

# Agent-based P2P 环境下开放式服务体系架构研究<sup>\*</sup>

夏 阳<sup>1</sup> 陈贵海<sup>2</sup> 张 强<sup>1</sup> 徐 钊<sup>3</sup>

(中国矿业大学计算机科学与技术学院 徐州 221116)<sup>1</sup> (南京大学计算机科学与技术系 南京 210093)<sup>2</sup>  
(中国矿业大学计算机信息与电器工程学院 徐州 221116)<sup>3</sup>

**摘 要** 将 Agent 的概念引入 P2P 服务体系架构中,会对整个架构的灵活性和自适应性产生重要的影响。在深入讨论和研究 Agent-based P2P 环境独特的技术特点的基础上,设计并提出了 Agent-based P2P 环境下开放式服务体系架构模型,即 OSA-ABP。模型架构由 5 层分层模型和多 Agent 交互模型构成,与同类架构比较,在架构的开放性、异构 Agent 兼容性、资源统一描述发布等方面具有较高的技术优势。

**关键词** P2P, 移动代理, Web 服务, 迁移机制

## Open Service Architecture in Agent-based P2P Environment

XIA Yang<sup>1</sup> CHEN Gui-hai<sup>2</sup> ZHANG Qiang<sup>1</sup> XU Zhao<sup>3</sup>

(School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)<sup>1</sup>

(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)<sup>2</sup>

(School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China)<sup>3</sup>

**Abstract** Putting the concept of Agent into P2P service architecture brings the better influence on agility and adaptability for the architecture. Based on the deep presearch of relative knowledge, open service architecture in Agent-based P2P environment, named OSA-ABP, was proposed in this paper. The architecture was made up of five-layer models and multi-agent interactive model. Compared with the present models, the architecture has the technical advantages of opening, Agent compatibility, resource unified description and etc.

**Keywords** P2P, Mobile-Agent, Web service, Migration mechanism

## 1 引言

P2P 模式弱化了服务器的概念,使任意两台机器可以对等的方式对话,继而人们提出了 Web Services, 网格计算等概念重新诠释了下一代的互联网。不管是网格服务还是 Web 服务,虽然其自身都存在不同的内涵和发展趋向,但是“Service”即服务的概念已经被大多数研究者和企业所接受。一些世界顶尖的研发小组和知名企业开发出了一系列服务提供平台,并开始在互联网中显示出了威力,P2P 环境下的 Web 服务平台正是研究的热点之一<sup>[1-3]</sup>。

然而,P2P 并不是万能的,在研发过程中,人们发现它其实伴随着许多先天的不足<sup>[4]</sup>,例如网络管理困难,垃圾信息难以处理等等,这些难题也一度引起了一些学者对于 P2P 技术的怀疑。

移动 Agent 可以解决和优化原有 P2P 网络中的许多方面的问题:在 P2P 网络中,资源查询会产生巨大的通信流。而大部分查询是冗余的。移动 Agent 却能移动到每个节点上,通过本地化的运行来减少这些情况的发生。移动 Agent 存储它需要的所有数据<sup>[5]</sup>,当生成 Agent 的机器下线了,仍然能继续执行搜索任务,搜索完后,携带结果返回原始节点,或等待原始节点再次上线。它可以巡行到原始节点不知道的节点上,发现更多的资源,而接受它的节点也可以得到它曾访问

过的其它节点的资源信息。但如果不想或不能容纳它,也可以拒绝接受。移动 Agent 可以通过克隆在网络的不同方向上分派,并行运行,从而可以更快地发现资源并提高容错性<sup>[6]</sup>。2003 年,美国芝加哥大学的 Gianluca Moro 等专家深入分析评价了 Agent 引入 P2P 平台的机遇和挑战,证明了这个领域具有极大的研究价值<sup>[7]</sup>。

