基于JADE平台的多Agent系统开发技术

JIYU JADE PINGTAI
DE DUO AGENT XITONG KAIFA JISHU

于卫红 著　陈燕 主审

国防工业出版社
National Defense Industry Press

基于 JADE 平台的多 Agent 系统开发技术

于卫红 著
陈 燕 主审

国防工业出版社
·北京·

图书在版编目(CIP)数据

基于 JADE 平台的多 Agent 系统开发技术/于卫红著. —北京:国防工业出版社,2011.10
ISBN 978-7-118-07695-0

Ⅰ.①基... Ⅱ.①于... Ⅲ.①软件工具-程序设计 Ⅳ.①TP311.56

中国版本图书馆 CIP 数据核字(2011)第 190700 号

※

国防工业出版社出版发行
(北京市海淀区紫竹院南路 23 号 邮政编码 100048)
天利华印刷装订有限公司印刷
新华书店经售
*
开本 850×1168 1/32 印张 4⅞ 字数 126 千字
2011 年 10 月第 1 版第 1 次印刷 印数 1—3000 册 定价 28.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422 发行邮购:(010)68414474
发行传真:(010)68411535 发行业务:(010)68472764

前　言

在计算机和人工智能领域，Agent 可以看作是一个实体，它驻留在某一环境下能够自主、灵活地执行动作以满足设计目标。Agent通过传感器感知环境，通过效应器作用于环境。多 Agent 系统主要研究在逻辑上或物理上分离的多个 Agent，协调其智能行为，即知识、目标、意图及规划等，实现问题求解。

随着理论和应用研究的不断深入，目前，Agent 和多 Agent 系统已经成为计算机科学领域、信息工程领域、网络与通信领域、人工智能领域等的前沿研究方向之一。多 Agent 系统可以应用到很多方面，例如，在线拍卖协商、电力系统、智能交通系统、医疗诊断系统、远程教学系统等。

尽管多 Agent 系统的应用前景十分美好，但是由于理论上的复杂性，使得多 Agent 系统的开发困难重重。目前多 Agent 系统的开发平台有很多，如 JAFMAS、Aglets、Zeus 等，它们在功能实现上具有很大的局限性，在灵活性、实用性、扩展性等方面也存在很多不足之处。此外，这些开发平台的侧重点也有所不同，如有的侧重于 Agent 通信基础设施的搭建，有的则侧重于 Agent 内在含义的表现。

JADE 的出现为多 Agent 系统的开发注入了新的生机和活力。JADE 是用 Java 语言编写的，为了便于开发多 Agent 系统，JADE 提

供了两个重要的组成部分:遵循 FIPA 规范的 Agent 开发平台和开发 Agent 的软件包。JADE Agent 开发平台为分布式多 Agent 应用提供了最基本的服务和基础设施①Agent 生命周期管理和 Agent 移动性;②白页服务和黄页服务;③点对点的消息传输服务;④Agent的安全性管理;⑤Agent 的多任务调度,等。

本书主要为在 JADE 平台上开发多 Agent 应用程序的开发者提供指导,帮助读者快速熟悉 JADE 平台的体系结构、开发方法,掌握多 Agent 系统开发所需的主要知识。本书共有 7 章,主要内容如下:第 1 章 多 Agent 系统与 JADE 平台,介绍了 JADE 平台的体系结构、基本理论以及使用 JADE 平台时的环境配置等。第 2 章 JADE 编程基础,从创建一个最简单的 JADE Agent 开始,逐步详细讲述 JADE 的编程机制,包括创建步骤、Agent 标识符、添加行为、与其他 Agent 通信等。第 3 章 Agent Behaviour 详解,通过详尽的实例介绍了 Agent Behaviour 的原理及使用方法,主要包括:OneShotBehaviour、CyclicBehaviour、TickerBehaviour、SequentialBehaviour、ParallelBehaviour、FSMBehaviour 等。第 4 章 Agent Communication 详解,介绍了 Agent 通信的基本原理,并通过实例讲述了远程机器上 Agent 间通信的实现、基于对象序列化机制的 Agent 间的通信以及消息模板等。第 5 章 JADE Agent 与 JSP/Servlet,主要介绍了 JADE Agent 与 JSP/Servlet 的集成,包括传统架构下的集成及基于 JADEGateway 的集成。第 6 章 JADE Agent 与 Ontology,在介绍 ontology 基本原理的基础上,讲述了基于 ontology 的 Agent 间通信的实现过程。第 7 章 JADE Agent 与 Web Service,介绍了 Web Service 的基本原理、JADE Agent 与 Web Service 集成的必要性及

