统一的数据访问层，其它代码需要调用这一接口对数据库进行访问。既然在中介平台的 DAO 组件中已经实现了数据访问，那么在 Agent 里面再次编写这些数据访问代码就显得重复浪费了。为了克服这些缺点，我们在中介平台中开发了 Agent-J2EE 交互中间件来完成二者之间的交互。

Agent-J2EE 交互中间件是一个独立运行的 J2EE 应用程序，可以运行于各类 J2EE 应用服务器上并能够访问 J2EE 所提供的服务。它能以一种静止的 Agent 的方式运行，并与多 Agent 系统进行交互，同时又可以访问 J2EE 应用服务器内部署的 EJB 组件。交互中间件的实现机制如图 4 所示。

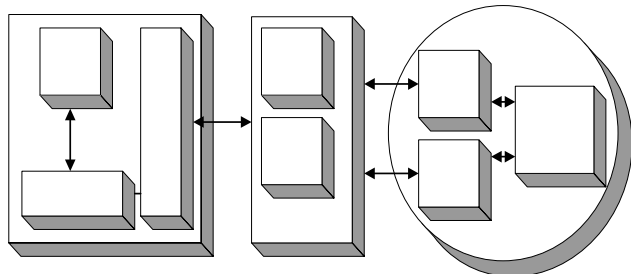


图 4 Agent-J2EE 交互中间件的设计

整个 Agent-J2EE 交互中间件由两部分组成：服务器对象

模块实现了服务器接口并提供了对 EJB 方法的访问。主模块负责启动通信服务接收器并将上述服务器对象进行注册。因此，当 Agent 需要调用 EJB 方法的时候，它只需使用 Grasshopper 所提供的标准通信服务就可以了。通过 Agent-J2EE 交互中间件，就能实现 Grasshopper 平台没有提供的 Agent 和 J2EE 组件之间的双向无缝集成，有效地提升了整个应用架构的灵活性和互操作能力。

3 结论

本文描述了一个综合了 Agent 和 J2EE 技术的电子商务中介平台的设计与实现。并详细介绍了系统的功能组成和 Agent-J2EE 互操作性的实现方法。通过对 Mobile Agent 和 J2EE 技术的有效整合，使电子商务中介平台具有更大的主动性和自治能力，同时也充分体现出了 J2EE 基础架构所带来的稳定可靠的优势，在中国旅游电子商务的实际应用中取得了良好的效果，能推广应用。

参考文献

- 1 Johnson R. 魏海萍译. J2EE 设计开发编程指南[M]. 北京: 电子工业出版社, 2003.
- 2 Bichler M, Segev A. A Brokerage Framework for Internet Commerce[J]. Distributed and Parallel Databases, 1999, 7(2): 133-148.
- 3 IKV++ Technologies AG. Grasshopper MAP(v2.2.3)[Z]. <http://www.grasshopper.de>, 2004.
- 4 Lopez M F. Overview of Methodologies for Building Ontologies[C]. Proceedings of the IICAI'99 Workshop on Ontologies and Problem-solving Methods(KRR5), 1999.

(上接第 254 页)

表 3 预测精度情况

年度	1997	1998	1999	2000	2001	2002	2003	2004
观测值	68664800	69071220	73317128	81746580	88924831	93049977		
预测值	67150869.67	70585145.65	74865016.21	80198685.39	86845621.69	95129180.48	105452334.33	118317276.59
误差	1513930.33	-1513929.65	-1547888.21	1547890.61	2079210.31	-2079204.48		
相对误差 (%)	2.020481	-2.191839	-2.111223	1.893524	2.338166	-2.234503		
平均相对误差	2.162344							

(单位: kW·h)

根据式(39)进行预测其精度情况见表 3。由表 2 和表 3 可以看出，改进的修正指数曲线预测法在电力系统年度负荷预测中，比文献[3]中介绍的方法具有更高的预测精度，这一精度完全能够满足生产和管理部门的需要。

3 结束语

文献[3]中求解修正指数曲线参数的方法存在一定的不足，即没有充分利用所有观测数据的信息，而且适用范围也有一定的限制。本文提出的改进的修正指数曲线预测法不但克服了这一不足，而且经示例计算结果表明，改进的修正指数曲线预测法比文献[3]中的求解方法具有更高的预测精度，平均相对误差为 2.110 691%，相对误差最大者未超过 2.7%。这一

预测精度完全能够满足生产和管理部门的需要，是一种行之有效的理想预测方法。

参考文献

- 1 赵宏伟, 任震, 黄雯莹. 基于周期自回归模型的短期负荷预测[J]. 中国电机工程学报, 1997, 17(5): 348-351.
- 2 梁志珊, 王丽敏, 付大鹏等. 基于 Lyapunov 指数的电力系统短期负荷预测[J]. 中国电机工程学报, 1998, 18(5): 368-371.
- 3 牛东晓, 曹树华, 赵磊等. 电力负荷预测技术及其应用[M]. 北京: 中国电力出版社, 1999.
- 4 同济大学数学教研室. 高等数学[M]. 北京: 人民教育出版社, 1987.
- 5 孙靖民. 机械优化设计[M]. 北京: 机械工业出版社, 2001.