

Agent 研究现状与发展趋势*

刘大有 杨鲲 陈建中

(吉林大学计算机科学系 长春 130023)

(吉林大学符号计算与知识工程国家教育部开放实验室 长春 130023)

E-mail: dylu@mail.jlu.edu.cn

摘要 对有关 Agent 理论和实践两方面的研究现状、关键技术和发展趋势进行了综述和讨论,试图勾勒出目前 Agent 研究的脉络、重点及其发展趋势。主要内容包括:(1) 智能 Agent 的特性、表示和推理;(2) 多 Agent 系统的联合意图、协商和规划;(3) 面向 Agent 的程序设计。最后,简单讨论了 Agent 技术目前所面临的挑战。
关键词 智能 Agent, 多 Agent 系统(MAS), 面向 Agent 的程序设计(AOP), 意识态度, DAL。

中图法分类号 TP18

Agent 的理论、技术,特别是多 Agent 的理论、技术,为分布开放系统的分析、设计和实现提供了一个崭新的途径,被誉为“软件开发的又一重大突破”。Agent 理论与技术研究最早源于分布式人工智能(DAI),但从 80 年代末开始,Agent 理论、技术研究从 DAI 领域中拓展开来,并与许多其他领域相互借鉴和融合,在许多不同于最初 DAI 应用的领域得到了更为广泛的应用。面向 Agent 技术作为一门设计和开发软件系统的新方法已经得到了学术界和企业界的广泛关注。

我们认为,对 Agent 的研究大致可分为智能 Agent、多 Agent 系统(multi-agent system,简称 MAS)和面向 Agent 的程序设计(agent oriented programming,简称 AOP)这 3 个相互关联的方面。智能 Agent 是 MAS 研究的基础,我们可以将有关智能 Agent 的研究统一在 MAS 的研究之下,这样,智能 Agent 被看成是 MAS 研究中的微观层次,主要研究 Agent 的理论和结构;而有关 Agent 间关系的研究则构成了 MAS 研究的宏观层次。智能 Agent 和 MAS 的成功应用要借助于 Agent 的应用方法,即 AOP 以及 AOP 开发工具或平台。

本文沿着从单个 Agent(智能 Agent)到 MAS 的研究和应用这一发展过程,突出讨论 Agent 的理论基础和技术等方面的研究现状,并对目前有关 Agent 研究的热点问题和关键技术进行了介绍。限于篇幅,本文不可能给出有关这些领域各方面的详细介绍,只希望勾勒出目前 Agent 研究的重要方面、关键技术及其发展趋势。本文最后简单讨论了 Agent 技术目前所面临的挑战。

1 智能 Agent

本节在讨论单个 Agent 特性的基础上介绍了 Agent 的 3 种表示和推理方法,并对其各自的特点进行了分析。这 3 种方法也是建造 Agent 的方法。

1.1 Agent 的特性

在应用 Agent 技术的系统中^[1],Agent 的定义和表现各不相同。那么,究竟什么是 Agent,它又有哪些特性?这些是 Agent 研究者和应用者最为关心的问题,也是目前 Agent 研究领域一直在争论和探讨的热点问题之一。

在有关 Agent 特性的研究中,最经典和广为接受的是 Wooldridge 等人有关 Agent 的“弱定义”和“强定义”

* 本文研究得到国家自然科学基金(No. 69673014)和高等学校博士学科点专项科研基金资助。作者刘大有,1942 年生,教授,博士生导师,主要研究领域为分布式人工智能,知识工程,专家系统。杨鲲,1969 年生,讲师,主要研究领域为分布式人工智能,多 Agent 系统和移动 Agent。陈建中,1968 年生,副教授,主要研究领域为分布式人工智能,多 Agent 系统。

