

多 Agent 系统的理论、技术及其应用^{*}

Theories, Technologies and Applications of Multiagent Systems

胡舜耕 张莉 钟义信

(北京邮电大学信息工程系 北京 100876)

Abstract In this paper, we introduce the background yielding multiagent systems, the present situation of study of their theories and technologies, and the applications in information processing on Internet and telecommunications and so for. In the same time, we analyze existing primary problems and provide some basic ideas solving them. Finally, we give some important studying directions in the future.

Keywords Agent, Multiagent system, Theory, Technology, Application

一、引言

分布式人工智能(DAI)主要研究在逻辑上或物理上分散的智能系统如何并行地、相互协作地进行问题求解。DAI是在传统AI的基础上发展起来的,分为分布式问题求解(DPS)和多Agent系统(MAS)两个方向,多Agent系统被看成是人工智能的实验平台,成为AI的研究热点之一。

那么,什么是Agent呢?虽然不同的研究者可能会有不尽相同的观点和看法,但一种较普遍的观点认为,Agent是一个计算机系统,它的主要特性是自主性、反应性、适应性和社会性。自主性是指Agent能在没有人或其它Agent直接干涉下持续运行,并能控制其动作和内部状态。反应性是指Agent能感知环境,并采取适当的行动以改变环境。适应性是指Agent能够对环境的变化作出反应,在适当的时候采取面向目标的行动,以及从其自身的经历,所处的环境和其它Agent的交互中学习。社会性即Agent能以平等的方式与其它Agent交互。

从Agent的概念不难看出,Agent区别于传统智能系统的显著特性在于它具备与其所处环境,其它Agent和人进行交互、协调和协作的能力。人工智能的研究目标是认识与模拟人类智能行为,单个Agent的研究往往将重点集中在对人类单个智能品质如计算能力、推理能力、记忆能力、学习能力、直觉

能力等的研究与模拟,然而人类智能行为是各种单个智能品质的综合体现,所以单个Agent往往无法充分刻画或恰当地模拟人类智能行为,再者AI已经成熟,它正致力于攻击那些更复杂、更现实、更大规模的问题。这些问题的解决超出了单个Agent的能力。单个Agent的能力受到它所拥有的知识、计算资源和世界视图的限制。更现实的问题是开放的,分布式系统的出现,如Internet,在一个开放系统中,信息资源、通信连接和Agent是不可预期地出现和消失的。目前,Internet上的Agent大多从事信息获取,信息过滤和信息收集等工作,这些能力要求各个Agent能够互操作,并在平等的交互中相互协调,所有这些都越来越清楚地表明,我们应该把研究重点放在多Agent系统而不是单Agent的研究上。

下面我们将概要地介绍多Agent理论和技术的若干研究方向,指出存在的主要问题,提出了一些解决问题的思路,阐述了多Agent系统在Internet信息处理和通信等领域中的应用,并展望了未来的研究方向。

二、多Agent系统

多Agent系统是一个松散耦合的Agent网络,这些Agent通过交互解决超出单个Agent能力或知识的问题。其中的Agent是自主的,它们可以是不同的个人,采用不同的设计方法和计算机语言开

^{*} 本项研究得到863高技术计划基金资助。

发而成的,因而可能是完全异质的。多 Agent 系统具有如下特性^[1]:(1)每个 Agent 拥有解决问题的不完全的信息或能力;(2)没有系统全局控制;(3)数据是分散的;(4)计算是异步的。人们期望多 Agent 系统在复杂的、开放的分布式问题求解领域大展身手。

1. 多 Agent 系统理论

单 Agent 理论的研究通常是基于带多个量词的多模态逻辑。这些量词用来描述信念、期望、意愿、意图、目标、承诺等心智状态。这一研究领域的两个最著名的成果是 P. R. Cohen 和 H. J. Levesque 的意图理论^[2]以及 M. Wooldrige 的一种基于分支时间的表示逻辑^[3]。相对而言,多 Agent 系统理论的研究尚不多见。