Agent-based P2P 的研究随着 P2P 网络技术的发展而不断受到重视,许多研究机构对此进行了深入的研究。复旦大学的基于对等计算的信息检索系统:PeerIS<sup>[8]</sup>,在 P2P 环境下成功引入了移动 Agent 的概念,获得了高效的访问效率和精确的查询结果;武汉大学和香港大学提出了“基于对等网络移动 Agent 技术的 Agent-P2P 框架”,该框架是一个开放式的平台,它与实现它的语言和操作系统无关,而是通过提供抽象的接口来实现。该平台中可以有多个软件 Agent 执行环境,执行环境可以采用不同的语言实现,相互之间可以通信;不同主机之间的 Agent 也可以通信,同时支持 Agent 的移动;国外对于 Agent-based P2P 的研究也非常重视,早在 2001 年,日本的 Kyushu University,英国 University of Southampton 等许多院校已经开发出了比较成熟的 Agent-based P2P 系统;此外,浙江大学、东南大学、中国科学技术大学也在这些领域有着长期深入的研究和积累。这些丰富的经验和积累进一步加速了这个领域的研究步伐。

<sup>\*</sup>国家自然科学基金项目(60573131),国家重点基础研究发展规划项目(2006CB303004)。夏 阳 博士研究生,副教授,硕士生导师,研究方向为网络计算、电子商务;陈贵海 博士,教授,博士生导师,研究方向为网络计算、组合数学等;张 强 硕士研究生,研究方向为 P2P 与 Agent 技术;徐 钊 教授,博士生导师,研究方向为通信与信息系统、光纤接入网技术。

然而,在长期的研究过程中,网络上已经存在大量的移动 Agent,并且在特定的领域上发挥着作用,由于缺乏一种统一的规范性标准,使得 Agent 之间无法进行很好的通讯,更无法实现服务的兼容性和开放性,业务增值服务和协同工作更是难以实现<sup>[10]</sup>。为了解决上述问题,在深入研究相关技术的基础上,本文给出了一种开放式服务体系架构 OSA-ABP。

## 2 OSA-ABP 架构

### 2.1 设计目标

Agent-based P2P 环境下开放式服务平台架构的设计目标主要着眼于处理如下几个方面的内容:

(1)环境的开放性。架构需要提供一种开放的模型标准,使其能够最大限度地兼容当前 Internet 环境下的漫游的移动 Agent,并能够使其开展协同服务。

(2)服务资源的动态性。应当允许所有的资源在授权和使用权的问题上具有高度的自主性,Internet 上所有资源都可以随意加入或退出该服务架构。

(3)资源使用的高效性。模型建立在对等网络的基础上,由于多 Agent 智能体的参与,资源深度发现以及资源的可靠性是关乎系统性能的一个重要指标。

(4)通讯的跨平台性。这需要解决多协议族支持下的 Agent 通讯问题,以及应当具备异构平台运作和跨语言交流的能力。

(5)资源高可靠性。建立资源监听和安全控制策略,保证资源的可靠性和安全性;整个体系模型建立在多个 Agent 组件的基础之上,实现其语言无关和系统无关性也是本系统开放性研究的重点。

### 2.2 OSA-ABP 分层体系架构

在面向对象的方法中将所有的内容都看成对象,同样在 Agent 系统中可以把所有的应用都当成 Agent 来对待,由于 Agent 系统中必须提供详细的资源访问控制权限,一般通过设置代理来实现,由于代理自动完成对资源访问权限的控制,也可以认为它们是一个 Agent,这样所有的资源也是通过 Agent 来进行处理,在 OSA-ABP 架构中,同样也是将系统中的应用组件都认为是 Agent。

OSA-ABP 结构的 5 层结构模型从下到上分为通讯层、资源层、代理环境层、特殊类层以及用户接口层,如图 1 所示。

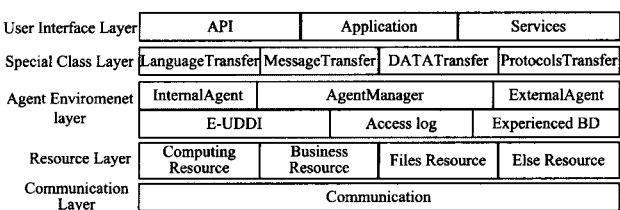


图 1 OSA-ABP 五层结构图

(1)通讯层:P2P 的一个重要特征是网络中节点可以随意退出和加入网络,这使得节点在通讯时需要更为复杂的处理策略,而且网络中瞬时存在很多浪费的、不实用的建立连接和关闭连接的信息,容易引发阻塞。因此在通讯时,特别是当网络拓扑发生变化时通讯层应当及时了解网络情况。