此为试读,需要完整PDF请访问: www.ertongbook.com

集成方法，重点介绍了基于 WSIG 的 JADE Agent 与 Web Service 集成的实现方法。

本书在编写的过程中，得到了大连海事大学管理科学与工程学科各位同事的鼓励和帮助。大连海事大学管理科学与工程学科带头人陈燕教授在百忙之中为本书的编写提供了若干宝贵意见并承担了主审工作。谨在此表示衷心的感谢。

同时感谢国防工业出版社的工作人员，特别是王京涛同志对本书的关心与指导。

最后感谢家人的理解和支持，多少的失意彷徨、多少的凄风冷雨，在亲情与关爱面前，都化作乌有，剩下的只有前行的动力。

作者

2011 年 6 月

目　录

第1章　多Agent系统与JADE平台

1.1　迈进Agent新时代

无论面向 Agent 的开发方法是否会象宣传的那样继面向数据流、面向数据结构、面向对象之后成为新一代的软件开发主导方法，随着软件系统网络化、系统服务能力、交互能力要求的不断提高，在系统中引入分布式处理因素、通信因素以及智能因素等已经成为必然。

现阶段面向对象的方法是软件开发的主流。面向对象的开发机制虽有很多优越性，但却不能完全体现现实世界的特点，难以实现自然建模，需经变换和抽象才能将现实世界映射为对象模型，难以很好地控制和简化系统的复杂度。面向对象技术尤其需要在主动对象和多线程技术的实现上加以丰富和完善。

20 世纪 80 年代中期以来，随着 Internet 的发展产生了新的计算模式——分布式计算技术，而传统人工智能与分布式计算日益融合逐步形成了一门新的学科——分布式人工智能。目前，分布式人工智能的研究重心主要是分布式问题求解和多 Agent 系统。

分布式问题求解采用“分而治之”的思想，由多个协作的、知识共享的问题求解器共同完成一项问题求解任务，因为，其中任何一个问题求解器都没有能力独立完成该项任务。多 Agent 系统则致力于研究一组自治的智能 Agent 的协作行为。虽然它们的求解风格不同，但都引入了一种软件实体：Agent。

Agent 并不是一个新概念，1977 年，Hewitt 提出了“演员”

（自包容的、交发执行的对象）的概念，是 Agent 的雏形。经过二十几年的发展，Agent 逐步成为人工智能（AI）及其他计算机领域内的一个重要研究课题。Wooldridge 和 Jennings 在 1995 年提出了目前较权威的 Agent 定义，获得了计算机领域专家的普遍认同。此定义包括以下两个子定义：

（1）弱定义：Agent 是一个基于软件（在较多的情况下）或硬件的计算机系统，它拥有以下特性：自治性、社会能力、反应性和能动性。

（2）强定义：Agent 在弱定义的特性基础上，还要包括情感人类的特性。

事实上，Agent 的概念涉及的范围很广。它可以被看作是一种程序，像机器人这样的硬件 Agent 也属于 Agent，但本书所涉及的只是软件 Agent，它们具有如下特性：

（1）Agent 可表示现实世界和计算机世界中的行为实体，能自主地代替用户执行特定的工作。

（2）Agent 是隶属于一个环境的系统。这里的环境是指操作系统、网络或多方博弈环境等。

（3）Agent 拥有知识库和推理能力，并通过与用户、资源或其他 Agent 进行信息交换和通信来解决问题。Agent 间通过高层的交互相互作用，这种交互在语义和知识层次上进行，区别于过程调用、函数调用以及对象之间的消息传递。

（4）Agent 自动认知环境的变化，并采取相应的行动，拥有经验学习的能力。

（5）Agent 是一个“主动”的对象，它不是根据外部的请求调用相应的方法进行响应，而是对接收到的外界刺激进行知识融合，通过自身心智状态的变化来控制动作规划，通过执行动作对外界刺激作出响应。

