

Agent:特性与分类^{*})

Agent: Properties and Classification

杨 鲲 翟永顺 刘大有 TP18

(吉林大学计算机科学系 吉林大学符号计算与知识工程开放实验室 长春 130023)

Abstract After analyzing definitions and descriptions of agent from many reference materials, we discuss the main properties of agent, and present that the most fundamental properties of agent should be: reactive, autonomous, goal-oriented and environment-specified, therefore giving the definition of minimal agent. Based on these properties, we discuss the multi-agent system(MAS) and agent-based system(ABS). After giving many classifications of agent, we produce the property-based classification of agent. At last, we discuss the relationship between agent technology and others related fields such as AI, expert systems and object-oriented technology.

Keywords Agent, Minimal agent, Properties of agent, Classification of agent, AI

1. 引言

Agent 技术,特别是多 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。著名的 Agent 理论研究者 Jennings 教授和 Wooldridge 博士甚至把它誉为是“软件开发的又一重大突破”^[12]。Agent 技术已经被广泛应用到各个领域,从相对简单的个人电子邮件过滤器到复杂的空中交通控制、柔性制造系统(FMS)等大系统,都是 Agent 技术大显身手的地方^[1]。然而,在这些系统中,Agent 的定义和表现又各不相同,那么究竟什么是 Agent 呢?它有哪些特性?一些研究者对此给出了不尽相同的解释(见下一节),但我们认为这些对 Agent 特性的描述要么太简单而不足以体现 Agent 的本质特性,要么太复杂以至于将大量的 Agent 排除在外。为此我们提出了“最小 Agent”的概念,并给出了最小 Agent 应满足的特性,亦即 Agent 的最基本特性。另外,我们在阅读文献的时候还常常看到自治 Agent、软件 Agent 和移动 Agent 等,它们的含义和相互联系如何?这涉及到 Agent 的分类问题(当然也与特定领域的突出特点有关),为此我们给出了 Agent 的分类。另外我们还对基于 Agent 的系统

(ABS)、多 Agent 系统(MAS)等的含义及其与 Agent 的相互联系进行了简单说明。最后我们简单地讨论了 Agent 与相关领域的关系,目的也是为了更好地了解 Agent 的特点。

目前,国内广泛开展了基于 Agent 技术的研究和应用系统开发,并取得了较大的成果,但对如何正确使用 Agent 这个词还存在着这样或那样的问题。搞清 Agent 的最基本特性也有助于正确使用 Agent 和判定一个系统是否是真正意义上的 Agent 系统。

在应用 Agent 技术设计和构建一个复杂软件系统时,常常要涉及到有关 Agent 三个方面的问题^[12],即:

(1)Agent 理论:主要研究 Agent 的概念、特性、形式表示和推理;

(2)Agent 体系结构:主要研究如何构建 Agent 以满足设计的要求,以及选择什么样的软硬件结构;

(3)Agent 程序设计:主要研究如何对 Agent 进行编程,Agent 语言应具备的原语,以及如何有效地编译和执行 Agent 程序。

本文的讨论主要涉及到第一个方面(即 Agent 理论)的部分内容。

^{*})本文得到国家自然科学基金和 863 计划的资助。杨 鲲 博士生,研究方向为 DAI 和 MAS。翟永顺 讲师,研究方向为人工智能。刘大有 教授,博士生导师,研究方向为人工智能、专家系统和 MAS。

2. 什么是 Agent?

研究者在有关 Agent 的研究工作中给出了各种 Agent 的定义,这些定义分别体现了他们自己对 Agent 的理解和应用。但需要说明的是,有些研究者强调某些特性并不意味着他们否定其它特性,我们这里列出众多 Agent 定义的主要目的是为了找出 Agent 的主要特性。

Russell^[1]等认为,“Agent 是任何能通过传感器感知(perceive/sense)环境并通过执行器对环境进行动作(act)的东西。”这个 Agent 的定义可以说是最宽的,任何一个东西,只要能从环境中获取信息并对环境进行操作均可以认为是 Agent。在这种定义下,如果把人的眼、耳等器官当作传感器,把人的手、脚、嘴等当作执行器的话,人也是 Agent;机器人也是 Agent,因为机器人有摄像机或红外仪可以作为传感器,并且有各种马达作为执行机构;甚至温度计也可以被认为是 Agent,因为它能感知环境温度,并对环境(假设环境也包括人的话)显示所测的温度;所有的程序也均是 Agent,因为被编码的位串可作为其感知和动作。很显然,这个定义在很大程度上依赖于我们如何定义环境、感知和动作。需要指出的是,在这个定义中,Russell 使用了感知和动作两个词。