多 Agent 系统理论的研究是以单 Agent 理论研究为基础的。除单 Agent 理论研究所涉及的内容以外,还包括一些和多 Agent 系统有关的基本规范,主要有如下几点:①多 Agent 系统的定义;②多 Agent 系统中 Agent 心智状态包括与交互有关的心智状态的选择与描述;③多 Agent 系统应具有哪些特性;④这些特性之间具有什么关系;⑤在形式上应如何描述这些特性及其关系;⑥如何描述多 Agent 系统中 Agent 之间的交互和推理。

不难想象,如果 Agent 能够通过推理知道其它 Agent 的心智状态,预测它自身的行为将对全局产生的影响,并预期其它 Agent 的行为,这将有利于增加多 Agent 系统的一致性和稳定性,并将大大增强多 Agent 系统的智能。我们也有可能通过引入推理机制,帮助探究系统产生冲突的根源,从而尽量避免或解决冲突。许多 AI 研究工作者致力于为理性 Agent 建立起一套形式化公理,藉此研究理性 Agent 推理的方法和机制。公理化是利用信念、愿望、意图、目标等 Agent 心智状态来形式化 Agent 行为模型而得到的。这些工作就是著名的 BDI(believe-desire-intention)系统。具有 BDI 型结构的 Agent 称为主动型的。主动型的 Agent 遭到了 R. A. Brooks 的批评。R. A. Brooks 认为智能是 Agent 与其所处环境相互作用的结果,他提出了无需表示和推理的智能。正是基于这样的考虑,人们构造了反应型 Agent,它们没有其所处环境的表示,利用刺激-响应行为工作。

复杂的系统不能没有推理能力,没有推理能力的系统是低智能的。我们可以综合主动型 Agent 和反应型 Agent 两者的优点,针对不同的智能层次的行为采取不同的处理方式,设计出复合型 Agent。此外,多 Agent 系统的推理方法和技术研究也是摆在

我们面前的紧迫课题。

2. 多 Agent 系统的体系结构

体系结构的选择影响有多少协作智能存在于单 Agent 自身内部。体系结构的选择影响异步、一致性、自主性和自适应性的程度。它决定信息的存储和共享方式,同时也决定 Agent 之间的通信方式^[4]。有些体系结构允许基于内容的消息发送,有些体系结构把发送消息功能从单个 Agent 分离出来形成集中的路由器。体系结构中必须有共同的通信协议或传递机制。另外,有些体系结构集中存储领域级信息,有些则通过 Agent 的局域数据库分布式地存储这些信息。对于特定的应用,应该选择与其能力要求相匹配的结构。下面我们简要介绍多 Agent 系统的几种常见体系结构。

(1)Agent 网络。在这种体系结构中,不管是远距离的还是近距离的 Agent 之间都是直接通信的。对这种类型的 Agent 框架,通信和状态知识都是固定的,每个 Agent 必须知道消息应该在什么时候发送去什么地方,系统中有哪些 Agent 是可以合作的,都具备什么样的能力等。分布式智能设计环境 DIDE 就是一个应用 Agent 网络结构的多 Agent 设计系统。但是,将通信和控制功能都嵌入每个 Agent 内部,就要求系统中的每个 Agent 都拥有有关其它 Agent 的大量信息和知识,而在开放的分布式系统中,这往往是做不到的。另外,当系统中 Agent 的数目越来越大的时候,这种一对一的直接交互将导致低效率。

(2)Agent 联盟。联盟不同于 Agent 网络,若干相距较近的 Agent 通过一个叫做协助者的 Agent 来进行交互,而远程 Agent 之间的交互和消息发送是由各局部 Agent 群体的协助者 Agent 协作完成的。这些协助者 Agent 可以实现各种各样的消息发送协议。当一个 Agent 需要某种服务时,它就向它所在的局部 Agent 群体的协助者 Agent 发出一个请求,该协助者 Agent 将以广播方式发送该请求,或者将该请求与其它 Agent 所声明的能力进行匹配,一旦匹配成功,则将该请求发送给匹配成功的 Agent。同样地,当一个 Agent 产生了一个对其它 Agent 可能有用的信息的时候,它通知它所在的局部 Agent 群体的协助者 Agent,该协助者 Agent 通过匹配,将此信息发送给对它感兴趣的 Agent。这种结构中的 Agent 不需要知道其它 Agent 的详细信息,因此较 Agent 网络结构有较大的灵活性。协助者 Agent 能够实现一些高层系统服务,如白页,黄

