

一种移动 agent 结构化迁移机制的设计和实现*

陶先平 吕建 张冠群 李新 董桓

(南京大学计算机软件新技术国家重点实验室 南京 210093)

E-mail: txp@softlab.nju.edu.cn

摘要 移动 agent 计算模式将成为未来网络计算的主流模式, 移动 agent 的迁移机制是其技术核心之一. 该文分析了现有移动 agent 系统中采用的几种代表性迁移技术, 提出了一种新的结构化迁移机制. 该机制的主要特点如下: (1) agent 的旅行计划和功能体完全分离; (2) 旅行计划本身也具有严格定义的结构; (3) 提供了 3 种灵活有力的迁移模式. 因此, 它能有效地控制移动 agent 的复杂度, 有利于 agent 的复用. 在该机制的基础上, 设计并实现了移动 agent 系统 Magent1.0.

关键词 移动 agent, 迁移, 旅行计划.

中图法分类号 TP393

移动 agent 技术是一种新兴的技术, 可有效地简化分布式系统的设计、实现和维护^[1]. 一般来讲^[2], 移动 agent 是一个独立运行的计算机程序, 代表用户完成特定的任务, 具有自主性、移动性、协作性、安全性和智能性等特性. 移动 agent 计算模式能有效地降低分布计算中的网络负载, 提高通信效率, 支持异步及自主交互, 支持非连接互操作, 可动态自适应, 具有一定的坚定性和容错能力. 其应用范围十分广泛, 如电子商务、并行科学计算、分布信息检索和发布、即时监控及通报、个人助理等各个方面. 因此, 移动 agent 计算模式必将成为未来的主流分布计算模式^[3].

Agent 的迁移机制是移动 agent 的核心技术之一, 受到来自学术界和工业界的广泛关注, 其代表性的工作有 Dartmouth 学院的 Agent Tcl^[4], General Magic 公司的 Telescript^[5], IBM 公司的 Aglets^[6], Mitsubishi 的 Concordia^[7] 等. 其中, Telescript^[5] 基于进程迁移理论^[7] 采用了语句级线程迁移模型, 迁移表达能力较强, 但系统负担重, 网络传输量大, 尤其是 JAVA, C++ 等语言并不支持线程状态的捕获. 因此, IBM 在采用 JAVA 实现其 Aglets^[6] 系统时, 设计了事件驱动方法实现语句级对象迁移模型, 系统负担较低, 但却增加了用户负担. 在上述几种语句级迁移中, agent 迁移条件和动作都是隐含在 agent 的代码中的, 对 agent 的设计、测试和调试都有一定的难度, 尤其在 Internet 环境中更是如此. 事实上, agent 的迁移信息应该可以从 agent 体中分离出来. Concordia^[7] 的 Itinerary 模型中的旅行计划就是该概念的体现, 旅行计划是一个独立于 agent 本身的数据结构, 存储并维护着 agent 的移动信息. Concordia 虽然可以实现迁移信息和 agent 体的分离, 但其旅行计划的描述能力和灵活性都不够, 不能有效地表达各种迁移现象.

比较以上几种方法, 我们认为, 将移动 agent 的迁移信息从 agent 体中分离出来, 用一种有足够能力的结构进行描述, 并提供灵活的迁移信息处理手段, 将静态表达和动态修改有机地结合, 应该可以克服现有移动机制存在的移动与功能体难以分离、结构化程度不够、难以理解和复用等问题, 从而为 agent 的移动提供有效的支持.

* 本文研究得到国家 863 高科技项目基金(No. 863-306-ZT02-01-4)、国家杰出青年基金(No. 61525204)和国家“九五”科技攻关项目基金(No. 96-729 1-08)资助. 作者陶先平, 1970 年生, 博士生, 讲师, 主要研究领域为面向对象技术, 分布计算技术. 吕建, 1960 年生, 博士, 教授, 博士生导师, 主要研究领域为软件形式化方法, 面向对象技术, 分布计算技术, 构件技术. 张冠群, 1976 年生, 硕士生, 主要研究领域为面向对象技术, 分布计算技术. 李新, 1974 年生, 硕士生, 主要研究领域为面向对象技术, 分布计算技术. 董桓, 1973 年生, 硕士生, 主要研究领域为面向对象技术, 分布计算技术.