（6）Agent 采用多线程的实现方式，提高对大量并发的支持，Agent 的行动不是一次性而是持续的。

源自分布式人工智能的多 Agent 系统，其基本思想是把

此为试读，需要完整PDF请访问：www.ertongbook.com

各种问题求解方法封装到一个个具有自主性的 Agent 中，通过 Agent 间的交互进行协调、协作、协商共同完成问题求解任务。

在开发多 Agent 系统时，需要明确以下几点。

1）No standards, no agents

没有标准就没有多 Agent 系统。我们可以看到，在任何一个领域，都可以说是“得规范者得天下”。多 Agent 系统目前已引起了广泛的关注，网上搜索、网上购物、网络管理、网上拍卖等很多方面都需要 Agent，这些 Agent 可能来自不同的设计者、不同的提供者、不同的组织。标准是实现开放性的基础，为了确保异质的 Agent 间互联和互操作等性能的实现，必须制定一些标准规范。

FIPA (Foundation for Intelligent Physical Agents)是目前致力于 Agent 技术标准化工作的重要组织。FIPA 定义了一组规范，用来支持 Agent 间以及基于 Agent 的应用之间的互操作。只有开发遵循 Agent 技术标准的、支持基于 Agent 系统开发的平台，才能尽快开发出基于 Agent 的应用系统以发挥 Agent 技术的优势。

2）No software tools, no agents

没有软件工具，无法实现多 Agent 系统。目前，很多 Agent 研究仍停留在理论阶段，总给人华而不实、纸上谈兵的感觉。理论领悟透彻了，能在生活中解决实际问题才称得上有价值。

而当前 Agent 研究的一个重要问题就是缺乏开发环境和编程工具的支持，大多数基于 Agent 系统是利用非 Agent 技术来实现的，这意味着 Agent 技术还不成熟以及 Agent 技术还没有真正为广大计算机工作者所认可和接受；尽管研究者已经提出了许多面向 Agent 的程序设计语言，但它们都有很多局限性，实用性不强，无法为大众广泛接受和使用。

顺应 Agent 与多 Agent 系统理论研究与实际应用的需要，一个崭新的多 Agent 系统开发平台——JADE 诞生了。

1.2 初识 JADE

JADE (Java Agent Development Framework)是一个完全由Java语言编写的多Agent开发框架，遵循FIPA规范，提供了基本的命名服务、黄页服务、通信机制等，可以有效地与其他Java开发平台和技术集成，极大地简化了开发多Agent系统的各个环节。

为了更好地理解JADE，应该首先了解一下FIPA及FIPA规范。

1.2.1 FIPA及FIPA规范

FIPA（The Foundation for Intelligent Physical Agents）是一个由活跃在Agent领域的公司和学术机构组成的非赢利的国际组织，成立于1996年，其目标是为异质的Agent和Agent系统之间能够互操作而制订相关的软件标准。FIPA的宗旨是“促进基于Agent的应用，业务和设备的成功。”

FIPA规范从不同方面规定或建议了Agent在体系结构、通信能力、移动能力、知识表达能力、管理和安全等方面的内容，对于Agent技术起到了很大的推动作用。

根据FIPA的定义：Agent是一个实现应用系统自治性及通信功能的计算过程，Agent间的通信使用ACL（Agent Communication Language）语言。一个FIPA Agent具有下列特征：