页、直接通信,问题分解和监控等。在 M. R. Cutkosky 等的系统中利用了 Agent 联盟结构。

(3)黑板结构。这种结构和联盟系统有相似之处,不同的地方在于黑板结构中的局部 Agent 群共享数据存储—黑板,即 Agent 把信息放在可存取的黑板上,实现局部数据共享。在一个局部 Agent 群体中,控制外壳 Agent(类似于联盟中的协助者)负责信息交互,而网络控制者 Agent 负责局部 Agent 群体之间的远程信息交互。S. E. Lander 等给出了一个黑板结构应用的例子。黑板结构的不足之处在于:局部数据共享要求一定范围的 Agent 群体中的 Agent 拥有统一的数据结构或知识表示,这就限制了系统中 Agent 设计和建造的灵活性。因此,开放的分布式系统不宜采用黑板结构。

3. 任务分派

所谓任务分派就是分配任务和问题求解资源给一个 Agent。在任务分派过程中,一个重要工作是最小化任务的相关性,这有两点好处:第一,减少问题求解 Agent 之间的通信量,提高问题求解效率。第二,减少潜在的冲突,提高解的一致性的几率。

任务分派是 DAI 研究领域中的一个早期研究问题之一。对多 Agent 系统的灵活任务分派早在 1983 年就受到了 R. Davis 和 R. G. Smith 的注意,他们的工作就是著名的合同网协议。在这个协议中,Agent 被动态分配为两种角色:管理者和合同者。一个 Agent 接受一个任务,从而成为管理者,其它 Agent 自动成为合同者。管理者首先将任务分解为需要共同完成的若干子任务(如果必要的话),然后利用合同网协议向能够执行这些子任务的 Agent 发送任务,且请求投标。接收到发送任务的 Agent 根据自身的状态和能力以投标作为应答,以表明它完成该任务的能力和意图。管理者在规定的期限内收集所有的投标,根据投标者的承诺,把任务分派给最好的投标者。

合同网协议实现了动态任务分配,允许 Agent 在同一时间为多个任务进行投标,提供自然的负载均衡(忙的 Agent 不必投标)。它的局限性是不能识别和解决冲突;对被拒绝的投标者,管理者不予通知;在任务执行过程中没有优先机制等。它也不适应其中的 Agent 是自私的多 Agent 系统。T. Sandhoim 和 K. Sycara 对合同网协议作了扩充。T. Sandhoim 和 V. Lesser 引进了不遵守承诺的惩罚措施,K. Sycara 利用金融选择定价理论获得了在不确定性环境中的灵活定约方案。

4. 冲突消解

多 Agent 系统中每个 Agent 具有自主性,在问题求解过程中会按自己的目标、知识与能力进行活动,常会出现矛盾和冲突。特别是对多 Agent 系统来说,我们可以根据新技术和不可预期的某些要求改变或重组 Agent 集,这种灵活性使得通过知识工程来解决冲突成为不可能。我们不太可能在设计或实现一个 Agent 时就把它建造成与所有其它潜在的也许此时还未曾构想的 Agent 相一致。所以,对于多 Agent 系统而言,动态的冲突管理是一种必然的要求。当然,我们在设计多 Agent 系统时应设法避免冲突,这也是不可忽视的。

