

Fan Yushun
Cao Junwei

范玉顺 曹军威 编著

多代理系统 理论、方法与应用

Multi-Agent Systems
Theory, Method and Applications

1.52



TUP

清华大学出版社



Springer

施普林格出版社

24

TP311.52
F-25

多代理系统

理论、方法与应用

范玉顺 曹军威 编 著

清华大学出版社 施普林格出版社

(京)新登字 158 号

内 容 简 介

软件代理技术和多代理系统是在复杂应用环境下进行软件系统开发的一个热点方法。本书在总结了现有的软件代理技术和多代理系统的研究和应用成果的基础上,对多代理系统的理论、方法、应用和开发方法进行了全面的介绍,为从事多代理系统技术研究和应用的人员全面了解多代理系统技术提供了详细的科研资料,是研究人员深入开展研究的基础。本书可以作为软件公司进行软件系统开发的参考资料,还可以作为计算机、自动化、企业管理专业研究生的学习教材。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: 多代理系统 理论、方法与应用

作 者: 范玉顺 曹军威 编著

出 版 者: 清华大学出版社 施普林格出版社
北京清华大学学研大厦,邮编 100084
<http://www.tup.tsinghua.edu.cn>

责任编辑: 马幸兆

印 刷 者: 北京昌平环球印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 12.25 **字数:** 279 千字

版 次: 2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷

书 号: ISBN 7-302-05447-9/TP·3212

印 数: 0001~4000

定 价: 25.00 元

前言

软件代理是一个能在特定环境下连续、自发地实现功能,同时与相关代理和进程相联系的软件实体。其显著的特点是具有自主性、社会能力、被动响应能力、主动响应能力、时间连续性、自适应性和可移动性。多代理系统是指由多个代理组成的系统,它是为了解决单个代理不能够解决的复杂问题而由多个代理协调合作形成的问题求解网络。为了使代理之间能够合理、高效地协作,代理之间的协调机制成为多代理系统研究的重点问题。一般地,每个代理被认为是一个物理的或者抽象的实体,能够作用于自身和周围环境,并可与其他代理进行通信。多代理系统的研究方法实际上是用模拟人类社会系统的运作机制来提高计算机系统解决复杂问题的能力。采用多个代理进行协作,通过任务分解和任务协调提高整个系统的能力是一个可行的途径。另外通过多代理之间的合作还可以克服单个代理知识不完全、处理的信息不确定等缺点。

软件代理和多代理系统的产生和发展是有深刻的应用背景的,其产生和发展也为提高软件系统处理复杂问题提供了有效的途径。随着计算机硬件技术、网络技术的不断提高,软件的覆盖范围和应用领域在不断扩大,特别是因特网(Internet)技术的迅速发展,导致软件系统的复杂程度呈迅速增长的趋势,除了软件系统的复杂性增加外,软件系统的分布式特性也更加明显,由位于多个物理地点的许多计算机系统协作完成科研和实际应用任务的工作模式已经取得了显著的成绩。实际应用和数据的复杂性、动态性与分布性的增加对软件系统的能力不断地提出更高的要求。人们希望,软件系统不仅能被动地响应信息处理的需求,完成事先指定的任务,还能具有一定程度的智能,能够主动地预测、适应乃至积极地寻找途径以支持用户完成任务。除了由人来协调不同系统之间的运作外,还要求各个系统的软件间能自动地进行合作以实现更加复杂的功能。从这种需求出发,来自计算机、人工智能、信息处理、自动控制等几个领域的研究人员开始对软件代理技术及多代理系统方法进行研究开发。进入 20 世纪 90 年代,软件代理的研究得到迅猛发展,各种各样的应用及方法的出现是软件代理成为潮流的重要标志。

本书在总结分析现有的软件代理技术和多代理系统的研究和应用成果的基础上,对多代理系统的理论、应用和开发方法进行了全面的介绍。其中第 1 章对多代理系统的产生背景、基本概念进行了介绍。第 2~5 章对多代理系统的相关理论和方法进行了介绍,包括知识表达与推理、代理通信语言、多代理协调模型和代理协商模型等理论和方法。第 6~10 章介绍几个有代表性的多代理系统的应用。第 11,12 章重点讨论多代理系统的实