本文通讯联系人:刘大有,长春 130023,吉林大学计算机科学系

本文 1999-04-23 收到原稿,1999-09-20 收到修改稿

的讨论^[2]。文献[3]在分析一些典型的研究报告和应用系统中,在对 Agent 的描述或定义的基础上指出,一个 Agent 的最基本的特性应当包括:反应性、自治性、面向目标性和针对环境性。每个 Agent 首先应具备这 4 条最基本的特性,然后再根据其应用情况拥有其他特性。Agent 可以拥有的其他特性包括:移动性、自适应性、通信能力(包括协商、协作等能力)、理性、持续性或时间连续性、自启动、自利等特性。一些研究人员还从 Agent 的精神状态出发,对 Agent 的特性进行了更为深入的研究(见本文 1.2.1 节)。

研究和开发人员没必要构建一个包括上述所有特性的 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几部分特性的 Agent 系统。但我们认为,既然是称为面向 Agent 的技术或系统,那么就应当满足上述提到的 4 条最基本的特性。我们可以根据 Agent 的特性给出一个 Agent 的简单定义: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户实现一系列目标的计算实体或程序。

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 的互操作性以及 Agent 间的协商和协作等问题。基于 Agent 的系统(Agent-Based system,简称 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^[4]。Jennings 等人还指出,ABS 可以是只借用 Agent 概念而用其他技术(如 OOP)实现的系统^[1]。

1.2 Agent 的表示和推理

从建造 Agent 的角度出发,单个 Agent 的结构通常分为思考型 Agent、反应型 Agent 和混合型 Agent。

1.2.1 思考型 Agent

建造 Agent 的经典方法是将其看作是一种特殊的知识系统,即通过符号 AI 的方法来实现 Agent 的表示和推理,这就是所谓的思考型或慎思型 Agent(deliberative Agent)。

(1) 思考型 Agent 与意识系统

思考型 Agent 的最大特点就是将其看作是一种意识系统(intentional system)。人们设计的基于 Agent 系统的目的之一是把它们作为人类个体或社会行为的智能代理,那么 Agent 就应该(或必须)能模拟或表现出被代理者具有的所谓意识态度(intentional stance),如信念、愿望、意图(包括联合意图)、目标、承诺、责任等。

已经有很多学者认为,把 Agent 作为意识系统来研究是合理的^[2-5]。意识态度和意识系统概念的引入更有助于研究者们以一种自然而又直观的方法来理解、描述、规范、推理和预测基于 Agent 系统的内部结构、运行规律和变化状态等。选择什么样的意识态度来刻画 Agent 是建造思考型 Agent 所首先要考虑的问题。不同的 Agent 模型或系统对意识态度有不同的认识和分类观点。

Shoham 等人根据应用的特点将意识态度分为信息类(刻画 Agent 所具有的信息)、动机类(Agent 的动作选择)、社会类(与 Agent 的社会、道德和理性行为有关)和其他类(如欢喜、恐惧等)。Wooldridge 等人^[2]则把单个 Agent 应具有的意识态度(或精神状态)分为两大类:信息态度和积极态度。前者是指 Agent 所拥有的关于自己、环境及其他 Agent 的信息和知识,如信念和知识等;后者是指那些能导致 Agent 执行动作的状态,如愿望、目标、意图、承诺、责任、能力等。他们还指出,一个理性的 Agent 总是基于信息态度去采取积极态度,如根据信念去选择目标和形成意图。在 Rao 和 Georgeff 提出的 BDI(belief-desire-intention;信念-期望-意图)模型中,用信念、愿望和意图这 3 类意识态度来刻画 Agent 的结构,并最终通过规划库来研究 BDI 模型的抽象性质和推理过程^[6]。

(2) Agent 形式化工具的选择

任何一个完备的形式化系统都要包括相互独立的两个方面的属性:形式语言及其语义模型。Wooldridge 曾经指出,在 Agent 形式化方面,经典的命题逻辑和一阶谓词逻辑是不合适的^[2]。目前最常用的形式化工具是模态逻辑(包括各种时态逻辑)和可能世界语义,即将意识态度看成是一种模态。在语法方面,通过在公式中引入一些非真值功能的模态算子,从而构成模态语言;在语义方面,用可能世界及其可达关系来解释信念、目标等意识概念的含义。目前,关于模态逻辑和可能世界语义的研究已形成一整套的相关理论,成为表示和推理智能 Agent 和多 Agent 系统的最有力的形式化工具。标准的可能世界语义的最主要的不足是其逻辑全知问题(logical omniscience),已有很多研究者对其进行了改进。