在多 Agent 系统中解决冲突的主要方法就是协商,协商技术包括讨价还价、重构、放松限制、调解和仲裁。协商技术通常是基于对策论的,如 J. S. Rosensechein 的工作。它假定 Agent 是无所不知的,并根据最大化效用的原则选择自己应采取的行动,且 Agent 的效用矩阵是共享的知识。但是,现实世界中的协商是在不完全信息条件下进行的,它应该是根据多准则,而不是单一的效用准则。协商者并非无所不知,其效用并非共享而是私有的,要模拟和处理现实世界中的协商问题,我们需要新的理论。C. Castelfranchi 等提出了通过建立社会规则来避免冲突的设想。但是社会规则 and 标准会妨碍多 Agent 系统的灵活性和适应性。如果说 Agent 模拟的是人类智能,那么多 Agent 系统就是模拟人类社会。人类社会不能没有规则,但即使人人都遵守规则,也总是难免有冲突发生,因为社会是在发展变化的,而在某一时期制定的规则不可能适应所有变化了的情况,何况规则很难完善,所以现实中的协商是必要的。我们认为,冲突管理既需要有协商机制和技术,也应该包括社会规则。

5. 多 Agent 系统的通信

通信的内容、数量、类型和合适的通信时间都会影响多 Agent 系统问题求解的一致性。在一个动态的,开放的环境中,Agent 可以是高度异质的,因此我们面临的关键问题之一就是 Agent 的互操作性。开放环境中的异质 Agent 若要相互协调和协商就会面临两个挑战:首先,它们必须能够找到其它 Agent;其次,它们必须能够相互操作。

对寻找其它 Agent 的问题,K. Decker 等采用了中介 Agent 的方法。由于开放环境中 Agent 的出现和消失具有随机性,加上有可能出现通信中断,采用中介 Agent 有利于多 Agent 系统互操作的稳定性。

为了解决 Agent 之间互操作的问题,我们需要一种通信语言。有两种流行的设计通信语言的方法^[5]:过程方法和声明方法。过程方法是基于这样一种思想:通信能由过程指令的交换来模拟。如 TCL, Telescript 等就是基于这种方法的。过程通信语言的缺点是:第一,设计过程有时候需要接受者信息,而发送者却不一定具备这些信息。第二,过程是单向的,而 Agent 共享的许多信息应该双向可用。如有时候我们要从 a 求 b,有时候又要从 b 求 a。声明方法是基于通信能由声明语句(如定义、假设等)的交换来模拟的思想。R. Neches 等定义的 Agent 通信语言 ACL 就是基于声明方法的,ACL 由三部分组成:词汇、内部语言 KIF (Knowledge Interchange Format) 和外部语言 KQML (Knowledge Query and Manipulation Language)。

ACL 词汇是由一本大规模的、开放的词典给出的。词典中每个词都有人类使用它时对其意义的理解描述,也有程序中使用的该词意义的形式解释(用 KIF 语言书写)。词典的开放性有利于对它的扩充,人们可以把特定领域或新的应用领域的新闻添加到词典里。

KIF 是一阶谓词演算的前缀形式,为提高其表述能力而作了许多扩充。它提供了简单数据、约束、否定、选言、规则、量化表示、元信息等的编码。尽管我们可以设计一个通信框架,其中的所有信息都采用 KIF 语句的形式,但这将导致低效,因为 KIF 语义的上下文独立性,每条消息都不得不包括诸如发送者、接受者、消息历史时间等明确信息。为了提高通信效率,ACL 语言提供了考虑上下文相关性的语言层,这就是 KQML 的功能。KQML 定义了一种 Agent 之间传递信息的标准语法以及一些动作,这些动作主要是从 Speech Act 理论中演化来的。它提供了一套标准的通信原语,使得使用这种语言的 Agent 之间都可以进行交流和共享知识。

ACL 已被应用于几个大规模的软件可操作示例中,结果表明它是一种有应用前景的语言。然而,要设计一种通用的通信语言是一件非常艰巨的工作,因为通信语言不是语法层面的语言,而是语义层面的语言。我们认为,通信语言可以设计成一种层次结构语言。首先按照语义相关性或任务相关性划分问题求解领域,语义相关性或任务相关性大的属于同一领域,再根据领域特点设计与之相适应的高效通信语言。然后再逐步向上归约,最终得到一种统一的高层通信语言。

