软件中的软件——“AGENT”
——多 AGENT 软件协同技术在网络上的应用

杨福泉　（北京市外国企业服务总公司　100027）

近年，AGENT 软件受到业界巨大关注并开始流行。AGENT 软件超越面向目标、客户/服务器系统技术，是一种最新系统模式，它将极大地推动信息技术的发展。比尔.盖茨先生在其新著《未来之路》里也谈到 AGENT 软件在将来的信息高速公路中的巨大作用，称它是软件中的软件。

80 年代出现的面向目标的软件着眼世界上物体的存在方式，将其模块化。随着面向目标概念的普及和产品的发展，人们发现它有一些不易克服的缺陷。AGENT 软件正是为解决这些问题应运而生。

与面向目标不同，AGENT 软件是把现实世界的人（或生物）的存在方式模块化。

目标是物体，AGENT 是“人”，这决定了两个软件模式的最大不同。

物体平常作为被动体存在，应外部要求而动作，动作完成后又进入等待状态，物体不能主动自觉地行动。人则可以自发地行动，决定自己的状态。人接到命令后，可以执行命令，也可以不执行。人在处理很重要的工作时，可能听到电话响而不接。AGENT 在处理某些重要操作时也可能无视外部要求。而且无论外部有无要求，AGENT 软件平常都活动着，作一些力所能及的工作。它是一种能动的、可自发行动的软件模式。

一、多 AGENT 软件协同的重要概念－“移民”（MOBILE AGENT）

网络管理一个重要要求是解决系统间通信问题，原则之一是尽量抑制通信数据量、减少对通信线路的依赖，提高通信效率。AGENT 的移民技术是解决这个问题有效方法之一。

在网络中，多个 AGENT 在不同的机器上生成运行。当用户的一个要求需要多个系统之间进行复杂的处理和频繁的信息交换时，往往要求长时间独占通信线路。而在无线通信时，线路被中断的问题时有发生，严重影响网络稳定性和效率。所谓移民技术就是在网络中把 AGENT 程序本身（包括数据）从 A 机器传送到 B 机器，移民后的 AGENT 出示自己的资格证明就可享受移居地的各种服务，同时在异地继续进行处理，依此类推，也可以继续移向 C、D、——最后再返回到出发点，报告执行结果。这样系统之间的通信只要求在移民的一瞬间独占线路。从理论上讲大大减少了通信的数据量。

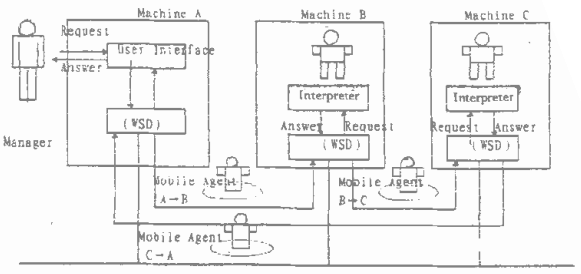

图 1　AGENT 移民原理图

移民是在 RPC 技术基础上提出的新方案，本来是为了随时把握网络故障问题。现在已经成为提高网上多 AGENT 系统效率的重要技术。移民的动作就是把 AGENT 程序与数据合为一体的 SCRIPT 模块送上通信线路。只要在此瞬间保证通信线路畅通，以后的数据交换都在局部进行。因此可以减少因通信线路问题导致的系统故障，也减轻了日常的数据维护工作量。

日本 NTT 数据通信公司利用 GENERAL MAGIC 提供的 TELESCRIPT 开发了电子销售支持系统 WebAgent。运行时它自动生成自己的 AGENT 并且自动记住要求的商品名、特征等信息。进入市场后广泛交换各种信息，比如可以从常驻在那里的目录 AGENT 了解经营该商品的店铺信息。AGENT 可以自己拷贝自身（如同孙悟空拔下自己的几根毫毛），送到所有有关的市场、经营店铺，找到最合适的商品后订货。

GENERAL MAGIC 的 AGENT 是用 TELESCRIPT 语言写成的，AGENT 的活动环境称为 TSE（TeleScript

Engine)。实践证明移民功能在分布式系统构造上是行之有效的。但是一些很大的 AGENT 程序,其移民本身的通信负担就很重。一旦在移民过程中线路出现问题,移民后的程序本身不能按计划运行了。为解决这个问题,采取只送出一部分 AGENT,在新系统上与当地 AGENT 模块组成新 AGENT 完成处理的方法—称为部分移民技术。