P. Maes^[6]认为,“Agent 是在复杂动态环境中能自治地感知环境并能自治地通过动作作用于环境,从而实现其被赋予的任务或目标的计算系统。”与上边 Russell 的定义相比,它在感知和动作的基础上又增加了两个限定词:自治(主)的和面向目标的。Maes 也把这种 Agent 称为自治 Agent。在这个定义中,“自治”的含义是指 Agent 的行为由其自身的经验决定的。

M. Coen^[1]认为,“Agent 是可以进行对话,协商的软件。”Coen 这里给出的其实是软件 Agent 的定义。对话即指 Agent 与环境(包括其它 Agent)的交互,其实质也是感知和动作,只是措辞不同而已。协商的目的是为了使各方达成共识^[4],协商必然要用到通信,即与其它 Agent(也可能包括人)进行通信,故我们这里使用可通信的这个词。

Hayes-Roth 认为,智能 Agent 除了能对环境进行感知和动作外,还应当能够进行推理以解释感知信息或决定执行什么动作,即 Agent 应当是可推理的。

Wooldridge 和 Jennings^[12]的 Agent 可以说是

对以上 Agent 定义的一个总结,他们认为,Agent 应当是一个硬件或软件系统(后者更常见),该系统具有如下特征:

- 自治性:Agent 的运行不受人或其它 Agent 的直接干涉,Agent 对自己的内部状态和动作有一定的控制权;
- 社交能力:Agent 可以通过某种 Agent 语言(如 KQML 语言^[2])与其它 Agent 或人进行交互;
- 反应性:Agent 能够感知环境,并能及时做出反应;
- 预动性:Agent 能够展现出一种导向目标的行为。

Agent 的社交能力其实就是 Agent 可通信的能力,反应性即是感知和动作(我们以后用反应性来表示 Agent 的感知和动作能力),预动性即是前边提到的面向目标的能力。另外,Jennings 等在文[4]中还使用“有弹性的”这个词来描述 Agent 的特性,它是指社交能力、反应性和预动性三者的综合。

White 认为运行于 Internet 中的 Agent 应当是可移动的。移动 agent 是目前 Agent 研究领域一个很活跃的分支。

C. Byrne 认为运行于复杂环境中的 Agent 还应当能够根据以前的经验校正其行为,即具有学习或自适应的能力。

L. Foner^[3]着重于从人机交互方面探讨 Agent 的特性,提出了用户的期望、人机对话以及用户接口的类人化等问题,而且 Foner 还从风险与信任以及协作等方面研究了 Agent 的特性。

一些研究人员,如 A. S. Rao^[7]等,从 Agent 的精神状态出发对 Agent 的特性进行了更为深入的研究。

Agent 的特性除了上边提到的外,还有持续性或时间连续性、自启动、自利等特性。

研究和开发人员没必要构建一个包括以上所提到的所有特性的 Agent 或多 Agent 系统,他们往往是从应用的实际需要出发来开发包含以上部分特性的 Agent 系统。但是既然称之为面向 Agent 的技术或系统,那么,它们还是有一些共性可寻的,这就是我们下一节要讨论的最小 Agent 的概念。

3. Agent 的最基本特性和最小 Agent

上边给出了很多关于 Agent 的定义或解释,不同的研究者对 Agent 定义时所使用的术语或所侧重的方面是不同的。其实 Agent 的准确定义是很难

给出的,就象我们无法准确给出一个“人工智能”的定义一样,但这并不影响我们对 Agent 的研究和应用。Russell 等^[1]也指出,“Agent 定义是一个分析系统的工具,而不是用来将世界分为 Agent 和非 Agent 两大类”。而且精确的定义只能采用数学或逻辑等精确的描述手段来实现,而这样做一方面陷入一个个具体的 Agent 模型(有关 Agent 形式表示的问题超出本文的讨论范围),如 Bratman 的 BDI 模型、Konolige 的演绎模型等,无法把握 Agent 的总体特性;另一方面,对没有世界模型的 Agent 也无能为力,典型的如 Brooks 创导的非表示智能和非推理智能。