（1）每个Agent都有一个名字；

（2）每个Agent都有一个用于通信目的的定位符；

（3）一个Agent可以向一个或多个Agent发送信息；

（4）Agent可以在一个目录服务中注册它的目录项；

（5）Agent可以在目录服务中修改或删除其注册目录项；

（6）Agent可以在目录中查询感兴趣的目录项。

FIPA定义了Agent平台应提供的若干服务，如图1-1所示。

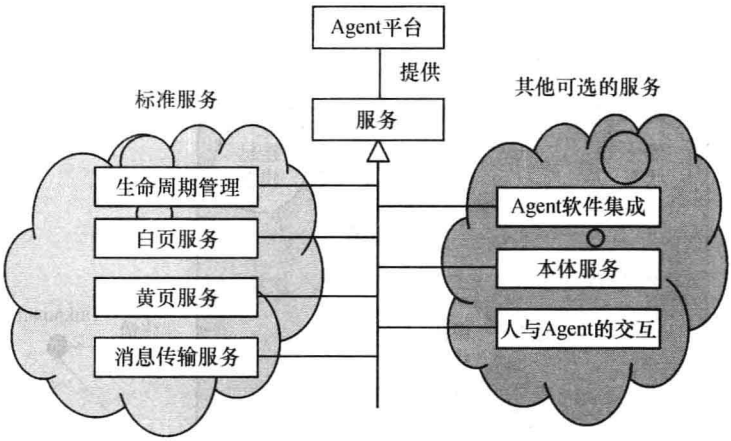

图 1-1　Agent 平台提供的服务

其中有三种最基本的服务：Agent 管理系统（Agent Management System，AMS）、目录服务（Directory Facilitator，DF）和消息传输服务（Message Transport Service，MTS）。基于 FIPA 规范的 Agent 管理参考模型，如图 1-2 所示。

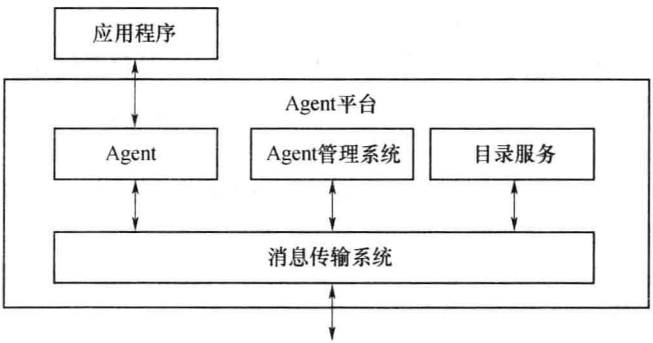

图 1-2　基于 FIPA 规范的 Agent 管理参考模型

每一个平台只能有一个 Agent 管理系统，负责 Agent 的命名、定位和控制服务。每个 Agent 必须在一个 Agent 管理系统中注册以便得到一个有效、唯一的 Agent 标识（AID）。Agent 管理系统

