

多 Agent 系统体系结构

Multi-Agent System Architecture

颜跃进 李舟军 陈跃新

(国防科技大学计算机学院 长沙410073)

Abstract The ability of Multi-Agent System to resolve complex, open and distributed problems makes it one of the recent hotspots of AI research. In this paper, we briefly introduce the concept of Multi-Agent system architecture that has been divided into two parts, namely individual architecture and society framework. And then we analyze and compare some types of the two parts.

Keywords Agent, Multi-Agent System (MAS), Architecture, Individual architecture, Society framework

1. 引言

基于 Agent 的系统是目前计算机科学领域中一个非常重要和活跃的研究方向,代表了一种新的方式和途径,可用于概念化、分析、描述和实现复杂而庞大的系统,因而体现了一种新的软件开发范型。这种范型在开发诸如 Internet 这样的分布、开放环境中的软件系统时尤其具吸引力。

近年来, AI 研究者通过对单 Agent 理论、技术和工具的研究,掌握和理解了单个认知实体的一些行为和推理特性。然而,随着网络和分布计算技术的发展,一些现实系统往往异常复杂、庞大,并呈现出分布式特性,以致于单 Agent 因个体所拥有的知识、计算资源和视图的限制而力不能及。多 Agent 系统(MAS)提供了一种解决复杂问题的分而治之的方法。当一个问题特别复杂、庞大、不可预测时,唯一的解决办法是将问题划分,构造一些有一定功能的 Agent,由这些 Agent 去分别处理子问题。当子问题之间出现互依赖性时,系统中的 Agent 必须通过相互协商与合作来控制互依赖性。

多 Agent 系统可以定义为“a loosely-coupled network of problem solvers that work together to solve problems that are beyond their individual capabilities”^[6]。这些问题求解器就是一个一个的 Agent,它们是具有交互性、反应性和主动性等特征的自主计算实体。实际系统中的 Agent 可由不同开发者在不同时间用不同工具和技术实现,各具不同领域不同程度

的问题求解能力。多 Agent 系统一般具有个体信息不完全、个体能力受限、无全局控制、数据分散化和计算异步等特点^[6]。

与传统单 Agent 的 AI 研究相比,开始于80年代初的 DAI 研究近几年来增长较快,其中对多 Agent 系统的研究更是发展迅速,究其原因,不仅是为了解决单 Agent 所不能解决的规模庞大、结构复杂的应用问题,而且是基于以下几个方面考虑^[3,4]:

1) 如何有效地利用现有软件系统来促进新的软件系统的开发是目前人们所关心的一个重要问题,多 Agent 系统研究者试图通过对现有软件系统进行包装,使之成一个多 Agent 系统^[6]。其中每个 Agent 有其特定的资源和能力。不同的 Agent 之间通过合作和协商等手段来实现资源和能力的共享。

2) 如果一个问题域涉及多个物理或者逻辑上分布的问题求解实体,每个问题求解实体仅拥有问题求解所需的有限数据、信息和资源,不同的问题求解实体之间必须相互交互才能最终求解问题。对这类问题,多 Agent 系统中 Agent 的自治性以及 Agent 之间的合作、协同等特征为我们提供了一种自然的建模方式:现实世界中的实体可以被直接映射为 Agent,每个 Agent 都有一定的资源和信息,它们相互交互以完成问题求解。

3) 多 Agent 系统可以从计算高效性、可靠性、可扩展性、健壮性、可维护性和灵活性六个方面来提高软件系统的性能。

那么,怎样构造多 Agent 系统?多 Agent 系统应

颜跃进 硕士生,研究方向为多 Agent 系统与分布式人工智能, **李舟军** 副教授,研究领域为 Agent 理论、计算机理论, **陈跃新** 副教授,研究领域为:分布式人工智能。

该具有怎样的体系结构?本文举例介绍了多 Agent 体系结构的研究和发展,并对它们进行了分析和比较。

2. 多 Agent 体系结构

体系结构就是对要完成的特定任务进行特定分解的一般方法学。Agent 体系结构研究的是怎样为构造 Agent 按一定方法将 Agent 分解成各种功能模块。

多 Agent 系统是 Agent 群体,其中的各 Agent 通过相互协商与合作来实现多 Agent 系统的功能。因此,多 Agent 体系结构分为个体 Agent 体系结构和 Agent 群体社会结构两部分。其中个体 Agent 体系结构是指在多 Agent 系统中单个 Agent 的构造结构,它与单 Agent 体系结构研究的内容联系紧密;Agent 群体社会结构通过定义角色和相互关系为 Agent 提供了一种交互框架。