(2)资源层:这是 OSA-ABP 系统的一个重要组成部分,这里包含了对等网络所能提供的所有资源,如商务资源、计算资源、文档资源等等。在 OSA-ABP 平台下,资源可以粗略地划分为两大类:独立资源和交互资源。

(3)代理环境层:代理环境层是 OSA-ABP 体系架构的核心,它主要由 Agent 实体和目录资源组成,是整个架构的智能体部分。其中 Agent 实体部分又分为内部 Agent、外部 Agent 和 Agent 管理器 3 大部分,如图 2 所示。

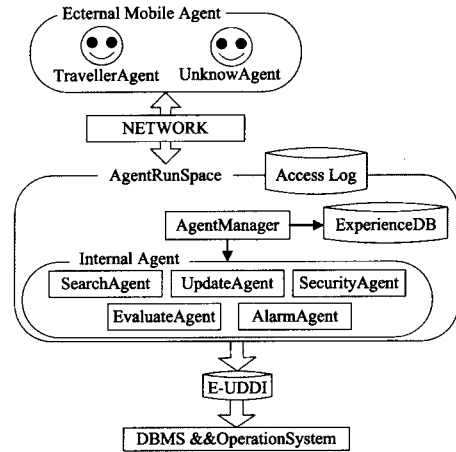


图 2 代理环境层结构模型

模型中各个部分被灵活地设计成彼此相互独立的移动智能 Agent,并以松散耦合的形式相互对立地存在,由 Agent-Manager 统一调度。在这里统称为智能服务中间件。

(4)特定类层主要功能包括:语言转换、消息转换、数据转换和协议转换。负责异构环境下的 Agent 的通讯问题。

(5)用户接口层:处于节点的最高层,可以从工具、应用和服务几个方面来实现。实际针对各种领域情况进行应用拓展,提供友好的人机交互界面进行操作,并根据不同的终端进行适配。另外一个重要的作用就是通过用户接口层结合现有技术,可以实现资源服务与通讯服务的融合。

### 2.3 OSA-ABP 空间协作模型

#### 2.3.1 空间模型设计

根据系统架构的设计要求,对于代理环境层,OSA-ABP 针对性地给出了以 Agent-Manager 为中心的空间协作模型。该模型的结构如图 3 所示。

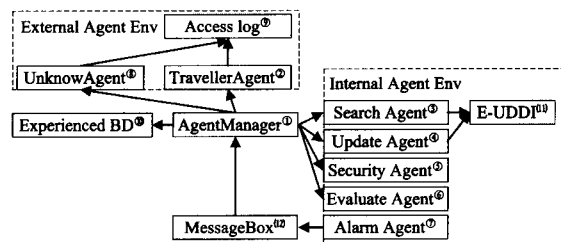


图 3 多 Agent 空间协作模型图

AgentManager 是 Agent 环境层的核心,产生 Agent 运行空间,即 AgentRunSpace,所有的外部 Agent(与本系统同构的 TravellerAgent 或异构的未知移动 Agent)只有连接到 AgentManager 后才能进入 AgentRunSpace,继而允许调用其他的功能 Agent。在 AgentRunSpace 资源访问完毕后,AgentManager 将负责释放或销毁各种 Agent 实体,同时释放 AgentRunSpace 空间。