本文通讯联系人: 陶先平, 南京 210093, 南京大学计算机软件新技术国家重点实验室

本文 1999-05-24 收到原稿, 1999-08-03 收到修改稿

从这个角度考虑,现有的迁移机制在能力和灵活性上都存在一定的不足。

针对以上目标,本文提出了一种完全结构化的迁移机制,其结构思想体现在以下 3 个方面。(1) agent 本身的结构化。迁移信息的描述和处理在结构上完全独立于 agent 功能体,因此,无论是 agent 的功能体还是 agent 的迁移都可单独设计、开发和使用。(2) 迁移信息的描述和处理结构化。可完整地刻画从迁移请求到恢复执行的全部信息,有足够的能力和灵活性表达多种迁移现象。(3) 提供了 3 种典型的迁移模式和多种灵活的修改手段,使得该机制在极端情况下可以实现完全的语句级迁移。总的说来,该机制结构化程度高,具有较强的能力和灵活性,便于实现和使用。我们已在移动 agent 系统 Mogent1.0^[8]中加以实现。本文首先简要介绍 Mogent1.0 系统,然后对上述结构化迁移机制详细进行解释,最后给出一个例子,并对该机制的特点进行归纳。

1 Mogent 1.0 简介

Mogent 1.0 是由南京大学计算机软件新技术国家重点实验室自主设计和开发的移动 agent 系统,于 1998 年 10 月完成。它以 Internet 为基础网络环境,以 Java 语言及其环境为基本的语言支撑,是纯 Java 的。

从概念上讲,Mogent 1.0 由 Host 和 Mogent 两个实体构成。Host 是 Internet 上的某个节点的抽象,由 IP 地址或域名标识。我们假设一个节点上只有一个 Host。Mogent 是一个移动 agent 的概念模型,由一个二元组 (HID, MID) 标识,其中 HID 是创建该 Mogent 的 Host 之标识,而 MID 是该 Mogent 当前所在节点 Host 的标识。在 Mogent 1.0 中,一个移动的 agent 就是一个 Mogent。Mogent 包括两方面的内容:旅行计划和任务体。旅行计划刻画了该 Mogent 的移动方式和移动路径。任务体则由完成该 Mogent 所承担任务的一系列方法构成。Host 提供了一个统一的执行环境,维护及管理活动在本地的 Mogent。

从系统实现上讲,Mogent 1.0 的 HostServer 由 4 部分组成:迁移支撑系统、通信协作支撑系统、安全保障支撑系统以及开发及监控支撑系统。其中,迁移机制是完全结构化的,旅行计划和任务体完全隔离且可动态装配。迁移方式也进行了扩充。旅行计划本身可在旅行前的预定或在旅行间对环境感知后自我更新。通信协作机制设计并实现了基于 MPI 的辅助通信基础构架;提供了基本的通信原语,支持点到点通信、多点通信和同步/异步的消息收发;针对 agent 的协作需求,提供了组通信的概念和支撑。安全机制是 Mogent 1.0 的可选功能,对用户透明。它提供了 agent 保护、站点保护和消息的保护这 3 个方面的内容,由数字签名和认证、资源管理及监控、信件加密和解密这 3 部分组成。同时提供了一个方便地进行 Mogent 编辑、装配和管理的开发环境。Mogent 监控工具和界面能使用户对网络中运行的由本机创建的 Mogent 和本机上外来的 Mogent 进行实时的监视和操纵。

2 一种新的结构化迁移机制

Mogent 1.0 的迁移机制的结构化体现在 3 个方面:(1) 迁移的表达和处理在结构上完全独立于 agent 功能体,因此,无论 agent 的功能体还是 agent 的迁移都可单独设计、开发和使用;(2) 迁移信息采用结构化的数据结构——旅行计划进行刻画,并提供了旅行计划动态更新手段;(3) 针对不同的迁移需求,提供了 3 种基本的迁移模式。因此,该结构有足够的能力和灵活性来表达多种迁移现象。