现方法和开发工具。

感谢赵博博士为本书编写了第9章。希望本书能够为从事和有志于开展多代理系统研究和应用的研究人员了解多代理系统提供有益的参考,本书也可作为计算机、自动化、企业管理和机械制造专业研究生的学习教材。

软件代理技术和多代理系统已经引起越来越多的研究和应用人员的关注,其相关理论、方法、应用和工具还处于迅速发展之中。由于掌握的相关资料和作者水平有限,书中的缺点和错误在所难免,欢迎广大读者批评指正。

作 者

2001年11月于清华园

目 录

第 1 章 概述	1
1.1 多代理系统的产生背景	1
1.2 多代理系统的基本概念	4
1.3 多代理系统的相关理论	8
1.4 多代理系统的应用前景.....	11
1.5 多代理系统的开发方法及工具.....	14
第 2 章 知识表达与推理	17
2.1 知识表达.....	17
2.1.1 概述.....	17
2.1.2 对象的语义网表达方法.....	19
2.1.3 基本概念与规则.....	22
2.1.4 PVS 规则	24
2.2 基本的知识学习操作.....	25
2.2.1 一般化的形式化定义及其规则.....	25
2.2.2 基本问题解决方法.....	29
2.3 知识学习.....	33
2.3.1 知识启发.....	34
2.3.2 规则学习.....	36
第 3 章 代理通信语言	42
3.1 概述.....	42
3.2 代理通信语言的组成与特征.....	44
3.2.1 代理通信语言的组成.....	44
3.2.2 代理通信语言的特征.....	47
3.3 知识查询与操作语言.....	48
3.3.1 KQML 语言简介	49

3.3.2 KQML 代理环境	52
3.4 KQML 应用	54
第 4 章 多代理协调模型	56
4.1 概述	56
4.2 协调模型	57
4.3 控制驱动的协调模型	60
4.4 数据驱动的协调模型	63
第 5 章 代理协商模型	67
5.1 概述	67
5.2 代理精神模型	68
5.2.1 形式化模型	69
5.2.2 语法	70
5.2.3 语义	71
5.2.4 代理类型	75
5.2.5 形态关系	77
5.3 讨论与论点评价	80
第 6 章 AARIA 代理系统	85
6.1 概述	85
6.2 系统需求	85
6.3 系统设计	88
6.4 AARIA 示例	91
6.5 AARIA 和其他的基于代理的生产控制系统结构	93
6.6 结论	95
第 7 章 基于 ADEPT 的经营过程管理系统	96
7.1 概述	96
7.2 ADEPT 代理结构与经营过程建模	97
7.3 用户报价经营过程示例	103
7.4 小结	105
第 8 章 分布信息提取中移动代理技术的应用	106
8.1 概述	106
8.2 移动代理系统综述	108
8.2.1 多语言系统	108

8.2.2 基于 Java 的系统	109
8.2.3 其他系统	110
8.2.4 移动代理方法的比较	111
8.3 技术报告搜索器应用实例	111
8.4 规划	113
8.5 待研究解决的问题	116
第 9 章 MAS 技术在生产调度研究中的应用	117
9.1 生产调度问题概述	117
9.1.1 调度问题的基本概念	117
9.1.2 典型的生产调度问题	119
9.1.3 经典调度问题的算法研究及其局限	121
9.2 多代理调度系统	122
9.2.1 多代理调度系统的发展历程	122
9.2.2 对多代理调度系统的一般分析	123
9.3 实体型 MASS	125
9.3.1 实体代理的划分	125
9.3.2 基于合同网协议的系统结构	127
9.3.3 基于拟市场模型的系统结构	129
9.4 过程型 MASS	130
9.4.1 过程代理	130
9.4.2 基于多代理机制的调度算法集成	131
9.5 MASS 应用前景展望	135
第 10 章 基于代理的 CIMS 应用集成平台运控系统	137
10.1 CIMS 应用集成平台	137
10.2 运行管理与控制系统	138
10.3 对运控代理模型结构的研究	139
10.4 基于代理的动态应用集成	144
第 11 章 面向代理的软件工程	149
11.1 概述	149
11.2 面向代理方法的产生背景	150
11.3 面向代理的方法	152
11.4 面向代理的软件的生命周期	155
11.5 面向代理开发中需要注意的问题	157