2.1 个体 Agent 体系结构

多 Agent 系统中每个 Agent 都是处于一定的环境中,通过感知外界环境(包括 Agent 群体中的各种事件)并进行一些推理来决定完成某些动作,然后执行这些动作。Agent 的智能也主要是体现在这些推理上。推理依据和推理的熟练程度决定了个体 Agent 的体系结构。

主动型体系结构 又称为基于逻辑的体系结构。传统 AI 研究工作者致力于为理性 Agent 建立起一套形式化公理,公理化是利用信念、愿望、意图、目标等 Agent 心智状态(表示成逻辑公式)建立 Agent 行为的形式化模型而得到的。这种 Agent(体系结构如图1)通过传感器感知外界环境,并将其表示成 Agent 的某种心智状态(信念),然后从这些心智状态出发,经过依据公理的逻辑推理来决定所执行的动作。不难看出,这些推理过程就是定理的证明过程,而 Agent 就相当于一个定理证明器。

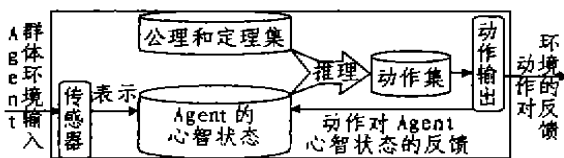


图1 主动型 Agent 体系结构

以著名的 BDI(believe—desire—intention)体系结构^[7]为例,我们简要介绍一下基于这种结构的 BDI Agent 的推理过程和组成部件。BDI Agent 的推理过程如图2所示,从图中我们可以看出,这种 Agent 由以下几个部件组成:

信念库:当前信念是 Agent 对环境和其它 Agent 信息在内部的逻辑表示。信念库是 Agent 当前信念的

集合。

意愿库:当前意愿是 Agent 由当前信念和意图决定的认为(或希望)自己能够执行的动作。意愿库是当前意愿的集合。

意图库:意图库是当前意图的集合。当前意图表示 Agent 对自己和其它 Agent 承诺将要执行的动作。

信念产生器:接受传感器感知的外界环境和其它 Agent 的信息作为 Agent 的输入,并结合当前信念库的内容产生新的信念。

意愿产生器:根据外界环境的内容在不违背已有承诺的前提下决定 Agent 能够完成的动作,也就是基于当前信念库和意图库的内容产生新的当前意愿。

意图产生器:是一个能体现 Agent 主动性的部件,它根据外界环境的内容和 Agent 能够完成的动作在不违背已有承诺的前提下(避免意图冲突)决定 Agent 承诺的动作,即新的当前意图的产生要基于信念库、意愿库和意图库。

动作执行器:执行 Agent 承诺动作的部件。

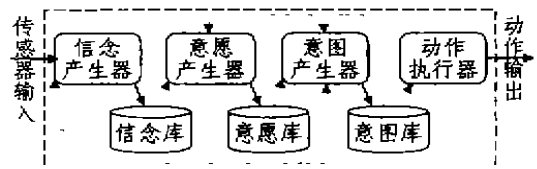


图2 具有 BDI 结构的 Agent

不难想象,在基于逻辑的体系结构下,如果理性 Agent 能够通过推理知道其它 Agent 的心智状态,预测它自身的局部行为对全局产生的影响,并预期其它 Agent 的行为,这将有利于增加多 Agent 系统一致性和稳定性,并将大大增强多 Agent 系统的智能。而且,我们也因此有可能在系统中引入冲突检测机制,以达到尽量避免或解决冲突的目的。

基于逻辑的体系结构的主要缺陷在于不能很好地适应实时环境,这是因为推理过程的计算复杂性致使 Agent 不能在实时系统的要求下有效地得出结论。另外,对复杂、动态的外界环境的解释和表示也是 AI 研究者们正在进一步探讨的问题。

反应型的体系结构 基于逻辑的体系结构遭到了以 R. A. Brook 为代表的一些 AI 研究者的反对,他们认为智能应该体现在 Agent 与环境的相互作用上,因而拒绝表示环境和基于这些表示进行推理。因此,人们设计出了具有反应型结构的 Agent,它们没有对外界环境的表示和繁杂的推理,利用刺激—响应行为工作。

R. A. Brook 提出的包容体系结构是最著名的反应型体系结构。这种结构的 Agent 具有两个重要特

征:一是它所具有的行为直接以传感器感知的外界环境信息为点火条件,中间不需要逻辑表示和推理。二是将它的所有行为分为优先级不同的几种类型,比较急迫、原始的行为被安排在优先级较高的类型中,而较为次要、抽象的行为被安排在优先级较低的类型中。