意识态度的其他表示方法还有元语言与对象语言以及具有符号解释结构的演绎模型等。

(3) 典型工作介绍

Bratman 最早提出用意图、信念、期望等概念来描述 Agent 系统,其观点为后来的许多相关工作,尤其是 BDI 模型奠定了理论基础。第 1 个对信念和意图概念进行形式化描述的当推 Cohen 和 Levesque,他们以 Bratman 的工作为基础,以言语行为理论为背景,采用形式化逻辑方法讨论了自主 Agent 的信念、目标、意图、动作等的“理性平衡”(rational balance),分析了意图在维持平衡中的作用。该工作被公认为是对 Agent 理论和性质进行形式化研究的奠基性经典工作,并为绝大多数相关研究工作所引用。在所有有关 Agent 理论和结构的研究中,BDI 模型以其坚实的理论基础和方便的可操作性而成为目前的研究和应用中使用最多的 Agent 模型。BDI 模型研究的典型代表当数澳大利亚的 Rao 和 Georgeff,他们提出了一系列描述 Agent 意识态度的 BDI 逻辑^[6]。这种 BDI 逻辑用 3 个基本模态算子刻画信念、愿望和意图 3 个元素;它把命题形式的分时序逻辑 CTL* 扩展为一阶形式,并定义了一些时态算子、路径算子和路径公式来描述时间。关于单 Agent 意识态度的其他形式化工作还有很多,它们仍然是目前 Agent 理论研究的一个热点。

1.2.2 反应型 Agent

从上面的讨论我们可以看出,符号 AI 的特点和种种限制给思考型 Agent 带来了许多尚未解决、甚至根本无法解决的问题,这就导致了反应型 Agent (reactive Agent) 的出现。反应型 Agent 的支持者们认为,Agent 的智能取决于感知和行动(所以在 AI 领域也被称为行为主义),从而提出 Agent 智能行为的“感知-动作”模型。他们认为,Agent 不需要知识,不需要表示,也不需要推理,Agent 可以像人类一样逐步进化,Agent 的行为只能在现实世界与周围环境的交互作用中表现出来。反应型 Agent 的支持者们还认为,符号 AI 对真实世界中客观事物及其行为工作模式的描述是过于简化的抽象,因而不可能是真实世界的客观反应。

反应型 Agent 的最有力的支持者当数 MIT 的 R. Brooks。他提出一种不同于符号 AI 的,称之为子前提结构 (subsumption architecture) 来的建造 Agent 的控制机制。该结构是由用于完成任务的行为 (behaviours) 来构成的分层结构,这些行为相互竞争以获得对机器人的控制权。这种虽然简单的结构在实践中被证明是非常高效的,它甚至解决了传统符号 AI 很难解决的问题。

1.2.3 混合型 Agent

反应型 Agent 能及时而快速地响应外来信息和环境的变化,但其智能程度较低,也缺乏足够的灵活性。思考型 Agent 具有较高的智能,但无法对环境的变化作出快速响应,而且执行效率相对较低。混合型 Agent (hybrid Agent) 综合了二者的优点,具有较强的灵活性和快速响应性。

混合结构的系统通常被设计成至少包括如下两部分的层次结构:高层是一个包含符号世界模型的认知层,它用传统符号 AI 的方式处理规划和进行决策;低层是一个能快速响应和处理环境中突发事件的反应层,它不使用任何符号表示和推理系统。反应层通常被给予更高的优先级。采用分层结构时要处理的主要问题是,各层应采用什么样的控制框架以及各层之间应如何交互。

采用混合型 Agent 结构的一个典型实例是过程推理系统 (procedural reasoning system, 简称 PRS),它是一个在动态环境中推理和执行任务的 BDI 系统。