在 A 机器上 AGENT1 与 AGENT2 共同处理某一任务。需要移向 B 机器处理时,可以只把 AGENT 移民,移民后的 AGENT1 与在 B 机器上通过管理 AGENT 介绍的 AGENT3 共同完成 B 地的作业。这种方法进一步减轻了移民时通信线路的负担,并且避免在新系统上占用更多的资源,使移民技术日臻完善。

实现 AGENT 移民需两个必要条件。①TSE 环境要足够广泛,也就是说网络要足够大,否则在一个只有十几台机器的局部网络上,移民的威力无法显示。②移民的 AGENT 宛如一只特洛伊木马,其活动方式与病毒无异。移民过程中很有可能损害移居地主机的资源,也可能自己受到损害。所以网络的可移民的广泛领域必须统一规制,建立严格的保安措施。

二、多系统的整体流程无法静态描述

多 AGENT 系统的各个 AGENT 是异步动作的,处理内容随时变动,可以说他们是以不同的步调各行其事。所以多 AGENT 系统非常适合于各种分布系统之应用。

UNIX 机器的网络管理软件——简单网络管理规程 SNMP(Simple Network Management Protoco)里就运用了 AGENT 概念。而 HP 的 OPenView, IBM 的 NetView, DEC 的 PLYCenter, SUN 的 SunNetmanger 都是基于 SNMP 的产品。

SNMP 规程下使用两种程序模块,管理者 NMS (Network Management Station)和 SNMP AGENT(管理者 NMS 也可以视为一种 AGENT),SNMP AGENT 在机器启动时即被激活。网络管理员启动管理者程序 NMS,并通过它向 SNMP AGENT 询问情况。SNMP AGENT 作出回答。这时,NMS 管理者是客户机,SNMP AGENT 是服务器,系统是一对多的 C/S 模式。

发生故障时,SNMP AGENT 向 NMS 管理者模块发送信息,NMS 据此向网络管理员发警报。此时,SNMP AGENT 是客户机、管理者程序成为服务器。可见,这种由多 AGENT 模块构成的 C/S 系统程序之间的关系是动态变化的,此一时是客户机,彼一时又成为服务器。这也是 AGENT 自律性的又一重要特征。它的另一个好处是多个 AGENT 的独立性强,某个模块出了问题不会影响其他程序,从而提高了系统整体一可靠性。

AGENT 是异步动作的,AGENT1 进行处理时,AGENT2 在何处、做什么,AGENT1 并不知道。而且这种情况,这种知与不知也不是一成不变的。所以 AGENT 系统整体没有固定的模型,也就无法象以往的系统那样静态地描述它的整体流程。当然系统总有大的原理可循,如果必要也可以做到短时间同步运行,但是绝不是"步调一致向前进"。唯其如此,才能实现更深刻意义上的并行处理,体现 AGENT 的主动性、自律性特点。

由于无法静态描述多 AGENT 系统之整体流程,其开发过程也不是自上而下的顺序方式(TOP - DOWN),而是 BOTTON - UP 方式。传统的软件设计方法是预先做一个静态模型,然后充实各个模块的内容。AGENT 软件的构筑则不能这样,是先编制局部模块内容,再动态地构筑整个系统。

三、AGENT 软件的专用语言

目前已有一些专用的 AGENT 语言处理系统问世。象 MAGIC 公司的 TELESCRIPT,富士通与英国大学在 PROLOG 基础上开发的 APRIL, IBM 推出的基于 JAVA 语言环境的 AGLETS 都是编制 AGENT 软件专用语言,但是开发 AGENT 软件并不一定要使用这些专用语言。

从一些实例看,AGENT 的实体就是程序(执行起来称为进程)。一个 AGENT 就是一个程序,用什么语言编写不是主要的。

由于面向目标的软件开发环境的效率,可靠性都有很大提高。利用现有的面向目标的软件开发环境去开发 AGENT 软件是切实可行的。从这一点来说,面向目标的软件和 AGENT 软件是共存的。实现分布式目标环境、异步信息交换的中间软件也可以在开发 AGENT 软件上发挥作用。

美国 GENERAL MAGIC 和东京工业大学的系统都采用了翻译器(INTERPRETER)实现移民不是偶然的巧合。使用翻译器后,AGENT 程序可以用文本数据表示。它不会象二进制数据那样容易与通信控制码混淆,发送信息比较简单。在不同的处理器、操作系统的机器之间进行移民时,翻译器可以在一定程度上克服各系统的差别,是使移民较容易实现的行之有效技术。

用面向 AGENT 开发的专用语言书写软件时,不记

述处理顺序，而直接描述 AGENT 的状态、判断原则，“必须如何如何…”等等。生成的 AGENT 就会先天智慧，依“计”而行。

下面是用 AGENT 专用语言开发的一节程序。是一些描述 AGENT 的责任、信念、知识、计划、动作选择的语句，如 CMT(责任)、K(知识)等，描述 AGENT 的心理状态。这样的程序可以使计算机更亲切、更机智。但与目前我们熟悉的编程思想、方法有很大距离。

①CMT(9:00, Bill, Allan, At(9:30, Doc38, Room12)).