Agent 管理器的任务主要包括下列一些基本任务:(1)创建旅行者 Agent: createTA(),(2)登记移动 Agent: Register

(object o), (3)命名 Agent: nameTA(), (4)分配任务列表: createTask(), (5)创建代理空间: createRunSpace(), (6)激活代理线程 activeAP(), (7)资源回收 destoryRunSpace(), (8)消息发送: sendMessage()。下面以 AgnetManager 类作为例子给出其接口描述:

```
public class AgentManager extends Aglet {
    Register register = null;
    public void onCreation(Object o) {}
    ////////////////for Agent//////////////////////
    //create Traveller and give it the taskList
    public void createTA() {}
    //give the name and uniqueID of travellerAgent
    public void nameTA() {}
    //create the agentRunSpace and accept the external agent
    public void CreateARC() {}
    //actve the internalAgnat agent in agentRunSpace
    public void activeInternalAgent() {}
    public void sendData(String p0) {}
    //registe and unregiste to peerNode
    public void register(object o) {}
    public void unregister(object o) {}
    public void destoryRunSpace() {}
    public void releaseAgent() {}
    //give the suggestion of nextHo Pof TA
    public String NodeRecommment() {}
    ////////////////for NetWork//////////////////////
    //register and unregister Node to E-UDDI
    public void register(PeerID p, E-UDDI e) {}
    public void unregister(PeerID p, E-UDDI e) {}
    //register and unregister Resource to E-UDDI
    public void register(Resource p, E-UDDI e) {}
    public void unregister(Resource p, E-UDDI e) {}
}
```

### 2.3.2 AgentRunSpace 设计

为了充分考虑 Agent 的智能性以及其对等空间模型的分布性和对等性, Agent 的调度和资源的服务采取了空间分层的概念:

(1) AgentRunSpace 的搜索是递进式搜索模式。在本地环境中, 充分利用本地资源进行深度搜索, 如不能满足服务条件, 则转向本虚拟社区环境进行搜索, 最后转向整个网络进行搜索, 如图 4 所示。

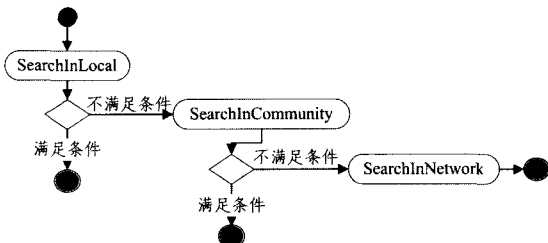


图 4 递进式搜索模式

(2) AgentRunSpace 是可扩展的环境, 允许 AgentRunSpace 在本地资源不足的条件下进行空间扩展, 其扩展的外在表现是 TravellerAgent 副本的迁移。

(3) AgentRunSpace 是受监听的空间。每个 Agent 在被激活或被创建后, 在生存状态下都必须接受 AgentManager 的监听和控制, 其消息通过 MessageBox 与 AgentManager 产

生交互。

### 2.3.3 E-UDDI 设计

OSA-ABP 系统架构的信息检索使用了基于 Web 服务的分布式信息检索, 在此基础上, 我们将所有的资源统一包装成 Web 服务的形式发布, 但其发布形式则充分考虑了 Agent-based P2P 的自身特点, 进行了较大的改进, 提出了 E-UDDI 的概念。

E-UDDI(Extended UDDI): 拓展 UDDI, 我们完全使用 UDDI 的发布模式, 将所有的资源和服务统一规范成 Web 服务进行发布, 使用 SOAP 通道, 使用 WSDL 进行统一发布解释。与 UDDI 不同的是, 资源不仅仅局限于静态资源, 动态资源的地址和交互策略、交互协议都被发布到 E-UDDI 中, 允许资源在 AgentSpace 中重组。

E-UDDI 与 UDDI 的不同点:

(1)资源的覆盖面不同。UDDI 是 Internet 中所有 Web 服务的发布和管理机构。而 PeerNode 中的 EUDDI 主要是发布本地的服务, ServerNode 也可以发布本社区的 Web 服务;

(2)Web 服务的种类不同。UDDI 的 Web 服务多是发布的计算资源, 以商业应用为主。E-UDDI 中发布的是各种资源, 包括本地的硬件资源、计算资源、数据资源以及商务资源等;

(3)设定的数目不同。在 Internet 中一般追求 UDDI 的唯一性, 例如 Microsoft 的 UDDI 在一个网络环境中只设计一个, 目的用于集中该网络环境中的服务<sup>[13]</sup>; 而 E-UDDI 在每个网络节点都存在;

