

移动 Agent 互操作性研究^{*}

彭德巍 何炎祥

(武汉大学软件工程国家重点实验室 武汉430072)

摘要 移动 agent 互操作的问题阻碍了移动 agent 系统的广泛应用,本文主要阐述如何解决该问题,通过深入分析与移动 agent 系统互操作性有关的 CORBA、MASIF 和 FIPA 规范,系统地提出了移动 agent 技术与三者结合实现互操作性的可能性和必然性,并且分析了互操作的四个层次,最后给出了实现该互操作性移动 agent 平台的方法并与现有系统进行了比较。

关键词 移动 agent, 互操作性, CORBA, MASIF, FIPA

Study of Interoperability in Mobile Agent Environment

PENG De-Wei HE Yan-Xiang

(National Key lab of Software Engineering, Wuhan University, Wuhan 430072)

Abstract The interoperability of mobile agent platform prevents the mobile agent system to be applied widely. The paper mainly represents how to solve the problem. OMG CORBA, MASIF and FIPA standards are closely analysed here, then are combined to guarantee the interoperability of Mobile Agent System. We present four layer of interoperability in our system and the method to implement it and compare our system to the existing system.

Keywords Mobile agent, Interoperability, CORBA, MASIF, FIPA

1 介绍

移动 Agent 技术为分布开放系统的分析、设计和实现提供了一种崭新的方法,被誉为是“软件开发的又一重大突破”,其技术特点和优势在于:动态执行、异步计算、并行求解、智能化路由等。通过业务请求 Agent 动态移动到服务器端执行,使得 Agent 不经过网络传输这一环节而直接访问服务器资源,从而避免了大量数据在网络中传输,降低了系统对带宽的依赖,减少了网络冲突,在低带宽、非稳定连接的网络环境下依然能够保持稳定的工作^[1]。

但从目前的研究情况看,移动 Agent 本身的应用并未如预期的那样理想。诸多的问题阻碍了基于 Agent 系统的广泛应用,其中安全性问题,集成问题,互操作问题是最大的影响。目前,商业化和实验性的移动 Agent 系统多达70多个,这些系统在体系结构和系统实现上都存在着较大的差异,严重阻碍了移动 Agent 系统的互操作和移动 Agent 技术的推广。标准的建立是移动 Agent 技术走向成熟的前提,也是目前解决互操作问题主要方法。其中最新、而且比较有影响的是“移动 Agent 系统互操作草案”MASIF (Mobile Agent System Interoperability Facility)。FIPA 组织也在致力于 Agent 技术互操作

规范的制定和推广。

2 移动 Agent 和 CORBA

移动 Agent 可以在执行期间从一个主机移动到另一个主机,并在目的地继续执行,它可以动态地选择在何时移动或将移动到何地,并且以异步方式返回结果。这样就有效地处理网络连接不可靠、带宽不够或网络不可靠中断时发生的情况。此外,一些代理也可以协同提供分布式和协作业务^[2,3]。

移动 Agent 技术所提供方案的优势体现在多种应用领域,例如对网络、系统和业务的管理、移动计算、电子商务,以及智能信息的恢复和过滤等。在智能网络或主动网络领域中,移动 Agent 与主动网络或智能网络技术的结合使得基于移动性的主动实体可以动态向所需网络中嵌入应用和用户明确的行为。此外,移动 Agent 技术在电子商务和入侵检测等方面都体现出很大的优势^[4]。

CORBA 技术^[5]和移动 Agent 技术在许多方面不同,最显著的是 CORBA 采用对象一旦被分配就一直停留在固定位置直到它们在 ORB 中注册,而移动 Agent 可以动态、自主地在执行期间移动。CORBA 拥有一个普遍被接受的规范,有许多的资源,系统和业务组件遵从这个规范。相反,各种不同

^{*} 基金项目:软件工程国家重点实验室开放基金资助(项目编号:20020208)。何炎祥 教授,博士生导师,计算机学院院长,现从事分布并行处理、Agent 技术,数据开采等方面的研究。彭德巍 博士生,现从事分布并行处理、Agent 技术方面的研究。

的、不能互操作的 Agent 平台使我们考虑将 CORBA 引入移动 Agent 系统,通过二者的结合来解决互操作问题,其中包括不同移动 Agent 平台之间的互操作,移动 Agent 平台与遗留系统(legacy System 非移动 Agent 系统)的互操作问题。

3 移动 Agent 平台的互操作性

移动 Agent 互操作性问题并没有得到足够重视,在如何实现互操作性问题上没有明确的定义,而且在一些移动 Agent 系统的上下文中也没有清晰地体现出来,这一状况在 OMG 提出了 MASIF 规范后,得到一定程度的改善。

我们不能简单认为移动 Agent 在不同移动 Agent 平台之间移动就实现了互操作性。因为互操作性有更广泛的涵义,它能够完全实现异构系统之间的交互,而不管系统是否是基于移动 Agent 技术的。用户到达移动 Agent 系统的应用层,不仅可以通过系统明确接口如 ATP 到达同一移动 Agent 平台的应用层,还可以通过其它不同的协议与接口如 HTTP 协议到达。