第 12 章 多代理系统开发工具	161
12.1 概述.....	161
12.2 JATLite 简介	165
12.3 JATLite 代理消息路由器	166
12.4 JATLite 的层次结构	168
12.5 JATLite 应用实例	170
参考文献.....	174

1.1 多代理系统的产生背景

“代理”一词对应的英文单词是“agent”，其原意主要指一个可以代表其他执行的人或者有权力和权威执行某事的人，或者是指完成某事的一种方法和手段。很早以前，agent 一词就用于表示在商业运作中承担中间人的角色，大家对于 agent 这个角色的功能和作用基本上理解为有一定的独立决策和行动能力，能够主动地采取一定的方法和手段完成委托人交给的任务。这个在商业领域的专有名词目前已经广泛地应用于计算机领域，经过多年的研究，agent 已经被研究人员赋予了许多不同的含义，并且给出许多种不同的定义。虽然现在意义上的代理已经在许多重要方面不同于早期的概念，但是关于代理的基本功能和作用本质与商业运作中中间人的角色的功能和作用没有太大的变化。

人类热衷于非人类代理的想法由来已久，但直到第二次世界大战以后，随着计算机技术和控制理论的发展，类似带有自主性或者自治性(autonomy)的代理装置才开始出现。与今天的代理最密切相关的“祖先”恐怕是早期一些基本的控制装置，如车间生产线控制器、飞机的自动起降和飞行控制器。如果机器人可以被称为一种硬件代理的话，软件代理则是一个软机器人(soft robot)，它在计算机的软件世界中完成它的使命。

代理技术在计算机领域的研究和应用源于 20 世纪 70 年代美国麻省理工学院研究人员开展的一系列关于分布式人工智能的研究。麻省理工学院的研究人员在对信息系统处理复杂问题的能力进行分析时发现，通过协作将一些简单的信息系统组成一个大的系统可以显著提高系统处理复杂问题的能力，并且通过定义合理的协作机制可以提高整体系统的智能水平。由此产生了使用具有一定智能，能被动地响应信息处理的需求，除了完成事先指定的任务外，还能够主动地预测、适应乃至积极地寻找途径以支持用户完成任务的软件代理的概念和方法。

最初研究软件代理的工作者主要是要研究分布智能的计算模型，进而从实用性的角度出发解决两个问题：一个是简化分布计算的复杂性；另一个是克服人机界面的局限性。从本质上讲，解决这两个问题是为了更好地抽象计算机接口这一趋势的延续：一方面是希

望用更加有效、通用和一致的代理与代理之间的接口取代目前的程序与程序之间的接口,以便进一步抽象和封装计算机硬件、软件和通信模式的实现细节;另一方面希望能够进一步用代理来封装用户需求,从而简化执行复杂任务所需要的人与程序之间的操作界面(接口)。

1. 简化分布计算的复杂性

大多数人认为未来的计算环境将由运行于多种异构平台上的分布软件系统组成,而目前的软件系统大多数处于分离(disjoint)状态,或者只能在一些基本方面进行通信和合作,如文件传输、打印服务和数据库查询等等。很多软件系统之间只是通过特殊的接口进行特定的连接。近几年,随着面向对象技术的推广和一些基本标准如 TCP/IP, HTTP, ODBC 的形成,才开始出现系统或服务之间封装式的连接。图 1.1 描述了系统连接的变化过程。