2.1 旅行计划的结构

旅行计划是 Mogent 1.0 迁移机制的核心。旅行计划由若干旅行步构成,每个旅行步由 5 个部分组成:移动方式、目的地、执行条件、入口方法和计划更新。如图 1 所示。

| | | | | |
|-----------------------------|-------------------------------|----------------------------------|--------------------------|-------------------------------------|
| Migration mode ^① | Destination list ^② | Execution condition ^③ | Method list ^④ | Itinerary modification ^⑤ |
|-----------------------------|-------------------------------|----------------------------------|--------------------------|-------------------------------------|

①迁移方式,②目的地列表,③执行条件,④入口方法列表,⑤计划更新方法。

Fig. 1 Itinerary step

图 1 旅行步组成

Mogent 按照旅行计划进行的移动可以描述如下:Mogent 将按“移动方式”部分给定的方式访问“目的地”部分指明的站点。每到一个新站点,先进行 Mogent 对象的解序列化,然后执行“执行条件”中给出的方法或条件,结果为真时,调用“入口方法”中指明的方法,Mogent 恢复执行,否则,该“入口方法”将被跳过,Mogent 迁移至下

一目的地. 最后,当本地执行结束时,或按计划进行下一步移动,或按“计划更新”给出的方法产生一个新的计划.

在上述结构中,“移动方式”有3种:顺序方式 SEQ、选择方式 SEL 和分发方式 ALL. 移动“目的地”描述一般采用站点的标识如 IP 地址等.“执行条件”是一个布尔函数或表达式.“人口方法”则指明一个方法名及其入口参数,该方法的体应在 Mogent 体中来定义.“计划更新”有3种:正常结束、自定义和用户干预.当 Mogent 按照某个旅行计划完成任务无需进一步移动时,可采用“正常结束”方式.在“自定义方式”下,该部分将给出一个 Mogent 体中定义的方法,调用该方法将产生一个新的旅行计划.而在“用户干预”方式下,Mogent 将请示位于创建站点的用户,并从原地携带用户给出的新计划回到当前站点并继续旅行.

2.2 3种基本的迁移模式

针对不同的迁移需求,Mogent 1.0 提供了3种基本迁移模式:顺序方式 SEQ、选择方式 SEL 和分发方式 ALL.

2.2.1 顺序方式 SEQ

该模式的一般形式如图2所示.

| | | | | |
|-----|-------|-------|-------|-----|
| SEQ | D_1 | C_1 | M_1 | T |
| | ... | ... | ... | |
| | D_n | C_n | M_n | |

Fig. 2 Sequence mode

图2 顺序方式

在图2中,SEQ代表顺序方式; D_i 表示第*i*个目的站点; C_i 表示第*i*个站点上恢复执行的前提条件,它是一个布尔函数或表达式; M_i 表示Mogent在第*i*个站点上的执行恢复入口,它是一个Mogent体中定义的方法; T 是计划更新方法或关键字.

在顺序方式下,Mogent按旅行计划依次访问结点 D_1, D_2, \dots, D_n ,当Mogent到达某个 D_i 时,调用或判断执行条件 C_i ,只有当 C_i 为真时,Mogent才会从入口方法 M_i 处开始在 D_i 上的执行,否则, M_i 将被忽略,Mogent将向 D_{i+1} 移动.如此过程不断重复,直到Mogent访问过所有的 D_i 直至 D_n .此时, T 被调用,产生了新的旅行计划.

2.2.2 选择方式 SEL

该模式的一般形式如图3所示.

| | | | | |
|-----|-------|-------|-------|-------|
| SEL | D_1 | C_1 | M_1 | T_1 |
| | ... | ... | ... | ... |
| | D_n | C_n | M_n | T_n |

Fig. 3 Select mode

图3 选择方式

在图3中, D_i, C_i, M_i 和 T_i 的解释同图2.在选择方式下,Mogent将顺序访问结点 D_1, D_2, \dots, D_n .当Mogent到达某个 D_i 时,执行条件 C_i 将被调用或判断,当 C_i 不为真时, M_i 将被忽略,Mogent向 D_{i+1} 移动;当 C_i 为真时,Mogent从入口方法 M_i 处开始在 D_i 上执行, D_i 上执行结束即是该旅行计划执行完毕,Mogent不再访问 D_{i+1} 至 D_n 的后继站点.在 D_i 上执行结束后, T_i 将被调用,以产生新的旅行计划.