但把握 Agent 的最基本特性是必要的。我们把在上节中所提到的 Agent 的主要特性列为表 1。

表 1 Agent 的主要特性

特性	其它名称	含义
反应的	感知和动作	及时感知环境的变化并执行动作以作用于环境
自治的		对自己的行为或动作有控制权
面向目标的	预动的 有目的的	Agent 能够展现出一种导向目标的行为
可通信的	协商 协作	与其它 Agent(也可能包括人)进行通信
可推理的		解释感知信息或决定执行什么动作
可移动的		Agent 能够跨平台持续运行
自适应的	学习	Agent 能够根据以前的经验校正其行为

作为存在于某个环境中的 Agent,它首先得具备感知环境和动作的能力,所以 Agent 首先得是反应的。Agent 还应当是自治的,它不应当象普通程序那样完全由外界控制其执行,也不应当象对象那样其内部定义的方法可由外界调用。Agent 对自己的动作有绝对的控制权,不允许外界(人或其它 Agent)的干涉。这一特性将 Agent 与程序和对象划开了界限。自治性是 Agent 很重要的一个特性,也是 Agent 被国内一些学者译为“自主体”的重要原因。所以在多数情况下,Agent 也被称为自治 Agent。另外,Agent 还应当是面向目标的,或说其具有预动性,这也是其被广泛应用于各类复杂系统的一个重要原因。另外,我们认为 Agent 还必须是“针对环境的”,在某个环境中存在的 Agent 换了一个环境有可能就不再是 Agent 了。例如,一个只有可视仪的机器人 Agent 如果到了一个没有光线的环境

中就不再是 Agent 了,因为它无法对环境进行感知。各类 Agent 往往只满足于它们所存在的环境。

在多 Agent 系统(MAS)中才要求 Agent 是可通信的(指与其它 Agent 进行通信),尤其是需要协作和协商的 MAS 中,Agent 的可通信性是极其重要的。Agent 的可推理性也不是必须的,有些 Agent 可以完全不用推理,如 Brooks 提出的非推理的智能。在同质环境中的 Agent 可以不具备移动性,对 Agent 移动性的要求常体现在异质环境中,典型的如 Internet。Agent 的自适应性或学习能力也不是每个 Agent 必须具备的。

综上所述,一个 Agent 的最基本特性应当包括:反应性、自治性、面向目标性和针对环境性,我们将只满足这四条特性的 Agent 称为最小 Agent。每个 Agent 首先是一个最小 Agent,然后才可以根据其应用情况再拥有其它特性。

我们也可以根据 Agent 的特性给出一个 Agent 的简单定义如下:Agent 是一类在特定环境下能感知环境,并能自治地运行以代表其设计者或使用者实现一系列目标的计算实体或程序。

多 Agent 系统(MAS)是由多个 Agent 组成的系统,它在 Agent 理论的基础上重点研究 Agent 的互操作性,内容包括 MAS 的结构、如何用 Agent 进行程序设计(AOP)^[2,3],以及 Agent 间的协商和协作等问题。

基于 Agent 的系统(ABS)是指使用了 Agent 思想或技术的系统。ABS 可能只包含一个 Agent,如用户接口 Agent 或软件秘书等,但通常是指 MAS 的应用实例^[1]。Jennings 等^[4]还指出,在原理上,ABS 可以是只借用 Agent 概念、而用其它技术(如 OOP)实现的系统。

4. Agent 的分类

从上边的讨论我们也可以看出,Agent 是一个综合的术语,概括了一系列各式各样的 Agent。通过对 Agent 的合理分类也可以在一定程度上了解 Agent 的特性和其具体表现情况。我们可以从不同的角度对 Agent 进行分类。

根据 Agent 的工作环境可将其分为软件 Agent 和人工生命 Agent^[13],前者“生存”于计算机操作系统、数据库以及网络等环境中,我们平常所提到的 Agent 常常是指软件 Agent,人工生命 Agent 则“生存”于一种人造的环境中,如计算机屏幕等。软件 Agent 与程序不同,所有的软件 Agent 均是程序,但

并非所有的程序均是软件 Agent,只有满足 Agent 四条最基本特性的程序才能称为软件 Agent。软件 Agent 又分为同质 Agent 和异质 Agent,进而还可以根据环境情况再分为数据库 Agent、文件 Agent、网络 Agent 或 Internet Agent 等。