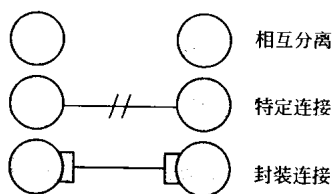


图 1.1 系统连接的变化过程

随着从基于网络操作系统的分布计算模式向基于 Internet 和 Intranet 的分布计算模式的转移,作为支持这种分布计算模式的基本支撑软件环境且独立于操作系统的互操作网络服务开始出现,如命名(naming)服务、目录(directory)服务和安全(security)服务等。Netscape 公司就是提供这些互操作网络服务软件的一个典型代表。由一些软硬件厂商组成的联盟正在制定一整套独立于操作系统的网络服务规范以及应用开发接口(如 Java)。在这些网络服务规范和应用开发接口广泛应用后,不管计算机上装载的是什么操作系统,它们都可以实现整个网络的“即插即用”,即与网络上其他计算机上的应用系统实现无缝的集成和互操作。而计算机操作系统的地位将由目前的计算机系统的核心退化成为硬件设备驱动器的简单组合。

高层次的软件间互操作(interoperability)要求软件系统本身有定义和解释描述系统的能力和知识,这样才可以保证软件系统能够自动地在多个协作的软件系统间,完成任务规划(task planning)、资源分配(resource allocation)、执行(execution)、监听(monitring)以及可能的干预(intervention)功能。为实现这一点,一个智能代理可以起到全局资源管理的作用。然而,单个代理在小型网络系统下也许能起作用,但当系统协调操作的数量增加时,便会变得不合实际,单代理的能力将成为整个系统的瓶颈。进一步智能化互操作是在每一个相互协作的系统中,应用多个对等代理来完成任务的协调,并且要求代理能够在高层次上对用户意图而不是具体实现加以反映。因此不但要有类似在底层上基本通信原型的封装(encapsulation)能力,同时还应提供规划层上高层次的封装能力。图 1.2 给出基于代理的互操作系统原理图。图 1.2(a)表示使用单个代理作为互操作系统的协调器,代理和应用之间的连接线表示代理与应用之间的通信;图 1.2(b)表示使用分布代理作为互操作的协调器,代理与代理之间的连接线表示代理与代理之间的通信。

2. 克服人机界面的局限性

目前,直接操作人机界面(DMI)已成为计算机软件的基本功能,它相对于命令行交互的人机界面是一个很大的进步,在 DMI 方式下,用户可以通过可视化的方式对软件对象

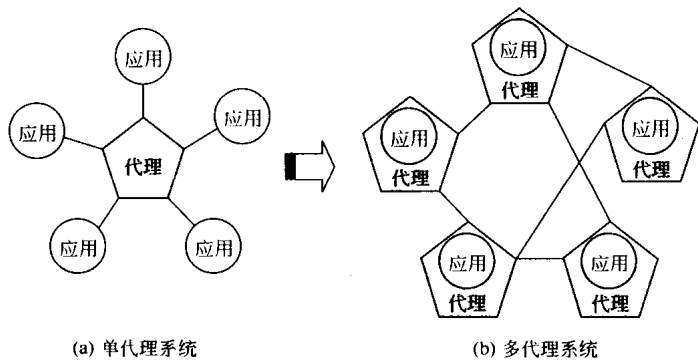


图 1.2 基于代理的互操作系统

进行操作。但是,在 DMI 方式下,用户要不断地被告知可作用于哪类对象。如果被操作的对象与现实世界有一定自然的联系,用户可以根据以往获取的经验很快地知道对象可以做什么和怎样去做。不过,当任务在规模和复杂程度上增加时,DMI 的局限性就暴露出来了,而且这种局限性给用户的操作带来很大的不便。在这种情况下,研究人员提出了应用面向代理的方法来提高用户操作的方便性。表 1.1 列出了 DMI 可能遇到的问题和应用面向代理方法(AOA)相应的好处。