反应型 Agent 只需对它所处环境的当前状态作出反应,而不必考虑历史因素,也不用为将来制定计划。只要反应型 Agent 之间进行简单的交互,就能组织复杂的全局行为,这样,一群反应型 Agent 在完成某个任务时,不会因为某一个 Agent 的失败而失败。因而,健壮性和容错性是反应型体系结构的两个重要特点。另外,反应型 Agent 速度快,适用于实时环境。

纯反应型 Agent 的缺陷在于它基于局部信息做决定,而不考虑非局部信息,也无法预测自己的局部行为将对全局造成什么影响。不难想象,在多 Agent 系统中,这种近视的行为将对系统造成无法预测的后果。

·层次型体系结构 在实际的应用环境中,仅有基于逻辑的 Agent 或仅有反应型 Agent 是不够的,对于既需要有一定推理又需要满足实时要求的复杂系统,我们可以综合主动型 Agent 和反应型 Agent 两者的优点,对于不同智能层次的行为采用不同的处理方式,构造层次型 Agent。

一种显然的构造层次型 Agent 的结构是分为两层,一层是逻辑层,一层是反应层。逻辑层用逻辑将环境表示成 Agent 的心智状态,并能进行熟练的推理;反应层能对环境中的不需要推理的事件直接作出反应。多 Agent 系统中的层次型 Agent(图3)一般是三层结构,最底层是反应层,中间层是逻辑层,最上层是合作层(或社会层)。合作层主要刻画和处理 Agent 之间的合作。具体地说,它要接受其它 Agent 的消息,把其转换成 Agent 的内部表示,并进行一定的推理决定执行某些动作来对该消息作出反应。另外,怎样合作与交互在具体的结构中有不同的实现。

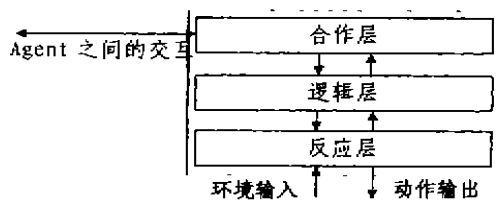


图3 MAS 中层次型 Agent 体系结构

在 AI 领域,对层次型体系结构的研究是 Agent 体系结构研究者当前感兴趣的重点之一。然而,我们很难评价哪一种体系结构一定优胜,因为每一种结构都有它适用的范围。在设计多 Agent 系统时不仅要构造个体 Agent 的体系结构,而且怎样使这些 Agent 交

互与合作、为 Agent 群体选取合适的社会结构也至关重要。

2.2 Agent 群体社会结构

一般说来,Agent 群体社会结构是指存在于各 Agent 中的通讯和控制模式,通过定义 Agent 之间的权威关系,为 Agent 提供了一种交互框架。体系结构的选取不但影响单 Agent 内部协作智能的多少,而且也会影响多 Agent 系统中异步、一致性、自主性和自适应性的程度。体系结构不但决定了 Agent 之间的通讯方式,也决定了系统中信息的存储和共享方式。有的社会结构中每个 Agent 均可发送和接收消息,而有的将消息机制从单个 Agent 中分离出来形成集中的通信 Agent。在同一种体系结构中,必须有共同的通讯协议或消息机制。另外,有些社会结构集中存储领域级信息,而有些则将这些信息分布地存储在单个 Agent 的局部数据库中。在选取 Agent 群体社会结构时,要根据不同的应用,选择与其能力要求相适应的结构。下面简要介绍一下多 Agent 系统中几种常见的社会结构,并对它们在控制结构、通讯、信息存储等方面进行了分析和比较。

1. 网络结构 这是一种平板结构,Agent 网络社会中设有特权者,每两个 Agent 之间都是对等的关系。通信对等、信息局部化是这类结构的显著特点。在这种体系结构中,无论距离远近,Agent 之间的通讯都是直接的(如图4)。*[Sandholm 1993]*系统中的网络市场结构和*[Lewis 和 Sycara 1993]*系统中的网络专家群结构用的都是这种体系结构。这种体系结构不但要求 Agent 保存的信息量大,而且要求 Agent 之间的通讯链路多。系统中的 Agent 数目增多将直接导致系统的效率降低。