根据 Agent 的控制结构和方法又有多种分类方法。根据控制机制可以将 Agent 分为基于规则的 Agent、基于规划器的 Agent、基于神经网络的 Agent、基于机器学习的 Agent、基于模糊逻辑的 Agent 等;或分为集中控制 Agent 和分布控制 Agent。Brustoloni 还根据控制结构的适应性程度将 Agent 分为常规 Agent、规划 Agent 和自适应 Agent。

根据 Agent 的理论模型中是否采用了物理符号系统可将 Agent 分为思考型 Agent(包含对环境的符号表示和推理,如著名的 BDI Agent^[7])、反应型 Agent(不包含对环境的符号表示和推理,以反射的方式响应环境的变化,如 Maes 的 Agent 网络结构)和混合型 Agent(思考型 Agent 和反应型 Agent 的有机结合,如 Georgeff 的过程推理系统 PRS)。

根据其功能或其应用领域又可以对 Agent 进行各种分类,下面我们给出 Jennings 等在文[5]中给出的 Agent 的应用领域以作为 Agent 的一种分类方法。Jennings 等将 Agent 的应用主要分为四大方面:(1)工业应用,具体包括过程控制(如 ARCHON)、工业制造(如 YAMS)、空中交通控制(如 dMARS)等;(2)商业应用,具体包括信息管理(如各种 Internet 上的信息收集器和信息过滤器)、电子商务(如 Kasbah)、商务过程管理(如 ADEPT)等;(3)医疗应用,具体包括患者监视(如 Guardian)、医疗保健等;(4)娱乐应用,具体包括游戏(如 Tetris)、交互式戏剧和电影等。

从本文的陈述角度来看,一种直接的方法是按表 1 给出的 Agent 特性来分类,即所谓的 Agent 特性分类法。当然每个 Agent 均应满足我们在上一节中提到的四个最基本的特性,在这基础上增减其它特性便可以产生各种 Agent,例如,可移动的推理 Agent。这样就自然形成一种有关特性的层次分类结构,如可移动的推理 Agent 是推理 Agent 或可移动 Agent 的一个子集。

5. Agent 与相关领域的关系

5.1 Agent 与人工智能(AI)

智能 Agent 主要源于 AI 的研究工作,也有人

将 AI 的任务定义为构建智能 Agent^[8]。但要注意的是,人工智能中的智能与 Agent 中所寻求的智能是有所不同的。前者常常包括学习功能、推理功能等;而对 Agent 的智能的要求是:它可以根据环境等因素决定下一步要执行的动作,至于是否要具备学习或推理的功能要根据领域问题的要求而定。所以也有人认为:“智能 Agent 是 99% 的计算机科学和 1% 的人工智能”。但有些 AI 技术无疑是构建 Agent 的最好方法。

5.2 Agent 与专家系统(ES)

专家系统是 AI 向实用化方向发展中的一个十分引人注目的领域,它能够利用人类专家的知识来解决该领域的问题。比较著名的专家系统如 Stanford 大学研制的 DENDRAL 系统,它可以根据质谱仪的显示结果和有关数据推断出复杂有机化合物的分子结构。专家系统与 Agent 的一个重要区别是象 DENDRAL 这样的专家系统在本质上是非嵌入式的,即它们不与环境发生直接的交互,它们不是从传感器而是从用户获得信息。另外,专家系统通常不要求操作的实时性。需要指出的是,随着网络技术的高速发展和信息一体化的要求,目前有很多专家系统(尤其是多专家系统)的特点与 Agent 非常相似,一个典型的例子是用于工业过程控制的 ARCHON 系统^[10],它既是一个多专家系统,又是一个多 Agent 系统。也就是说,Agent 技术可作为构建专家系统的一种方法。

5.3 Agent 与面向对象(OO)技术

面向对象(OO)技术发展日益成熟,而且得到了极大的应用,成为目前软件开发的主流技术。Shoham 在 OOP 的基础上提出了 AOP(面向 Agent 的程序设计)技术,并将 AOP 视为一种特殊的 OOP^[9]。Shoham 还在理论模型的基础上对二者进行了比较^[9],下面我们借鉴 Jennings 等人^[4]的观点,从应用的角度看一下二者的区别。