表 1.1 DMI 与 AOA

	典型 DMI 的局限性	AOA 的好处
信息搜索	搜索空间巨大,在大规模分布系统中,很难用浏览和传统的检索方法迅速搜索到所需要的信息	代理可以后台运行,所提供的搜索及过滤功能可帮助人们获取大量的信息资源
执行时间	只能够响应实时的用户要求,但有时用户可能需要首先规划好某项任务,然后在未来的某个特定时间执行,或者用户需要在离开机器时让软件自动响应系统产生的事件	代理可以被引导在特定的时间执行任务,或者当系统产生的事件到来时自动被唤醒
扩展功能	大多数直接操作人机界面,不能通过行为和对象的组合得到更高层次的功能	代理可以设计成可扩展和可重组的,而一般的按钮式的界面对象却不能。代理不仅能执行命令,还可以和用户进行通信和交流,从而共同完成任务
方便操作	对于简单任务,界面功能简单、有良好的可预知性,但对于大任务就显得过于僵化和严格	通过在目标和策略层上进行引导,代理在创造条件帮助解决问题方面具有更大的柔性
适应范围	面向软件功能,而不是面向用户任务和环境	代理可以将用户任务及条件考虑在内
学习能力	不能通过学习对行为加以改进	代理可以通过学习不断改进其算法和行为

许多来自不同领域的研究者从自身的研究角度出发,开发出各种变化形式的代理,这在一定程度上导致了“代理”这一名词的泛滥。许多程序和系统并没有真正的代理性质却被冠以代理的名字,如一些系统被称为代理程序仅仅是因为:① 可以在远程机器上执行

一些事先规划好的任务;② 用高层次的编程语言和脚本指令指导完成了一些计算机底层的计算任务;③ 抽象或者封装了不同的信息源或服务之间存在的差别;④ 实施了某些简单的或者复合的“认知功能”;⑤ 具有分布式智能的某些特点;⑥ 在人和程序之间承担了一定的中介作用;⑦ 执行了某些“智能助理”的功能;⑧ 在自我引导(self-directed)下可以活动于不同计算机之间;⑨ 能以用户更加可信的方式表示自己;⑩ 使用了代理通信语言(ACL)等等。

近些年来,国内许多学者也对代理技术、多代理系统和多代理系统的应用开展了广泛的研究。目前,代理技术已经广泛地应用于图形处理、产品开发过程管理、冲突协调、计算机辅助设计、车间调度、决策支持、供应链管理、开放式信息系统结构、集成平台与集成框架、 workflow管理、办公自动化、电子商务、CSCW、Internet 应用等众多领域。无论这些应用的出发点和取得的结果如何,有一些基本的认识是共同的,即采用代理技术和多代理系统是为了提高软件系统的柔性,提高应用系统的智能水平,最终提高系统解决复杂问题的能力。

1.2 多代理系统的基本概念

1. 代理的基本定义

一个能为许多软件代理研究者接受的定义是:软件代理是一个能在特定环境下连续、自发地实现功能,并且是与相关代理和进程相联系的软件实体。连续、自发的要求来源于环境的变化,要求代理能在没有人的引导和干涉下以柔性、智能的方式对用户的需求实时地加以响应,更理想的情况是在某一特定环境下,在一段时间内,反复实现某一功能后能吸取经验教训,即所谓的学习功能。另外,还希望代理能与环境中的其他代理和进程通信、合作,甚至可以在不同的地方来回移动。现在许多软件代理都是相当脆弱和特殊的,没有一个能在一般意义下满足这些要求,因此,目前的软件代理只能被视为一种统一的代名词,用以概括诸多特定条件下带有局限性的代理类型。归纳起来,代理具有下面的基本属性:

- 自主性:代理能够在没有人或其他代理干预下完成其大部分功能,控制其内部状态。
- 社会能力:代理能够主动和其他代理或人交互,以实现其目标。
- 被动响应能力:代理能感知周围环境并对其间的变化产生实时响应,这些动作的执行基于触发规则和预定的执行计划。
- 主动响应能力:代理能够主动地进行基于自身目标和信念的活动。
- 时间连续性:代理的进程是连续的,而不是向一般程序那样只一遍就结束计算。
- 自适应性:代理能适应环境的变化,这往往是通过学习机制实现的。
- 可移动性:代理可以携带数据和指令移动到其他环境中并在那里执行指令。