从当前的研究和应用现状来看,思考型 Agent 占据主导地位,因为多数研究和开发者都喜欢使用自己已经较为熟悉的符号 AI 技术和方法;反应型 Agent 的研究和应用目前尚处于初级阶段;混合型 Agent 由于集中了上述两种 Agent 的优点而成为当前的研究热点。

在所有有关 Agent 理论和结构的研究中,BDI 结构以其坚实的理论基础和方便的可操作性而成为目前研究和应用领域中使用最多的 Agent 结构,但同时也在一定程度上存在着滥用 BDI 名称的现象^[7]。

2 多 Agent 系统 (MAS)

MAS 的协作求解问题的能力超过单个 Agent,这是 MAS 产生的最直接的原因。导致 MAS 研究逐渐兴起的其他原因还包括:与已有系统或软件的互操作;求解那些数据、能力和控制具有分布特性的问题以及提高系统的效率和鲁棒性等。与单个 Agent 相比,MAS 有如下特点:每个成员 Agent 仅拥有不完全的信息和问题求解能力(因而其观点是有限的),不存在全局控制;数据是分散或分布的,计算过程是异步、并发或并行的。

MAS 的研究历史最早可以追溯到 80 年代中期的 Actors 模型,接着是 Davis 和 Smith 提出的合同网协议,合同网协议至今仍被认为是关于通信、MAS 协商研究的经典工作。目前,MAS 研究的主要方面包括:MAS 理论、多 Agent 协商和多 Agent 规划等。其他比较热门的 MAS 研究还包括 MAS 在 Internet 上的应用、移动 Agent 系统、电子商务、基于经济学或市场学的 MAS 等。移动 Agent 可以自主地在网络上从一台主机移动到另一台主机上连续运行,这种灵活性为网络环境,尤其是 Internet 环境下的应用程序(如电子商务)提供了很多潜在的优点。目前已经有不少较为成功的移动 Agent 系统^[6]。

2.1 MAS 与联合意图

在前面我们讨论了关于单个 Agent 的意识态度的表示和形式化处理等问题。对于 MAS,除了考虑上述因素外,还要考虑多个 Agent 意识态度之间的交互问题,这是 MAS 理论研究的重要部分之一。

能够对环境中其他 Agent 的意识态度进行推理是 Agent 间共存、竞争或协作的要求,Agent 之间的协同、协商和协作行为是在其各种精神状态的支配和控制下才产生、进行和完成的。与共享精神状态相关的理论主要涉及到相互信念、联合目标和联合意图等概念,其中以联合意图为代表。在多 Agent 环境下,相互信念即是所谓的公共知识,它与多 Agent 通信密切相关。

从目前的研究现状来看,对联合意图的研究不外乎以下两种观点:(1) 宏观观点,即在每个个体 (Agent) 之外存在一种联合意图控制整个组织或社会 (MAS) 的协作行为;(2) 微观观点,即每个成员的意识态度导致个体动作一起完成协作目标。Haddadi 提出的关于联合意图的社会/组织观点和个体观点即分属于上述两种观点^[7]。Jennings 提出的关于联合意图的承诺和公约 (convention) 也分别对应于上述两种观点。Bratman 和 Rao 等人对联合意图的要求则只限于微观观点,他们认为,每个团队 (team) 成员都对联合行为作出承诺,联合意图是实现共享联合目标的方法,成员间要相互承诺,并及时把各自动作的成功或失败通知给其他成员。

下面给出一些与联合意图相关的典型的形式化工作。Rao 和 Georgeff 等人基于单个 Agent 系统的 BDI 模型递归定义了社会 Agent 的概念,引入了社会规划。Wooldridge 和 Jennings 以承诺和公约概念为基础,用分枝时序逻辑建立了协作问题求解系统的形式化框架。Haddadi 基于 BDI 逻辑给出了一个协作系统的形式化模型,而且还说明了该模型在 COSY 系统中的应用^[8]。

2.2 多 Agent 协商

协商 (negotiation) 是 MAS 实现协同、协作、冲突消解和矛盾处理的关键环节。有关多 Agent 协商的关键技术可以概括为协商协议、协商策略和协商处理这 3 个方面的内容。