维护着 AID 目录，用于 Agent 生命周期管理。遵循 FIPA 规范，一个 Agent 的生命周期如图 1-3 所示。

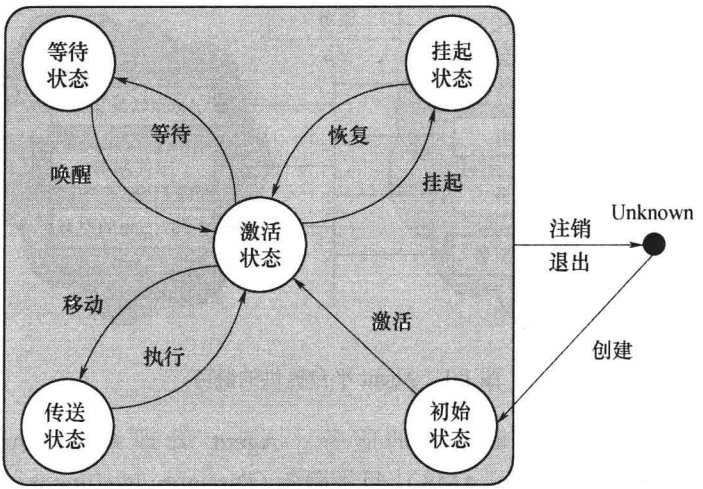

图 1-3　Agent 的生命周期

根据 FIPA 中的 Agent 平台生命周期，JADE Agent 可以处于这几个状态之一，它们在 Agent 类中用几个常量来表示。

（1）初始状态（AP_INITIATED）：Agent 对象已经建立，但是还没有由 AMS 注册,既没有名字，也没有地址,而且不能与其他 Agent 进行通信。

（2）激活状态（AP_ACTIVE）：Agent 对象已经由 AMS 注册，有正规的名字和地址，而且具有 JADE 的各种特性。

（3）挂起状态（AP_SUSPENDED）：Agent 对象当前被停止，内部的线程被挂起，没有 Agent 行为被执行。

（4）等待状态（AP_WAITING）：Agent 对象被阻塞，等待其他事件。内部的线程在 JAVA 监控器上休眠，当条件满足时被唤醒（典型的情形是消息到达）。

（5）删除状态（AP_DELETED）：Agent 死亡。内部线程的执行被终结，Agent 不再在 AMS 上有注册信息。

（6）传送状态（AP_TRANSIT）：移动 Agent 移动至一个新的位置时进入这个状态，系统继续缓存将被送到这个新位置的消息。

（7）复制状态（AP_COPY）：这是 JADE 在 Agent 克隆时的一个内部状态。

（8）离开状态（AP_GONE）：这是 JADE 在移动 Agent 移至一个新的地点时的一个内部稳定状态。

目录服务 DF 是 Agent 平台的必须部分，它提供平台内的黄页服务。

消息传输服务是默认的跨平台的 Agent 消息传输机制，提供了不同 Agent 之间的 ACL 消息交互机制。在消息传输机制中，ACC（Agent Communication Channel）是消息的传输通道，是 Agent 平台上为 Agent 提供消息交互的实体。MTP（Message Transport Protocol）是不同 ACC 之间的消息交互协议。

1.2.2 JADE 平台的体系结构

JADE 是一套免费开源的多 Agent 系统开发框架，提供了 Agent 赖以生存的运行时环境。JADE 平台体系架构如图 1-4 所示。

从图 1-4 可以看出在 JADE 平台中利用容器（container）容纳 Agent，一个平台（platform）可以有多个容器，一个容器可容纳多个 Agent。容器可以位于不同的主机上，在一个 JADE 平台中，有且仅有一个称为主容器的容器，其他容器启动时，都必须在主容器中注册。在图 1-4 所示的网络中，存在两个不同的 JADE 平台，其中一个由三个容器构成；另一个有一个容器。每个 Agent 在 JADE 平台上都有一个唯一的名字，一旦一个 Agent 知道网络上另一个 Agent 的名字，它们便可以进行透明的通信，而不需要了解彼此的实际位置。

为了便于开发多 Agent 系统，JADE 提供了两个重要组成部分：遵循 FIPA 规范的 Agent 开发平台和开发 Agent 的软件包。JADE Agent 开发平台为分布式多 Agent 应用，提供了最基本的服务和基础设施。

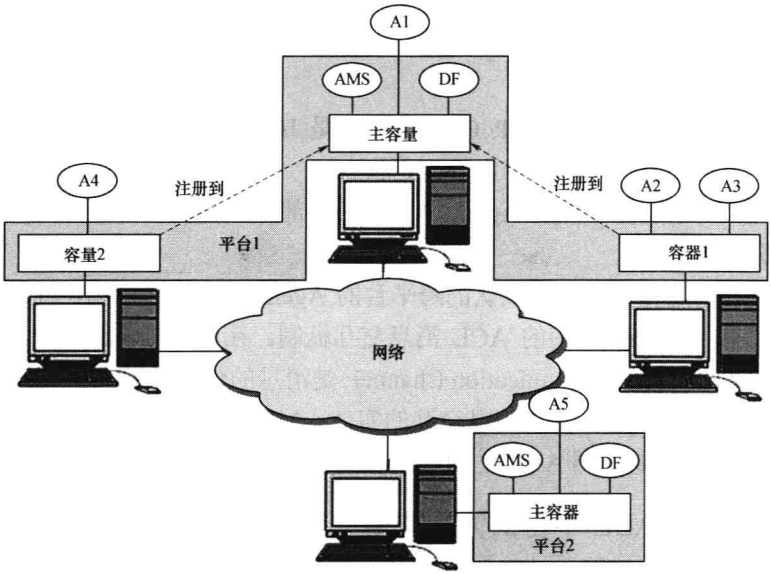

图 1-4　JADE 平台的体系结构

（1）Agent 生命周期管理和 Agent 移动性；

（2）白页服务和黄页服务；

（3）点对点的消息传输服务；

（4）Agent 的安全性管理；

（5）Agent 的多任务调度。

1. 图形工具

为了帮助用户管理和监控 Agent 运行时的状态，JADE 平台提供了一系列图形工具。

（1）RMA 图形控制台（Remote Management Agent，RMA）：远程管理 Agent，提供了对 Agent 平台进行管理和控制的图形界面。通过 RMA 控制台可以启动其他 JADE 工具。如图 1-5 所示。

此为试读，需要完整PDF请访问：www.ertongbook.com