对代理概念的定义和理解是在实际应用过程中逐渐丰富和发展起来的。在这样的过程中,有两个既相互联系又相互区别的定义方法值得注意,即描述性定义方法和归因化定

义方法。

(1) 描述性定义方法

描述性定义方法(as description)是通过对事物内在属性(attributes)的完整描述来定义事物。由此出现了许多用于描绘代理属性的词语。例如:反应力(reactivity)用于描绘代理针对不同的感知有选择地产生不同响应的能力;人格化(personality)指代理可以区分模糊属性(如情感)的不同特征的能力;适应性(adaptivity)指代理具有学习和积累经验并不断提高的能力;移动性(mobility)指代理可以在自引导下在不同计算机之间移动。如果要描述所有的代理可能具有的属性及其不同的组合,就必须提供一种简单的刻画代理类型空间的方法。

在代理的研究领域中已经提出了几个具有代表性的代理分类方法。其中一个很有影响的代理分类方法是在 IBM 公司的论文^[66]中提到的,它用交互性、智能性和移动性三维定义代理类型空间,见图 1.3。交互性刻画代理在系统中与其他软件实体的相互关系。最起码的,代理应该可以异步运行。代理如能在某种程度上表达用户意图,其交互性便有所增加。进一步地,代理可以与数据、应用、服务或其他代理进行交互。智能性用于描述代理推理、学习等行为的智能化程度,也就是代理接受用户目标和执行任务的能力。最基本地,代理可以表达用户一定程度的喜好。高层次的智能包括掌握用户模型并有推理能力。代理还应该学习和适应环境,这不但包括用户目标的变化,而且包括代理资源的变化。移动性包括代理自身通过网络移动于系统之间的能力。移动代码可以在一台机器上形成,然后在另一台机器上运行。移动对象可以在执行过程中携带积累的状态数据从一台机器转移到另一台机器。

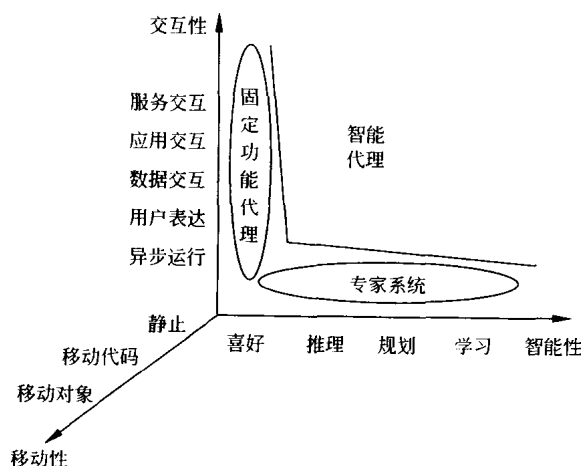


图 1.3 智能代理范围示意图

Nwana 提出了另外的代理分类法^[114],即代理可以根据如下特征分类:

- 移动性:静止或移动。
- 符号推理模型的表达方式:协商式的或反应式的。
- 主要属性:自主性、互操作性和学习能力。从这些属性可以延伸出其他 4 种代理类型,协作(collaborative)代理、协作型学习(collaborative learning)代理、界面(interface)

代理和智能(smart)代理,见图 1.4。

- 角色:例如信息代理,Internet 代理。
- 混合哲理:把两种以上的方法混合应用在一个代理中。
- 第二属性:其他一些从属于主要属性的次要属性。

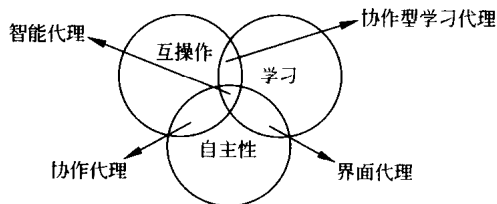


图 1.4 Nwana 根据主要属性的划分方法

Franklin(1996)给出的关于代理的定义是:一个自主代理可以看成是一个系统,它置身于一定的环境,感觉各种环境的变化,并可以根据自身的功能作用于环境^[58]。从这个定义出发,一个温度调节系统也可以称为一个代理系统,图 1.5 给出一个更广泛意义下代理的划分方法。