协商协议的主要研究内容是 Agent 通信语言 (ACL) 的定义、表示、处理和语义解释。Müller 认为^[9],协商协议的最简单形式是如下一条协商通信消息:(〈协商原语〉,〈消息内容〉)。其中协商原语即消息类型,它的定义通常基于言语行为理论。消息内容除包含消息的发送者、接收者、消息号、发送时间等固定信息外,还要包括与协商应用的具体领域有关的信息描述。协商协议的形式化表示方法通常有 3 种:巴科斯范式表示、有限自动机表示和语义表示。巴科斯范式表示具有简洁、明了的特点,是最常用的表示方法。采用纯语义表示的协商工作不多,研究者更多的是给出非形式化的语义解释。ACL 中最著名的是 KQML 语言^[10]。协商策略是 Agent 决策和选择协商协议和通信消息的策略。协商策略包括一组与协商协议相对应的元级协商策略和策略的选择机制或函数两部分内容。协商策略基本上可以分为 5 类:单方让步、竞争型策略、协作型策略、破坏协商和拖延协商。后两类策略显然不利于推进协商进程,而单方让步策略只在协商陷入僵局或协商不再有意义时才起作用,所以只有竞争型和协作型策略才是有意义的。竞争型策略一般是指协商参与者坚持自己的立场,在协商过程中表现出竞争行为,使协商结果向有利于自身利益方向发展的协商对策。合同网协商模型、劳资协商、基于对策论的协商过程等都属于此类。协作型策略则是指协商各方都从系统整体利益出发,在协商过程中互相合作,它们采取的协商对策有利于寻找互相能接受的协商结果。采用协作型策略的协商过程包括部分全局规划 (PGP)、FA/C 等。Agent 应动态、智能地选择适宜的协商策略,从而在系统运行的不同时刻表现出不同的竞争或协作行为。策略选择的通用方法是依据影响协商的多方面因素,给出适宜的策略选择函数。策略选择函数可能包括效用函数、比较或匹配函数、兴趣或爱好函数等几种。策略选择函数的设计除了要综合考虑影响协商的各种因素之外,还要考虑冲突综合消解以及与应用领域有关的属性等。

协商处理包括协商算法和系统分析两部分内容.协商算法用于描述 Agent 在协商过程中的行为,包括通信、决策、规划和知识库操作等.系统分析的任务是分析和评价 Agent 协商的行为和性能,回答协商过程中的问题求解质量、算法效率以及系统的公平性和死锁等问题.

协商协议主要处理协商过程中 Agent 间的交互,协商策略主要涉及 Agent 内的决策和控制过程,而协商处理则侧重于对单个 Agent 和多 Agent 协商社会的整体协商行为的描述和分析.前两者描述了多 Agent 协商的微观方面,而后者则刻画了多 Agent 协商的宏观层.

有关 MAS 协商的典型工作有下面几个. Sycara 以劳资协商为背景对非协作类的多 Agent 相互作用进行了研究,给出基于实例推理和多属性效用优化理论的“劝说性辩论”模型. Grosz 等人在会谈理解研究中,给出支持人机交互通信的形式化模型. Wellman 等人把面向市场方法用于设计 Agent 间的协调过程,提出所谓基于一般平衡理论的“面向市场程序设计”机制.还有基于经济学理论、对策论和 Nash 平衡理论的多 Agent 协商研究等.

2.3 多 Agent 规划

规划是连接精神状态与动作执行之间的桥梁,有关动作和规划的研究一直是 Agent 研究的活跃领域. MAS 中的规划与经典规划不同,属于适应性规划(reactive planning),需要反映出环境的持续变化.

目前对 MAS 中规划的研究主要从如下两个不同的角度进行:(1)将规划看作是一种可以在世界状态间转换的抽象结构,典型地,如与或图;(2)将规划看作是一类复杂的 Agent 精神状态.这两种方法都在一定程度上降低了经典规划中解空间搜索的代价,从而有效地指导了资源受限 Agent 的决策过程.其中第 1 种方法的应用更广.常用的方法是将 Agent 的规划库定义为一个与或图结构,其中的每一条规划包括如下 4 个部分:(1)规划目标,这是规划的点火条件,表示该条规划能达到的目标;(2)规划前提,表示该规划被执行前必须满足的环境或状态条件;(3)规划体,是规划的程序部分,由规划序列和规划子目标组成;(4)规划结果,表示执行规划后对环境或状态的更新结果.