2.2.3 分发方式 ALL

该模式的一般形式如图4所示.

| | | | | |
|-----|-------|-------|-------|-------|
| ALL | D_1 | C_1 | M_1 | T_1 |
| | ... | ... | ... | ... |
| | D_n | C_n | M_n | T_n |

Fig. 4 All mode

图4 分发方式

在图4中, D_i, C_i, M_i 和 T_i 的解释同图3.在ALL方式下,Mogent首先被“克隆”复制*n*份.然后同时向每个目的地派遣一份.在每个 D_i, C_i 为真时, M_i 被调用,否则被跳过.最后, T_i 负责产生新的旅行计划.

仔细分析上述3种方式,我们可以发现,它们和结构化程序设计方法学中的顺序结构、分支结构及循环结构

有一定的相似之处;SEQ 方式对应于顺序结构、SEL 方式对应于分支结构、ALL 方式对应于循环结构。在 SEQ 方式下,Mogent 依次顺序访问所有目的地址,相当于执行顺序结构中的每一条指令;SEL 方式如同典型的 switch 语句,Mogent 按旅行计划依次访问当前站点,判断执行条件,相当于在分支结构中进入每个 case 分支,判断其入口条件,但 Mogent 只从第 1 个满足条件的入口方法进入唯一的站点,执行完毕即完成此次旅行,这也相当于 case 分支的 break 语句;再看 ALL 方式,ALL 方式下“克隆”出的 n 个 Mogent 在 n 个地址上的行为相当于在不同的数据集上将 Mogent 重复执行了 n 次,进行了一个次数为 n 的循环。值得提出的是,ALL 方式在移动 agent 技术的并行任务处理方面具有潜在的优势。

2.3 旅行计划的开发和装配

Mogent 1.0 系统中的旅行计划和功能体是可以单独开发的,开发出的功能体和旅行计划可以存放于由系统管理的功能体库和旅行计划库中,以便复用。为此,Mogent 1.0 的开发及监控支撑系统提供了 Mogent 的管理工具,支持 Mogent 的开发、装配、运行及监控。如图 5 所示,为实现开发、装配、运行等功能的工具界面。

其中,旅行计划可以点击“New”来新建,也可在 Mogent 库中装载入一个已有的旅行计划,与此同时,Mogent 的功能体也可同样创建,但两者之间相互独立。旅行计划和相应的功能体确定以后,点击“Build”,支撑系统将自动完成 Mogent 的旅行计划和功能体的装配,形成一个完整的 Mogent。再点击“Run”,Mogent 将按旅行计划开始它的自主运行。

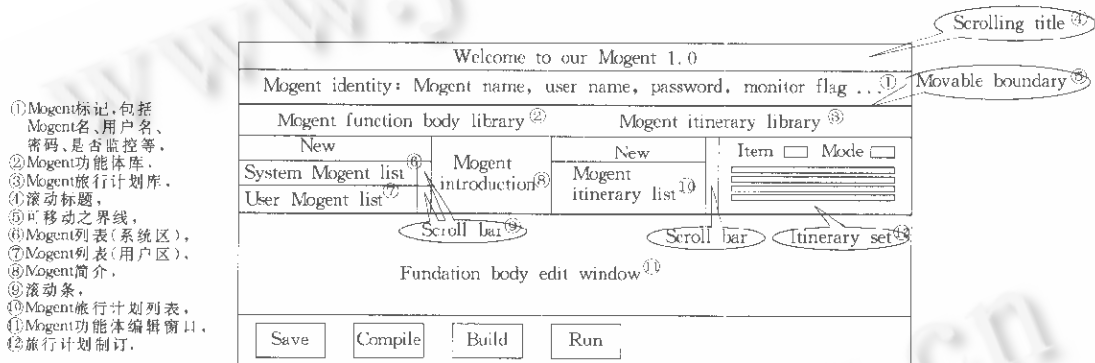


Fig. 5 Mogent developer kit
图5 Mogent 开发及装配工具界面

2.4 示例

e-market 是我们在 Mogent 1.0 基础上进行的一个电子集市试验。我们用一个局域网模拟该电子集市。集市中有 4 类角色:Market Admin, Shop, Advertiser 和 Customer。