6. 多 Agent 学习

开放分布式多 Agent 系统的结构和功能都是非常复杂的。对许多应用而言,要想在设计多 Agent 系统的时候准确定义系统行为以使其适应各种需求是非常困难的,甚至是不可能的。这就要求多 Agent 系统具有学习能力。对于高级智能动物的人况且需要不断学习以适应变化了的环境,何况多 Agent 系统! 学习能力应成为智能系统的重要特征之一。

在人工智能领域,单 Agent 学习已有几十年历史。近年来,单 Agent 学习有了新的发展,以 Internet 为实验平台,设计和实现具有某种学习能力的用户接口 Agent 和搜索引擎 Agent 为大家所关注。B. Krulwich 等研制的信息发现 Agent 能从用户在浏览时发送给它的样本文件学习用户兴趣,并用经验的方法从中抽取最能表达文件内容的短语。T. R. Payne 和 P. Edwards 设计并实现了一个能监视和学习用户兴趣并根据用户兴趣过滤电子邮件的邮件 Agent 接口。A. Caglayan 等实现了一个用户接口学习 Agent 和一个跨平台的学习引擎—Learn Sesame。他们的学习 Agent 应用复合型的神经专家结构从用户那儿获取知识。他们利用神经网络方法实现知识获取,而推理和知识维护是通过专家系统技术实现的。现在的基于语法方法的搜索引擎有两个主要的不足:一方面,许多包含有同一关键词但内容并不相关的文本都被搜索并提供给用户。另一方面,许多内容相关的文本因为不包含所提供的询问的关键词而被拒绝提取。C. V. Goldman 等实现了一个能提供高效文本信息搜索的系统。该系统包括两部分:第一部分是 Musag,这个 Agent 学习把意义上相似的概念联系起来,并动态地为每一个给定的概念建立一部词典。第二部分是另一个 Agent,叫 Sag,它根据赋权的一组关键词获取文本。这样的系统能够获得包含有通过 Musag 学习到的与用户所提供的关键词相关的词汇的文本。

和单 Agent 学习相比,多 Agent 学习还是个比较年轻的领域,但发展速度是很快的。单 Agent 学习是多 Agent 学习的基础,许多 Agent 学习方法是单 Agent 学习方法的推广和扩充。事实上,前面所提到的用户接口 Agent 和搜索引擎 Agent 中所涉及的学习都可以认为是多 Agent 学习,因为在人机共生系统中,人也是一个 Agent。当然,多 Agent 学习比 Agent 学习复杂得多,有两个基本的理由:第一,Agent 的学习对象处在动态变化之中。第二,多 Agent 学习离不开 Agent 之间的通信,而这要付

出代价的。特别是在实时系统中,这是一个不容忽视的问题。

目前,在多 Agent 学习领域中,强化式学习(Reinforcement Learning)和在协商过程中引入学习机制引起了越来越大的兴趣^[6]。RL 结合了动态编程和有监督学习两个领域,以期产生强大的机器学习系统。在 RL 系统中,我们只给计算机一个要达到的目标,然后它通过不断地和环境进行交互以达到这个目标,许多研究者是这种机器智能的追随者,并为它在解决以前未能解决的大规模的复杂问题中的潜力而感到兴奋^[7]。D. Zeng 和 K. Sycara 提出了一个经济竞价协商模型,其中的 Agent 是自私的。模型强调学习功能,其中的 Agent 保存交互的历史信息,在观察它们的环境和其它协商 Agent 行为以后,应用贝叶斯方法更新它们的信念^[8]。

多 Agent 学习任重而道远,我们面临如下的挑战性的研究课题:多 Agent 学习的概念和原理;具备学习能力的多 Agent 系统的模型和体系结构;从单 Agent 学习方法到多 Agent 学习方法的推广和扩充;新的适应多 Agent 系统学习特征的新方法研究;多 Agent 系统的多策略和多观点学习;多 Agent 系统的组织自设计;多 Agent 学习的理论分析。