如何在自私的多 Agent 动态环境中实现 Agent 的灵活通信和动作执行,是当前 MAS 规划研究的热点.目前,多数 MAS 规划方法存在以下的不足之处:多 Agent 规划不太适合动态变化的环境;很多形式化较好的理论模型与实际距离太远;如何更好地解决多 Agent 规划中的资源冲突;如何在多 Agent 规划中引入质量因素等等.这些方面也是现在、乃至将来 Agent 规划研究的方向和趋势.

3 Agent 技术的应用:面向 Agent 的程序设计(AOP)

研究 AOP 的目的是为了在实践中更好地应用 Agent 技术. AOP 的概念最早是由美国 Stanford 大学的 Shoham 教授于 1993 年提出的^[11].他认为,AOP 是一种基于计算的社会观点的新兴程序设计风格和计算框架,其主要思想是利用 Agent 理论研究提出的能表示 Agent 性质的意识态度来直接设计 Agent 和对 Agent 编程. AOP 具有便于描述、能嵌套表示以及超陈述性编程等优点.

在使用 AOP 开发 MAS 时要做的工作包括 Agent 系统的规范、实现和验证. Agent 系统的规范是指 Agent 的表示,包括 Agent 所应包括的意识态度、所采用的结构、Agent 动作的描述、Agent 从信念到动作的转化方法(如采用规划)等.通常采用带量词的多模态逻辑来描述,有关这方面的讨论请见本文 1.2.1 节. Agent 系统的实现是指如何构造一个满足指定规范要求的系统,通常有两种方式:一种是直接执行逻辑语言,如 Concurrent MetateM 中所采用的方法;另一种是通过编译手段将逻辑语言转化为可执行的形式,典型地,如情景自动机(situated automata)的方法. Agent 系统的验证是指验证所实现的系统是否满足规范要求,主要途径有两种:语义方法(模型验证)和公理化方法(定理证明).有关这方面的详细介绍请参见文献[4].

目前,有关 AOP 的研究已经不再局限于把意识态度作为 Agent 程序的核心,其含义更加广泛,泛指能设计和建造 Agent 系统的各种软件技术和开发方法,包括各种开发工具和实验床.即使是 AOP 语言也并非均是基于 Shoham 的 AOP 思想,即不以 Agent 的意识态度为核心,这类 AOP 语言称为反应型语言,典型地,如 General Magic 公司开发的用于构造 Agent 社会的脚本语言 Telescript^[12].同时,将 Shoham 的以某种 Agent 逻辑为基础的 AOP 语言称为思考型语言.思考型 AOP 语言的代表是 Shoham 的 AGENT-0^[11],它包含一个直接与时间相

关的多模态逻辑和与该逻辑相对应的 Agent 语言. AGENT-0 阐明了 AOP 思想的精髓,但过于简单.为了更详细地示例 AOP,Thomas 对 AOP 做了进一步的研究,给出了一个称为规划通信 Agent(PLACA)的 AOP 语言,PLACA 增加了对动作和高层目标进行规划的操作符及意图状态修改规则,但在功能上仍然有限.目前比较成功的 AOP 语言有 AgentSpeak(L)和 April.前者是一个基于 BDI 逻辑的受限一阶 Agent 规范描述语言,它允许设计者用该语言直接描述 Agent 的行为(如信念、期望和意图等)和相互作用;后者是一个基于对象的并发语言,支持 Agent 间通信、模式匹配和符号处理等能力,适合于建造 DAI 和 MAS 的应用平台.

另外一类有代表性的 AOP 语言是将 OOP 技术与 AOP 思想相结合而构造的混合型 Agent 开发语言,如 DAISY, LALO(AGENT-0+KQML+C++)等. 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.