Market Admin 在集市的两个公开站点 Qinghuai 和 Xuanwu 上各有一个,负责对集市中 Shop 信息的管理。Shop 是电子集市中交易的提供方,在任意站点都可动态地创建或消亡,创建时应向集市中的每个已有的 Shop 派遣 Advertiser。Customer 是电子集市中交易的购买方。

在集市交易中,用户首先根据欲购商品的品种、名称、数量及购买策略创建一个 customer:Bookbuyer,它以 SEL 方式向任一 Market Admin 查询所有的相关 Shop。其旅行计划如下:

| | | | | |
|-----|----------|--------|-----------------|----------------|
| SEL | Qinghuai | Guard1 | QueryShop(Book) | CreateQueryIti |
| | Xuanwu | Guard1 | QueryShop(Book) | CreateQueryIti |

其后,Bookbuyer 根据 QueryShop 得到的 Shop 的地址在 CreateQueryIti 方法中更新自己的旅行计划,以 sequence 方式开始旅行,其新的旅行计划如下:

| | | | | |
|-----|-----------|--------|------------|--------------|
| SEQ | Yuhua | Guard2 | QueryPrice | CreateBuyIti |
| | Zhongyang | Guard2 | QueryPrice | |
| | Jianye | Guard2 | QueryPrice | |

最终, Bookbuyer 根据 QueryPrice 的询价结果, 由 CreateBuyIti 再一次更新旅行计划如下, 并迁移至最理想的购物地点: Zhongyang, 并执行 BuyBook 完成任务.

| | | | | |
|-----|-----------|--------|---------|------|
| SEQ | Zhongyang | Guard3 | BuyBook | Back |
|-----|-----------|--------|---------|------|

当 Qixia 有商店新“开张”时, 它将自动创建一个 Advertiser; BookAdv, BookAdv 携带 Qixia 的经营品种及地址以 All 方式进入每个已存在的 Shop, 向该 Shop 中所有的 Customer 进行一次广播, 传递 Qixia 的新商店的商品信息, 其旅行计划如下:

| | | | | |
|-----|-----------|--------|-----------|-----|
| All | Yuhua | Guard4 | Advertise | Die |
| | Zhongyang | Guard4 | Advertise | Die |
| | Jianye | Guard4 | Advertise | Die |

3 与其他方法的比较

从系统实现的角度, 移动 agent 的迁移可分为线程迁移和对象迁移. 从用户的角度来看, 迁移又有语句级和过程级之分. 而在采用 JAVA 为实现语言的前提下, 以 Aglets 为代表的语句级对象迁移和以 Mogent 为代表的过程级对象迁移较为合理, 其中后者更为理想, 原因如下:

(1) 语句级迁移的描述功能较强, 能够表达较复杂的迁移情况, 但 Mogent 的迁移机制不仅能完整地刻画迁移行为中的必要信息, 还提供了 3 种灵活的移动方式以及多种旅行计划的修改和更新手段, 在极端情况下, Mogent 迁移机制可以模拟实现全动态的语句级迁移. 因此, Mogent 虽然是过程级迁移, 但也有足够的能力和灵活性去处理各种迁移现象.

(2) 以 Aglets 为代表的语句级对象迁移虽然能有效地减轻系统的负担, 但它是以增加用户负担为代价的, 而 Mogent 用户则完全没有这方面的负担. 因此, 过程级迁移和对象迁移实现是一种更为自然的组合.

(3) 在语句级迁移中, 迁移信息和 agent 程序代码是交织在一起的, 增加了用户的编程负担, 降低了 agent 移动的灵活性, 不能很好地适应网络环境的多变及在不同网络环境中的 agent 的复用. 而 Mogent 旅行计划和功能体的分离减轻了用户的编程负担, 旅行计划的动态装配增加了网络环境中 agent 的复用度和 agent 移动的灵活性, 可较好地适应网络环境的多变.

(4) Mogent 旅行计划在 Mogent 运行过程中是可以根据环境动态调整的, 较好地体现了 agent 特有的自适应能力, 这是 Mogent 所特有的.