三、多 Agent 系统的应用

多 Agent 系统的应用研究开始于八十年代中期,近几年呈明显增长的趋势,范围涉及工业制造业、工业过程控制、空中交通控制、噪声控制、地面交通管理、农业、电信网和计算机网络管理、数据库、远程教育、远程医疗、信息提取、Internet 网上信息处理等广阔领域。

值得注意的是多 Agent 系统在电信、网络管理等领域的应用日见其多。R. Weismayer 和 H. Velthuisen 把一个与呼叫建立有关的实体表示成 Agent,通过这些 Agent 之间的协商来处理特征交互问题。一旦发现冲突,Agent 之间通过协商解决冲突,以便建立呼叫连接。通信领域中其它的基于 Agent 的应用包括网络控制、传输和交换、通信网业务管理和网络管理等。此外,J. Van Den Akker 和 A. Siebes 应用 Agent 技术建造主动型数据库^[9],通过引入推理能力来提高主动型数据库系统的适应性和灵活性。

目前的 Internet 实质上是一个巨大的开放分布式的信息资源聚集地。面对如此海量的信息资源,人们已经着手研究网上信息处理的手段和方法。Inter-

net 已成为多 Agent 技术天然的实验平台,在前面关于多 Agent 学习的讨论中我们已经看到,在 Internet 智能用户接口和 Internet 智能搜索引擎研制中,多 Agent 技术发挥了重要作用,而且潜力很大。N. M. Moussalle 等把 Agent 技术应用于智能教学系统开发。M. P. Singh 利用多 Agent 技术建造远程教学和健康信息系统^[10]。可以预期,在网上智能学校和网上智能医院的设计和开发中,多 Agent 技术将发挥潜在的不可估量的作用。Agent 技术还被应用于 Internet 网上信息的过滤、评估和集成等。

小结 设计和建造 Agent 系统是困难的。建造传统的分布式协同系统所遇到的问题在这里都会出现,但对多 Agent 系统的构筑而言,关键问题还在于如何实现自主 Agent 之间的灵活复杂的交互。下面是我们面临的几个重要的研究课题:(1)多 Agent 理论研究。由于没有合适的分析工具,目前所研制的多 Agent 系统大都是经验性的,这妨碍了多 Agent 技术和应用的发展。如何选择合适的心智状态,如何理清多 Agent 系统不同方面特征之间的关系,如何描述多 Agent 系统中 Agent 之间的交互和推理是多 Agent 系统理论中的关键问题。(2)高效的多 Agent 协调和协商技术。这是多 Agent 技术得以成功应用于解决实际问题的关键,请注意,在这里我们用的是“高效”而没有用“完备”,完备的多 Agent 协调和协商技术不一定高效,因而不一定实用。高效的多 Agent 协调和协商技术的关键又在于设计一种高效的开放的通信语言。通信语言的标准化将为高效地设计开发开放的,分布式多 Agent 系统创造便利条件。(3)多 Agent 学习方法研究,这个问题我们在文中第二部分中进行了阐述。(4)设计 Agent 系统的方法学研究。面向 Agent 的软件工程是一项复杂的系统工程,完全经验性的设计和开发是低效的。设计多 Agent 系统的方法学应使系统设计者能够把他们的应用清晰地表达和结构化成一个多 Agent 系统。(5)研制工业级的多 Agent 系统设计开发工具箱。一种技术能被成功地应用于解决实际问题的标志之一是拥有一整套辅助设计开发工具。对多 Agent 系统的设计开发而言,使设计者能够说明 Agent 的问题求解行为以及 Agent 应该怎么和在什么时候进行交互,可视化和调试 Agent 和整个系统的问题求解行为,这些都需要有相应的工具。(6)以应用研究带动多 Agent 理论和技术的发展和完善。多 Agent 系统是应复杂的,开放的和分布式的问题

(下转第 39 页)

$22) \wedge (c=24) \Rightarrow (d=4)$ 。这可以通过图 3 的 Rough 图直接得到。然而如果采用基于不分明关系的 Rough 集,那么决策表将被“过细”划分,共可导出 12 条确定规则(除 u_7 和 u_8 合并外,其它均单独构成一条规则)。