3.1 MASIF

OMG 的 MASIF^[6]标准建议对 Agent 管理、Agent 迁移、Agent 名称、Agent 系统名称、Agent 系统类型以及位置语法进行了标准化。标准定义了 MAFAgentSystem 和 MAFFinder 接口。其中 MAFAgentSystem 负责接收 Agent、列出 Agent、获得 MAFFinder 接口、获得 Agent 系统类型、获得 Agent 状态等,MAFFinder 提供注册、注销、查询等服务,实际上是一种名字服务。MASIF 的目标是不需要对现有各种 Agent 平台进行大量的修改,而仅通过用“add-on”模块来扩展,以实现现存不同结构的移动 Agent 平台之间以及移动 Agent 平台与遗留系统的互操作性。

3.2 FIPA

FIPA^[7]是由活跃于 Agent 领域的大学和公司组成的非赢利组织,FIPA 把重点放在 Agent 间通信的标准化问题上,FIPA 提出一个专门的 Agent 通信语言(ACL)来解决遵从 FIPA 标准的所有 Agent 之间的通信问题。FIPA 定义了为 Agent 平台服务的三种基本业务:Agent 管理系统(AMS)、目录服务设施(DF)和 Agent 通信通道(ACC)。AMS 提供类似 MAFAgentSystem 的管理功能,但不能解决在异构移动 Agent 平台之间 Agent 的移动问题。DF 提供类似于 MAFFinder 的注册查询等功能。Agent 通信通道提供了平台间的 Agent 通信。

4 实现互操作性的移动 Agent 平台模型

4.1 具有互操作性的移动 Agent 平台的结构

具有互操作性的移动 Agent 平台通常构建于各种异构的分布式系统之上。这些异构的分布式系统经常会提出各种复杂的以网络为中心业务要求,这使得移动 Agent 平台必须更加灵活,并具备可扩展性。

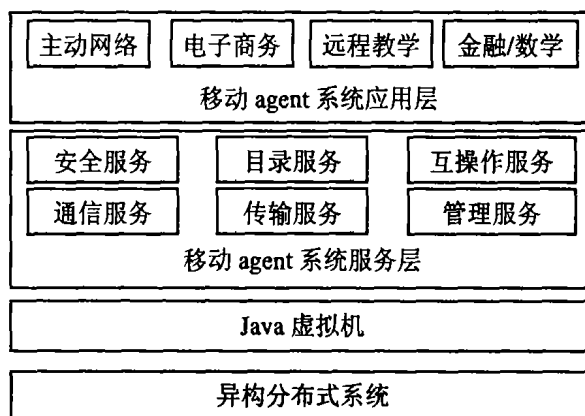


图1 具有互操作性的移动 Agent 平台的功能层次结构

图1描述了具有互操作性的移动 Agent 平台的功能层次结构由两个业务层组成,应用层和服务层,其中包括:通信服务、传输服务、管理服务、安全服务、目录服务、互操作服务。其中互操作服务允许移动 Agent 系统的移动 Agent 与现存软件或硬件组件通过与 CORBA/IIOP 规范相互协同作用来实现。移动 Agent 系统实现 MASIF 接口和 FIPA 接口,允许移动 Agent 系统与其它遵从 MASIF 规范或者 FIPA 规范的移动 Agent 平台相互作用。

4.2 移动 Agent 平台的互操作层次

移动 Agent 平台的互操作分为以下四个层次:

- ① MAS Agent 可以调用外部 CORBA 对象;
- ② MAS Agent 可以给一个 ORB 提供接口;
- ③ 任何外部实体可以通过标准 MASIF 接口访问 MAS;
- ④ MAS Agent 可以通过 FIPA ACC 发送业务与任何一个移动 Agent 平台进行通信。

前两个层次通过 CORBA C/S 扩展移动 Agent 平台可以实现。Agent 既可以作为 CORBA 客户,也可以 CORBA 服务器的方式来注册,以便为互操作性移动 Agent 系统之外的应用提供访问点。

第三个层次,互操作移动 Agent 平台通过 MASIF 规范来实现,通过 MAFAgentSystem 接口来控制遵从 MASIF 规范的移动 Agent 系统中的 Agent。MASIF 定义方法用来挂起/恢复/终止 Agent,或使 Agent 从一个系统移动到另一个系统。Agent 的跟踪功能允许 Agent 在 MASIF 中注册,提供一个移动 Agent 的命名业务。关于第四个层次,与 CORBA 的结合对遵从 FIPA 的移动 Agent 平台来说是必需的,Agent 的消息传送通过 CORBA/IIOP 实现。

4.3 互操作性的实现

互操作性移动 Agent 平台通过三个模块来实现上节提出的四种互操作层次。第一个是 CORBA C/S 模块,第二个是 MASIF 模块,第三个是 FIPA 模块。因为 MASIF 不支持 Agent 之间的通信,而 Agent 通信信道提供了异构 Agent 通信的互操作