4 结束语

尽管面向 Agent 技术在复杂系统的开发和实现中起着重要的作用,但 Agent 技术本身也存在着一些固有的缺陷.首先,用 Agent 技术开发的整个系统将是不确定的.由于 Agent 的自治性使其可以在一定程度上自由地作出决策,所以 Agent 之间的关系无法得到准确的预测和管理.其次,整个系统的性质和行为在系统的设计阶段不能确定.虽然设计者可以给出个体 Agent 的行为规范,但整体行为只有在运行时才能体现.

Wooldridge 等人在文献[7]中从政治、管理、概念、分析和设计、微观角度(Agent 级)、宏观角度(社会级)、实现这 7 个方面论述了在使用 Agent 技术开发应用程序时所存在的种种误区. Jennings 和 Nwana^[13]也多次提到了在应用 Agent 时所存在的一些问题.总结起来,最主要的问题有:(1)缺乏良好的用户使用界面;(2)缺乏明确的 MAS 应用系统设计规范和方法论,在现有系统中有很多是设计者将其主观经验加到传统 OOP 技术后的结果,但 Agent 技术要想成功地走向商业化,必须具备一套完整的结构化方法;(3)缺乏强有力的 Agent 互操作和 MAS 开发工具,尤其是缺乏软件可重用工具;(4)缺乏社会范围内的支持.这些问题是 Agent 技术所面临的挑战,也是未来 Agent 技术研究的主要方向.

参考文献

- Jennings N R, Sycara K, Wooldridge M J. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1998,1(1):275~306
- Wooldridge M J, Jennings N R. Intelligent agent: theory and practice. *Knowledge Engineering Review*, 1995,10(2):115~152
- Yang Kun, Liu Da-you. Agents: properties and classifications. *Computer Science*, 1999,26(9):30~34
- Wooldridge M J. Agent-Based software engineering. *IEEE Transactions on Software Engineering*, 1997,144(1):26~37
- Haddadi A. Reasoning about cooperation in agent systems: a pragmatic theory [Ph.D. Thesis]. Manchester, UK: University of Manchester Institute of Science and Technology (UMIST), 1995
- Rao A S, Georgeff M P. BDI agents: from theory to practice. In: Georgeff M P ed. *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95)*. San Francisco. ACM Press, 1995. 312~319
- Wooldridge M J, Jennings N R. Pitfalls of agent-oriented development. In: Sycara K P, Wooldridge M J eds. *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*. New York: ACM Press, 1998. 385~391
- Dale J. A mobile agent architecture for distributed information management [Ph.D. Thesis]. University of Southampton, 1997
- Müller H J. Negotiation principles. In: O'Hare, Jennings N R eds. *Foundation of Distributed Artificial Intelligence 6th Generation Computer Technology Series*. New York: John Wiley & Sons Inc., 1996. 211~229
- Finin T, Labrou Y, Mayfield J. KQML as an agent communication language. In: Bradshaw J ed. *Software Agents*. Cambridge: MIT Press, 1997
- Shoham Y. Agent oriented programming. *Artificial Intelligence*, 1993,60(1):51~92
- White J E. Telescript technology: the foundation for the electronic marketplace. White Paper, General Magic Inc., 1994

13 Nwana H. Software agent: an overview. Knowledge Engineering Review, 1996,11(3):205~244

Agents: Present Status and Trends

LIU Da-you YANG Kun CHEN Jian-zhong

(Department of Computer Science Jilin University Changchun 130023)

(Open Symbol Computation and Knowledge Engineering Laboratory of State Education Department Jilin University Changchun 130023)

Abstract This paper discusses and summarizes the key technologies, current status and developing trends of agent design and construction from both theory and practice views. The topics are as follows: (1) the properties, representation and reasoning of intelligent agent, (2) the joint-intention, negotiation and planning of multi-agent system, and (3) agent-oriented programming (AOP). The problems and challenges faced by the agent technology are also discussed.

Key words Intelligent agent, multi-agent system (MAS), agent oriented programming (AOP), intentional stance, DAI.