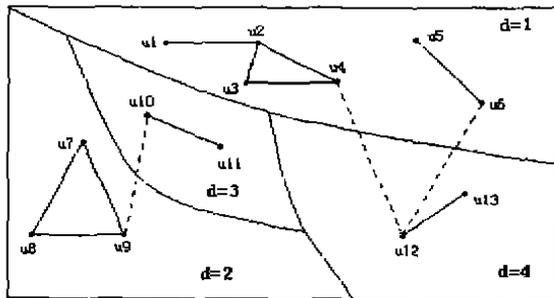


图 3 基于相似关系的 Rough 图

结束语 Rough 超图以图论的形式对 Rough 集重新进行定义,可有效地提高决策效率,方法简洁、直观明了。该方法为把图论知识引入 Rough 集的研究作了有益的尝试;文章还对比了不分明关系和相似关系对于 Rough 近似集的影响。我们认为进一步对 Rough 超图进行研究很有意义,例如将相似关系中的对称特性不予保留,那么超图就可以采用有向图模型来描述。另外,对其相似度量方法也有待

进一步研究。

参考文献

- 1 Pawlak Z. Rough sets—Theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht, 1991
- 2 Pawlak Z. Rough sets. Intl J. of Information and Computer Science, 1982, 11(5), 341~356
- 3 Electronic bulletin of the rough set community Available at <http://www.cs.uregina.ca/~roughset/>, Department of Computer Science, Regina University, Canada
- 4 Slowinski R, Vanderpooten D. A generalized definition of rough approximations. [ICS Research Reports 05/96]. Institute of Computer Science, Warsaw University of Technology, Warsaw, 1996
- 5 Klerowski M, Stepanuk J. Selection of objects and attributes—A tolerance rough set approach. [ICS Research Reports 54/95] Institute of Computer Science, Warsaw University of Technology, Warsaw, 1995
- 6 Krawiec K, et al. Construction of rough classifiers based on application of a similarity relation. In: Proc of the fourth intl workshop on rough sets, fuzzy sets, and machine discovery (RSFD'96). Tokyo, Japan, 1996

(上接第 24 页)

求解的需要而产生的,只有在解决实际问题的不断实践中,才能逐步地丰富,发展和完善多 Agent 理论和技术,同时多 Agent 理论和技术也需要在实践中得到检验。

主要参考文献

- 1 Sycara K P. Multiagent Systems. AAAI, 1998(Summer):79~92
- 2 Cohen P R, Levesque H J. Intention is choice with commitment AI, 1990, 42(2):213~261
- 3 Wooldridge M. Time, Knowledge, and Choice (Preliminary Report) In: Wooldridge M, et al., eds. Intelligence Agents I: Agent Theories, Architectures, and Languages. Springer-Verlag Berlin Heidelberg: Germany, 1996, 79~96
- 4 Lander S E. Issues in Multiagent Design Systems. IEEE Expert, 1997(March-April):18~26
- 5 Genesereth M R, Ketchpel S P. Software Agents.

CACM, 1994, 37(7):48~57

- 6 Gu P, Maddox A B. A Framework for Distributed Reinforcement Learning. In: Weiß G, Sen S, eds. Adaption and Learning in Multi-Agent Systems. Springer-Verlag Berlin Heidelberg: Germany, 1996, 97~112
- 7 Available at: <http://dir.yahoo.com/Science/Computer-Science/Artificial-Intelligence/Machine-Learning/>
- 8 Zeng D, Sycara K. Benefits of Learning in Negotiation. In: Proc. Of the National Conf. on Artificial Intelligence (AAAI-97). Menlo Park, Calif: AAAI, 1997, 36~41
- 9 Van Den Akker J, Siebes A. Enriching Active Databases with Agent Technology. In: Kandzija P, Klusch M, eds. Cooperative Information Agents. Springer-Verlag Berlin Heidelberg: Germany, 1997, 116~125
- 10 Available at: <http://www2.ncsu.edu/eos/info/dblab/www/mpsingh/research/>