OOP 的最基本特征之一是“封装”,但对象只封装了其内部状态,而没有封装其行为(在 OOP 中表现为方法),它不是完全自治的。例如,若一个对象 a 的方法 m 被指定为可以被其它对象激活(即指定为 public),则其它对象可在任何时候激活 m,对象 a 对 m 在何时被执行没有控制权。而在多 Agent 系统中的情形是这样的:若 Agent x 想让 Agent y 执行动作 k,则 y 有可能执行,也有可能不执行动作 k(即拒绝 x 的请求)。Agent 对其行为有完全的控制权,亦即同时“封装”了内部状态和行为。

其次,我们还要求 Agent 有良好的反应性和面向目标的特性,但这些要求不在 OOP 模型的要求之列。另外,在 MAS 中的每个 Agent 均有它们各自的控制线程,而在标准对象模型中整个系统只有一条控制线程(新开发的并发 OOP 也可以有多条控制线程),这其实也是 Agent 和对象二者自治程度不同的一种体现。但 OOP 常常作为 AOP 开发的有效工具,即用 OOP 技术实现 AOP 的思想。AOP 还是一个较新的领域,借鉴 OOP 的某些成熟特性(如封装、继承、多态等)对 AOP 的研究和发展有很大的意义。

参考文献

- 1 Coen M. Available at: <http://www.ai.mit.edu/people/sodabor/slideshow/total/p001.html>.
- 2 Finn T, et al. The KQML information and knowledge exchange protocol. In: Third Intl. Conf. on Information and Knowledge Management, 1994
- 3 Foner L. Available at: <http://www.foner.www.media.mit.edu/people/foner/julia/subsection3-4-1.html> which is titled as What's an agent? Crncial notions
- 4 Jennings N R, et al. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems*, 1998(1): 275~306
- 5 Jennings N R, et al. *Agent Technology: Foundations, Applications, and Markets*. Springer-Verlag Heidelberg, 1997
- 6 Maes P. *Designing Autonomous Agents*. Cambridge, MA: MIT Press, 1990
- 7 Rao A S, Georgeff M P. BDI Agents, From Theory to Practice. In: Proc. of the First Intl. Conf. on Multi-Agent Systems (ICMAS-95). San Francisco, USA, 1995
- 8 Russell S, Norvig P. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995
- 9 Shoham Y. Agent-Oriented Programming. *Artificial Intelligence*, 1993, 60(1): 51~92
- 10 Wittig T. ARCHON: An Architecture for Multi-Agent Systems. Ellis Horwood, 1992
- 11 Wooldridge M J. Agent-Based Software Engineering. *IEEE Transactions on Software Engineering*, 1997, 144(1): 26~37
- 12 Wooldridge M J, Jennings N R. Agent Theories, Architectures, and Languages: A Survey. In: Proc. of ECAI-Workshop on Agent Theories, Architectures, and Languages. Amsterdam, Netherlands, 1994 1~32
- 13 Franklin S, Graesser A. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: the Proc. of the Third Intl. Workshop on Agent Theories, Architectures, and Languages. Springer-Verlag, 1996

(上接第 29 页)

一个新的应用需求点。本文提出了一个主动合作计算模型,基于言语行为理论以及我们所提出的 Multi-agent 系统计算的理论框架,本文讨论了该合作计算模型所涉及的 Agent 通讯行为,分析了 Agent 如何基于这些通讯行为并通过对这些通讯行为的响应来完成主动合作计算,并进一步给出了主动合作计算模型的定义。

Multi-agent 系统的主动合作计算模型的提出,有助于我们进一步理解和认识 Multi-agent 系统的主动合作关系和主动合作计算,分析和评估 Multi-agent 系统中的不同合作计算模型;同时有关交互行为的全心满足语义的定义可以帮助我们深层次地理解和分析 Agent 如何参与合作、如何对各种交互行

为作出响应以指导 Agent 的设计,并给我们提供了一种有效的方法和手段用于规范和描述复杂、庞大的 Multi-agent 系统。

参考文献

- 1 毛新军. Multi-Agent 系统中 Agent 计算的理论框架. [博士学位论文]. 国防科技大学研究生院, 1998. 4
- 2 Haddadi A. *Communication and Cooperation in Agent Systems: A Pragmatic Theory*. Springer-Verlag, 1996
- 3 Singh M P. *Multi-agent System: A Theoretical Framework for Intentions, Know-how, and Communications*. Springer-Verlag, 1994