总之, Mogent 1.0 旅行机制具有以下几个特点: (1) 结构化程度高, 易于使用. 旅行计划和功能体的完全分离, 使其可以独立地开发和装配使用, 易于理解、开发和复用. (2) 功能较强. Mogent 迁移机制更为一般, 有足够的能力和灵活性表达多种迁移现象. Concordia 模型可视作 Mogent 的特例. (3) 动态方式和静态方式有机地结合, 具有较强的灵活性. 在极端的情况下, 该机制可以表达完全动态的迁移. (4) 便于对迁移机制的深入研究. 旅行计划具有完全的独立性和良好的结构, 便于定义其上的操作、抽取其设计模式、刻画其语义、研究其本质. (5) 系统实现采用过程级的对象迁移, 简单、高效且用户负担较小.

4 结束语

在移动 agent 系统 Mogent 1.0 的设计和实现中, 我们提出了上述新的结构化的迁移机制. 该机制在旅行计划的结构化、旅行计划与功能体的分离以及 3 种基本的迁移模式这 3 个方面体现出了足够的能力和灵活性, 能对 agent 的迁移需求进行全面的支 持. 在以此为基础的 Mogent 1.0 系统上, 我们进行了信息检索、电子集市、并行计算和角色游戏等多个试验, 结果表明, 该机制是一个结构化程度较高的迁移机制, 它实现简单、功能较强、使用方便, 便于研究.

参考文献

- White J. Telescript technology: mobile agents. In: Bradshaw J ed. Software Agents. Cambridge, MA: MIT Press, 1996

- 2 Tao Xian-ping, Lü Jian, Dong Huan *et al.* Mobile agent: one of the main paradigm of future distributed computing. *Computer Science*, 1999, 26(2): 1~6
(陶先平, 吕建, 董恒等. 流动 agent: 一种未来的分布计算模式. *计算机科学*, 1999, 26(2): 1~6)
- 3 Lange D B. Mobile objects and mobile agents: the future of distributed computing? In: Eric Jul ed. *Proceedings of ECOOP'98 -- Object-Oriented Programming. Lecture Notes in Computer Science 1445*, Berlin: Springer-Verlag, 1998. 1~12
- 4 Gray R S. Agent Tcl: a flexible and secure mobile agent system. In: *Proceedings of the 4th Annual Tcl/Tk Workshop*. 1996. 9~23. <http://www.cs.dartmouth.edu/reports/abstracts/TR98-327/>
- 5 Lange D B, Oshima M. *Programming and Deploying Mobile Agents With Aglets*. Reading, MA: Addison-Wesley Publishing Company, 1998
- 6 Wong D, Paciorek N, Walsh T *et al.* Concordia: an infrastructure for collaboration mobile agents. In: Rothermel K ed. *Proceedings of the 1st International Workshop on Mobile Agents 97*. Berlin: Springer-Verlag, 1997
- 7 Artsy Y, Finkel R. Designing a process migration facility: the charlotte experience. *IEEE Transactions on Computer*, 1989, 38(9): 47~56
- 8 Lü Jian, Tao Xian-ping, Dong Huan *et al.* Design and implement of mobile agent system Mogent 1.0. Technical Report, State Key Laboratory for Novel Software Technology, Nanjing University, 1998
(吕建, 陶先平, 董恒等. 移动 agent 系统 Mogent 1.0 的设计和实现. 科技报告, 南京大学计算机软件新技术国家重点实验室, 1998)

Design and Implementation of a Mobile Agent Structured Migration Mechanism

TAO Xian-ping LÜ Jian ZHANG Guan-qun LI Xin DONG Huan

(State Key Laboratory for Novel Software Technology Nanjing University Nanjing 210093)

Abstract Mobile agent computation will be the main paradigm of the future network computation. The mobility is the essence of mobile agent technology. After surveying some representative approaches to migration models used in mobile agent systems, the authors propose a new structured migration mechanism in this paper. In this mechanism, the itinerary and the function body of an agent are separated completely and the itinerary scheme itself has its own strictly defined structure. Besides, three kinds of migration mode are also provided. So the complexity of mobile agents can be controlled and the reusability is supported. This mechanism has been used in the mobile agent system Mogent 1.0.

Key words Mobile agent, migration, itinerary.