(4)设计的目的不同。UDDI 目的在于整合所有 Web 服务的资源以便利用; E-UDDI 目的在于本地资源的集中检索和 M-Agnat 跨平台的访问。

(5)服务信息格式不同。E-UDDI 中必须包含资源的具体网络地址和资源存储信息, 而这些在 UDDI 的 Web 服务中不作要求。

## 3 模型评价与性能分析

### 3.1 与相关模型比较

OSA-ABP 整合了现阶段 P2P 系统的所有基本服务类型, 其架构模型还采取了独特的 3 层网络模型, 这种架构下可能提供的增值服务以及可能带来的潜在价值是不言而喻的。将 OSA-ABP 系统与现存的服务提供解决方案进行几个方面的比较, 结果如表 1 所示。

表 1 服务提供模型架构性能比较

系统评价条件	C/S	Web Service	OSA-ABP
响应速度	一般	低	较高
搜索效率	低	一般	较高
数据可靠性	低	较高	高
安全性	一般	一般	较高
开放性	一般	较高	高
独立性	一般	较高	高
通讯代价	一般	较低	高
查全率	低	一般	较高
用户可操作性	一般	较高	高
服务能力	低	较高	高
协议兼容性	低	一般	高
异构平台兼容性	低	高	高
健壮性和容错性	低	一般	高
用户的监督力度	较高	低	较高
组服务能力	无	无	有

### 3.2 系统架构的测评

5 层结构模型的设计充分考虑了系统的扩展性和外延的  
(下转第 68 页)

案, MRCM 为本文 MRCM 通信方案。在进行方案比较时, 在网络中传输的数据进行了优化、压缩。如图 6 所示, 由于 MRCM 具有动态的规划能力, 因此其网络代价要比全远程通信方案节省约 42% 的流量, 比全 MA 方案节省近 28% 的流量。

**结束语** 本文提出了一种 MAS 系统动态选择协商通信方式的模型, 该模型在计算所需网络代价的基础上, 平衡目标服务器负载, 引入预测机制在通信前预测网络代价及服务器负载, 动态选择相应的通信方式(远程通信或 MA 迁移), 并且采用黑板结构实现 agent 对预测结果的共享。实验表明该模型对于网络代价、服务器负载的预测近似准确, 便于模型在实际中的应用。

本文在实现预测数据的共享时采用一个静态的 agent 作为黑板 agent, 虽然通信双方的数据为负载信息, 信息量不大且大小固定, 但仍然会给 MAS 系统带来额外的通信负担, 后续工作将就如何提高预测数据的共享率展开研究工作。

### 参 考 文 献

[1] Straßer M, Schwehm M. A performance model for mobile agent systems // H. R. Arabnia, ed. Proceedings of the International

Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '97). Las Vegas (USA), CSREA Press, 1997, 2: 1132-1140

[2] China T, Kannapan S. Strategically mobile agents // KRothermeled[R]. Berlin: Springer, 1997: 149-161

[3] Schlegel T, Braun P, Kowalczyk R. Towards Autonomous Mobile Agents with Emergent Migration Behaviour // AAMAS06. 2006: 585-592

[4] Arthur W B. Inductive Reasoning and Bounded Rationality. American Economic Review (Papers and Proceedings), 1994, 84 (2): 406-411

[5] 尚丽辉, 汪小帆. 一类拥塞问题研究综述. 控制与决策, 2004, 11: 1202-1207

[6] 王汝传, 徐小龙, 黄海平. 智能 agent 及其在信息网络中的应用. 北京邮电大学出版社, 2006

[7] Wooldridge M. 多 agent 系统引论. 北京电子工业出版社, 2001

[8] 韩伟, 韩忠愿. 基于黑板模型的多智能体合作学习. 计算机工程, 2007, 11: 44-47

[9] 刘群, 夏虹. 多源信息融合系统的主动式黑板结构与模型. 哈尔滨工程大学学报, 2007, 8: 910-916

(上接第 64 页)

多样性, 最大程度地将系统模型在逻辑上划分成粒度较小但相互独立的分层架构。这样的设计至少有如下几个优势:

(1) 通讯的跨平台性。数以百计的系统之间, 特别是未知移动 Agent 如何产生协作, 必须解决通讯模式的翻译问题。特殊类层的设计使这样复杂而庞大的工作成为可能。

(2) 提高高性能节点的利用率。实践证明, 网络中的各个 Peer 在实用性、带宽、处理能力等方面存在着明显的差异, 如果让所有的 Peer 地位平等, 在分派任务时同等对待, 那么将造成资源的浪费, 甚至会产生瓶颈, 危害整个网络的性能。用那些高性能 Peer 形成网络的支柱, 这样有能力的 Peer 就能够更好地分散网络的负担, 分配任务的时候根据 Peer 性能的不同来分配适当的任务, 这样就使资源的利用率更高, 做到物尽其用。

(3) 服务资源的高扩展性。由于系统中提出了 EUDDI 的概念, 这将允许我们使用 Web 服务的方式发布任何种类的服务, 当然也包括计算资源和数据资源。

(4) 模型的易维护性。即使是通讯协议的补充, 数据样式的变化, 甚而至于自定义通讯这样复杂的问题, 也无需停止庞大系统的运行, 只需对相应层次进行修改。

**结束语** 本文提出的开放式服务体系架构合理地融合了包括 Web 服务技术在内的多项技术, 将计算资源、Web 资源、数据库资源等统一包装成 Web 服务, 在 Agent-based P2P 环境下搭建了开放式服务发布、资源发现和共享模型, 该模型在服务架构的分层管理、多 Agent 合作等研究方面取得了一定成果。

### 参 考 文 献

[1] Lu T, Fu M. Using Mobile Agents for Object Sharing in P2P Networks [A] // First International Conference on Innovative Computing Information and Control [C]. 2006, 8(1): 741-744

[2] Ramachandran G, Hart D. A P2P intrusion detection system

based on mobile agents [A] // Proceedings of the 42nd Annual Southeast Regional Conference [C]. 2004, 4: 1029

[3] Baek J-W, Yeom H Y. An Approach to Provide Mobile Agents with Planning Capability for P2P Environment [A] // 11th International Conference on Parallel and Distributed Systems [C]. 2005, 7: 730-736

[4] 冯新宇, 吕建, 曹建农. 通用的移动 Agent 通信框架设计 [J]. 软件学报, 2003, 14(5): 984-990

[5] 徐非, 杨广文, 鞠大鹏. 基于 Peer-to-Peer 的分布式存储系统的设计 [J]. 软件学报, 2004, 15(2): 268-277

[6] 陶先平. 基于 Internet 的移动 agent 技术和应用研究 [D]. 博士学位论文. 南京大学计算机科学与技术系, 2001

[7] 夏启志, 谢高岗, 闵应骅, 等. IS-P2P: 一种基于索引的结构化 P2P 网络模型 [J]. 计算机学报, 2006, 29(4): 602-609

[8] 林波. 基于对等计算的信息检索技术 [D]. 博士学位论文. 复旦大学信息科学与工程学院, 2004

[9] Bisignano M, Modica G D, Tomarchio O. Mobile agent location management; a comparison between CORBA and P2P based systems [A] // Eighth IEEE Symposium on Computers and Communications [C]. 2003, 6: 1029

[10] Pang Xiao-lin, Catania B, Tan K-L. Securing Your Data in Agent-Based P2P Systems [A] // Eighth International Conference on Database Systems for Advanced Applications [C]. 2003, 5: 55

[11] Purandare D, Guha R. Preferential and Strata based P2P Model: Selfishness to Altruism and Fairness [A] // 12th International Conference on Parallel and Distributed Systems [C]. 2006, 7(1): 561-570

[12] Wang Dan. A resource discovery model based on multi-agent technology in P2P system Proceedings [A] // IEEE/WIC/ACM International Conference on Intelligent Agent Technology [C]. 2004: 548-551

[13] Cooper B, Hector G-M. Peer-to-peer resource trading in a reliable distributed system [M] // 1st International Workshop on Peer-To-Peer Systems. 2002(2429): 319-327