②K(9:00, Bill, CAN(9:00, Bill, At(9:30, Doc38, Room12))).

③B(9: 00, Bill, t + 15 < t1, CAN(t, Chris, At(t1, Doc38, Roomm12))).

④PF(9:00, Bill, CMT(t, Bill, a, At(t1, x, y))-.

B(T, Bill,CAN(t, Bill, At(t1, x, y))-

CAN(t, b, At(t1 < x, y)))

CH(t, Bill, Request(t, Bill, b, At(t1, x, y))).

说明：

①9 点，Bill 为 Allan 作准备，9 点半必须把 Doc38 文件送到 12 号房间。

②9 点，Bill 知道他不能在 9 点半把 Doc38 文件送到 12 号房间。

③9 点，Bill 相信在预定时间(t1)之前如果还有 5 分钟的话、Chris 可以把 Doc38 文件送到 12 号房间。

④9 点，Bill 必须为 a 在 t1 时把 x 送到 y，否则，如果 b 能做到，就制订委托 b 去做的计划。

执行结果：Bill 委托 Chris“9 点半请把 Doc38 送到 12 号房间”

注：CMT(责任)必须做…，K(知识)知道…(…一定是事实)，B(信念)相信…，PF(部分计划)　如果…就做…，CH(选择行动)做…

四、实用化的网络 AGENT 软件结构

AGENT 软件经过多年研制开发，已经进入实用化阶段。IBM 与 UNISYS 在网络 AGENT 应用上先行一步，已经把 AGENT 软件技术放在网络管理的核心位置。IBM 96 年 4 月发表了其用于工作流程管理的 FLOW MARK for MVS AIF，同年底又推出 Intelligent Agent。UNISYS 的 SYSTEMY 以业界规范 CORBA 为标准执行环境，强化多 AGENT 功能。

这两个公司的 AGENT 可以嵌入新开发的应用程序，或者追加在已有的应用程序上，也可以与网络管理、工作流管理、信息检索、电子信箱等应用程序动态组合。

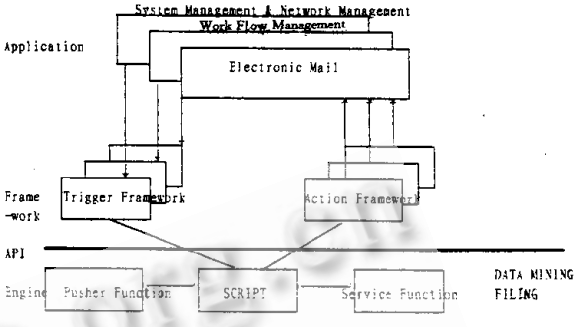

图 2　IBM 推出的网络 AGENT 产品结构图

IBM 的 AGENT 结构主要由引擎和框架两部分组成。引擎是专用的程序库，由控制应用程序之间运行的 SCRIPT 语言、运行环境的学习而具备的并且不断丰富变化的推论功能模块和服务功能模块三部分组成。可以用 JAVA 语言实现。

在专家系统里，推论一般是利用 PROLOG 语言内涵的后向推论功能对规则库(或称知识库)的操作，规则库的内容可以人工修改、编辑。这里的推论功能以应用程序的要求为启动动力，自动动态生成执行这些处理的 SCRIPT 程序。由于推论功能，不用事先对 SCRIPT 处理流程进行详细记述，它可以根据用户(或应用程序)的要求随机生成、随时变化。服务模块在不断变化的处理过程中抽出有规则性的东西，具体通过 DATA MINING 和 FILING 功能达到进一步学习、积累经验、提高智能的作用。

框架是使引擎的功能更便于与现有应用程序连结的中间软件。有了框架的支撑，不用把引擎嵌入应用程序。框架把发来的触发信息转换成引擎推论之动力。反之引擎得出的推论结果也通过 API 接口传达给运动框架，再发给应用程序。

IBM 提供了框架产品用于系统管理的工具群称为 TME，工作流管理的称为 Flow Mark，将来计划提供从通用机到 Windows NT 上应用的各种类型的框架产品。IBM 还公布了引擎与框架之间 API 接口的规格。鼓励各软件商以此为标准开发各种用户需要的框架产品。

(来稿时间：1997 年 8 月)