性,所以互操作移动 Agent 平台主要集中于 FIPA/ACC 的实现。Agent 管理系统和目录服务功能的实现可以映射到 Agent 管理和注册等相关功能,这些可以在互操作移动 Agent 平台的 MAFAgentSystem 和 MAFFinder 接口中得到。

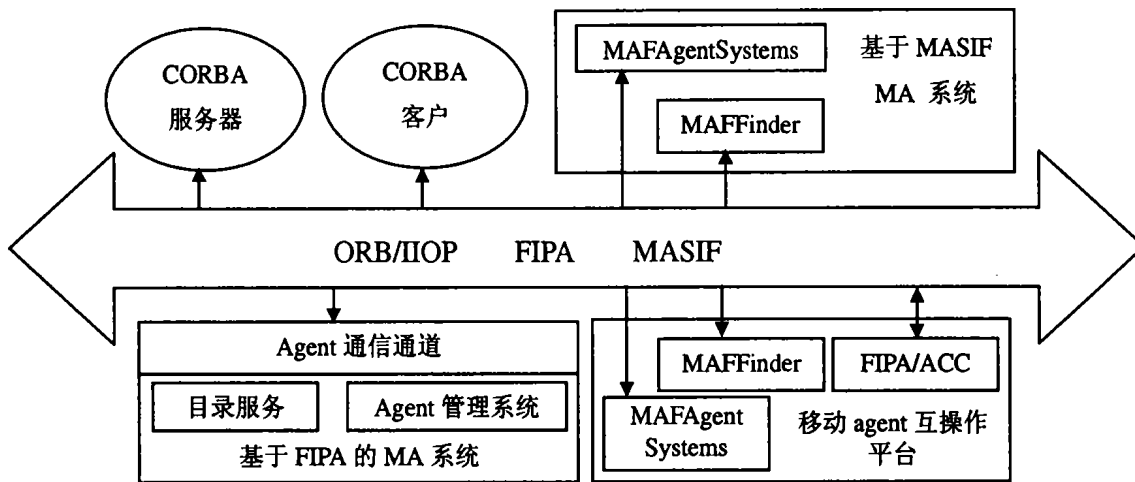


图2 基于 CORBA 的互操作移动 Agent 平台

Java 是一种跨平台,适合分布式计算环境的面向对象编程语言。CORBA 和 Java 是以一种合作的方式相互融合的。前者屏蔽了网络的细节获得网络透明性,而后者通过 Java 虚拟机获得了实现上的透明性。

相关研究 IBM 提供的 Aglets^[8]系统的通信 API 就源自于 MASIF,但它缺省的实现方式被称作 Aglets 传输协议(ATP)。此接口不能实现异构 Agent 之间的交互以及和遗留系统的交互。一些移动 Agent 系统已经接近实现与其它系统和应用之间的互操作性问题。JumpingBeans 曾提出了“mobile Agent CORBA”的思想^[9],但其本质仍是移动 Agent 支持系统,很少涉及 CORBA 特有的研究领域。

Voyager^[10]通过实现特殊的 ORB 机制(Voyager ORB)与分布式处理计算紧密结合。但是,Voyager 仅能够实现互操作层次中的前两个层次,而不能实现互操作层次中的后两个层次。SOMA^[11,12]实现了遵从 MASIF 和 FIPA/ACC 的类库。并且实现了 SOMA 2.0 版本和 Grasshoper 1.0 版本的互操作性。但它在可扩展性上还显得不足,另外它不支持移动 Agent 作为 CORBA 服务器进行注册,也不支持 http 协议的交互,本文在这方面都做了改进。

参考文献

- 1 Pham K A. Mobile software Agents : An overview[J]. IEEE Commun.1998,36(7)
- 2 何炎祥,陈莘萌. Agent 和多 Agent 系统的设计与应用. 武汉大学出版社,2001
- 3 史忠植. 智能主体及其应用. 科学出版社,2000
- 4 张云勇,等. 移动 Agent 技术. 清华大学出版社,2003
- 5 Object Management Group. The Common Object Request Broker: Architecture and Specification. <http://www.omg.org/corba>
- 6 Mobile Agent System Interoperability Facilities <http://www.omg.org>
- 7 FIPA 2000 Specifications, Foundation of Intelligent Physical Agents. <http://www.fipa.org>
- 8 Aglets Software Development Kit. <http://www.trl.ibm.com.jp/aglets>
- 9 Astra A. Engineering-Jumping Beans <http://www.jumpingbeans.com>
- 10 Recursion Software Voyager Voyager Tutorial. <http://www.recursionsw.com/>
- 11 Bellavista P, Corradi A, Stefanelli C. Middleware services for interoperability in open mobile Agent systems. Microprocessors and Microsystems, 2001, 25: 75~83
- 12 Bellavista P, Corradi A, Stefanelli C. CORBA Solutions for Interoperability in Mobile Agent Environment [M]. Antwerp, Belgium: IEEE Computer Society Press, Sep. 2000