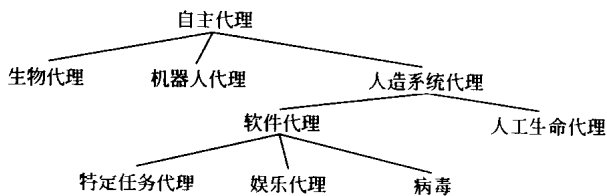


图 1.5 Franklin 代理分类法

(2) 归因化定义方法

归因化定义方法(as ascription)是用事物的外在行为和功能来说明事物的。“代理的行为即代理”就是用这种方法对代理进行简单定义的。由于代理的概念处于发展和逐渐完善阶段,一个人设计系统中的所谓的智能代理可能只是另一个人设计的系统中的某个智能对象,今天的智能代理也许在明天看来是很拙劣的程序。通过罗列一系列属性一次就全面、完整地说明代理的特征是不可能的,归因化定义便是这种条件下的产物。

时间和经历最终将决定“代理”名词的含义和寿命,就像现在常用的许多其他计算机名词,如“桌面(desktop)”、“鼠标(mouse)”等一样,开始仅是一种隐喻(metaphor),最终将代表一种具体的软件产品。随着软件代理技术的不断发展与应用,其名词一定会逐渐变成能为大家理解的某一意思的代名词。

软件代理之所以区别于其他一般程序就在于它具有自主性,这种自主性表现为它可以封装一些使用者不必要关心的属性和操作。有两种最基本的方法可以用来实现这种自主性:一种是智能的方法,即采用人工智能的相关方法开发的代理,又称为智能代理;另一种是协作的方法,即通过多个代理相互联系来实现功能,一般这样的系统属于多代理系统(multi-agent system, MAS)。

2. 多代理系统

多代理系统是指由多个代理组成的系统,它是为了解决单个代理不能够解决的复杂问题,由多个代理协调合作形成的问题求解网络。为了使代理之间能够合理高效地进行协作,代理之间的协调机制成为多代理系统研究的重点问题。一般地,每个代理被认为是一个物理的或者抽象的实体,能够作用于自身和周围环境,并可与其他代理通信。实际上,多代理系统的研究方法是用模拟人类社会系统的运作机制来提高计算机系统解决复杂问题的能力的。如同一个人无法完成许多复杂和巨大型的任务一样,单个代理也无法设计成有足够的能力来解决面临的许多问题,因此,采用多个代理进行协作,通过任务分解和任务协调提高整个系统的能力是一个可行的途径。另外通过多代理之间的合作还可以克服单个代理知识不完全、处理的信息不确定等的缺点。

多代理系统可以是紧密型的,也可以是松散型的,多代理系统的组成粒度也可粗可细。如人工神经网络就被某些人认为是一种紧密型的、精细的多代理系统,在这种系统中,每个元件或单元是一个非智能的代理,通过合作,使每个单元能够对数据进行反应,从而提高整个系统的智能水平。而分布式问题求解系统通常是由一组松散耦合的代理组成的分布式代理网。每个代理能够完成高级问题求解中的一部分工作,能够随环境而修改自己的行为,并且与其他代理进行通信和协作。如产品并行设计过程中的冲突协调和约束满足问题的求解就可以采用松散耦合的多代理系统来协作解决。

对于智能代理的需求更多地源于用户接口的智能化,对于多代理系统的需求更多地源于分布计算的复杂性。分布计算与人工智能技术相结合促进了分布人工智能的研究,多代理系统是分布人工智能研究中的一个重要部分,如图 1.6 所示。分布人工智能系统的另一部分重要的工作是分布问题求解(distributed problem solving, DPS)。DPS 一般针对信息管理的问题,而 MAS 针对行为管理。它们之间的交叉部分指一些用代理来进行信息管理的系统,所以 DPS 和 MAS 之间并不存在一个清晰的界限。