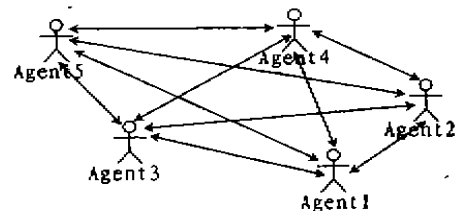


图4 Agent 网络结构

2. 层次结构(图5) Agent 群体中的 Agent 被分层,在每一层的决策和该层的控制权集中在其上层 Agent 上,由它(它们)控制和协调该层 Agent 的行为。同一层的 Agent 之间的通讯要经过上层。这种结构的 Agent 只需保存其下层 Agent 的相关信息,但通讯不如网络结构简略。

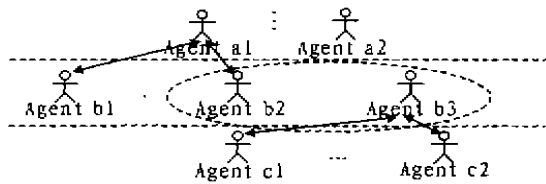


图5 Agent 层次结构

3. 联盟结构(图6) 将系统中的 Agent 根据某种方式(一般是根据距离远近,便于通信)划分成一个个的 Agent 联盟。联盟与联盟之间的关系与网络结构中 Agent 与 Agent 之间关系一样。每个联盟有一个协助 Agent,不同联盟中 Agent 之间的交互都是通过各联盟中的协助 Agent 协助完成,联盟尤其是按距离远近划分的联盟内部一般采用两层次结构,由协助 Agent 负责联盟中各 Agent 之间的协商与合作。在联盟内部,共享数据可根据需要选择分布式存贮或集中式存贮,但集中式数据存贮要求一定范围内的 Agent 群体中的 Agent 拥有统一的数据结构或知识表示。M. R. Cutkosky 等在系统中用到的联盟结构采用了分布式存贮,而 S. E. Lander 等利用黑板结构给出了集中式存贮应用的例子。

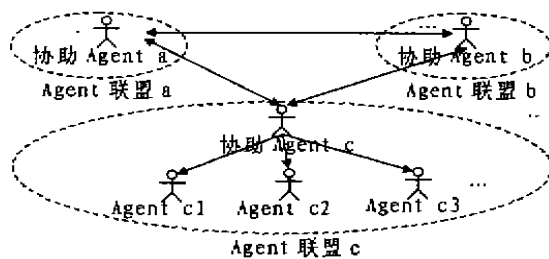


图6 Agent 联盟结构

结束语 设计和建造多 Agent 系统是困难而复杂的,多 Agent 系统体系结构是构筑多 Agent 系统的关键技术。在设计和实现多 Agent 系统体系结构和社会结构时,如何使设计和实现有效、快捷要依赖于以下两个方面的发展。

1)多 Agent 技术的标准化。标准化包括个体 A-

gent 体系结构、Agent 通讯、系统的开发方法学和开发工具等。多 Agent 技术的标准化必将使设计者们能够清晰、有效地将他们的应用问题描述和实现成多 Agent 系统。当前,Agent 技术的标准化的制定仍处于初级阶段,但已在个体 Agent 体系结构、通讯语言等方面有一定发展^[9]。

2)对多 Agent 应用的研究。多 Agent 系统是因求解开放、复杂和分布的应用问题的需要而出现。我们必须对问题的应用环境充分理解,才能为它构造合适的个体体系结构和社会结构。另一方面,我们构造好的多 Agent 系统也要在应用中接受实践的检验。

参考文献

- 1 胡舜耕,张莉,钟义信.多 Agent 系统的理论、技术及其应用. 计算机科学,1999,26(9):20~24
- 2 Sycara K P. MultiAgent Systems. AAAI,1998(Summer), 79~92
- 3 Wooldridge M. Intelligent Agents. In: MultiAgent system, a modern approach to distributed artificial intelligence, 1999 Massachusetts Institute of Technology. 1999. 27~77
- 4 Jennings N R, Wooldridge M. Application of Intelligent Agents. Agent Technology, Foundation, Applications, and Markets. 1998
- 5 Jennings N R, Sycara K, Wooldridge M. A roadmap of Agent research and development. Autonomous Agents and Multi-Agent Systems. 1998, 1:7~38
- 6 Durfee E H, Lesser V, Corkill D D. Trends in cooperative distributed problem solving network. In: Proc. IEEE Trans. Knowl. Cata Eng. 1989, KOE-11(1):63~83
- 7 Bratman M E, Israel D J, Pollack M E. Plans and resource-bounded practical reasoning. Computational Intelligence, 1998, 4:349~355
- 8 Wooldridge M, Jennings N R. Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 1995, 10(2):115~152
- 9 OMG Agent Working Group. Agent Technology Green Paper. 2000