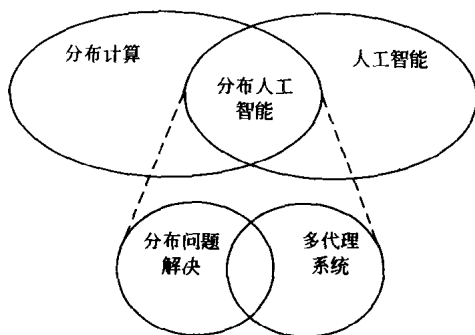


图 1.6 多代理系统与分布人工智能

DPS 的信息管理包括任务分解和结果综合,就是说,复杂的任务要被分解为简单的、易于管理的子任务,并被发派到不同的处理器上,然后不同子任务的结果被合并综合。一般的 DPS 系统由相同专业的人或小组完成,不同的解决问题的模块之间的关系存在很多事先的假设和定义。而 MAS 却没有这样的保证,不同的代理可以完全由不同专业的人来设计开发,尽管它们之间需要相互联系。MAS 中个别代理的行为要比代理之间相互协

调的行为简单得多,所以 MAS 的研究更强调代理行为之间的相互协调和作用。

目前多代理系统理论与技术已经在广泛的领域内得到了应用,开放系统和复杂系统是多代理系统典型的应用领域。多代理理论与技术得到广泛重视的主要原因是它能够被用来解决一些其他技术所不能或不容易解决的问题。

(1) 开放系统

开放系统是指系统结构具有不断动态变化的能力的软件系统。这种系统的特点是系统的组件是事先不确定的、可以随事件改变的,还可能是异构的,因为不同的组件可能是不同的人在不同的时间应用不同的技术和工具开发的。最典型的开放系统的例子就是 Internet,它可以被看成一个巨大的分布信息资源库,而网络中的每个节点都是由不同的组织或个人开发的。任何在 Internet 上操作的系统都必须有能力使用这些不同的信息源。用户可以给出一些限制条件,但不可能进行直接的操作。一些带有协商性质的功能是必须的,而这正是多代理系统所要研究的问题。

(2) 复杂系统

应用模块化的方法可以简化软件开发的复杂性。代理是一种将系统模块化的有力工具。如果遇到的问题特别复杂,规模大而且不好预测,那么比较好的办法是先开发一些模块化的组件,每一个完成一个特定的功能,这样当出现模块间相互联系的问题时,系统中的代理就要进行合作,以确保这种关联得到适当的实现。基于代理的方法就意味着将整个问题划分成小的、简单的组件,每一个都很容易开发和维护,而且专门解决某一方面的问题。这样的分解使得每一个代理都能采用最合适的方法来解决其特定的问题,而不必被强迫使用某种统一的方法让步于整个系统,这种解决问题的方法对于局部系统通常并不是最优的。代理从概念上提供了一种与对象相同的抽象方式,软件开发可以利用代理把复杂的系统概念化成许多相互合作的、自治的软件模块,对于许多应用来说,这种高层次上的视图用代理来抽象最合适。

1.3 多代理系统的相关理论

多代理系统需要借助一些相关的理论和技术来实现。系统中的单个代理通过一定的数据结构以及相应的操作对要解决的问题进行建模,就要用到知识表达和推理方法的相关理论。代理之间的通信与信息交流需要以一定层次的协议作基础,代理通信语言(agent communication language, ACL)可以被用于完成这样的功能。多个代理以一定的方式组织在一起以实现相互协作,就属于多代理协调问题。代理之间的高层次交互包括协商和竞争等,可以用逻辑模型进行表达和实现。本节对涉及多代理系统的相关理论进行简单的介绍,相关部分的详细内容在本书的后续章节进行介绍。

1. 知识表达与推理

代理的知识表达是要在现实世界的实际问题域与代理内部符号推理系统(symbolic reasoning system)之间建立起联系。符号推理系统利用领域模型进行推理工作,它包括