

多 Agent 系统体系结构的构造技术研究*

Research on Construction Technology of Multi-Agent System Architecture

李陶深 甘雯
Li Taoshen Gan Wen

(广西大学计算机与信息工程学院 南宁 530004)
(Col. of Comp. & Info. Eng., Guangxi Univ., Nanning, 530004)

摘要 分析利用面向对象的技术构造一个多 Agent 系统的方法,提出了一个基于多 Agent 的知识管理系统的体系结构,讨论了相关的构造技术。

关键词 多 Agent 面向对象 知识管理 建模语言

中图分类号 TP315

Abstract The method to construct a multi-agent system using object-oriented technology is analyzed. An architecture of knowledge management system based on multi-agent is developed. The related techniques are discussed.

Key words multi-agent, object-oriented, knowledge management, modeling language

近年来,随着分布式人工智能技术、网络技术及多媒体技术的发展,人们对资源共享及分布式协作提出了更多的要求。开发基于知识的协同设计环境,以模拟和帮助人类专家进行知识和信息处理,使设计过程逐渐实现自动化已成为现代工业设计的目标。协同设计系统是通过多个 Agent 的协作来解决设计问题的。由于 Agent 在分布式环境下具有的自治、协作等特性,具有一定的知识及问题求解能力,因此,多 Agent 系统是设计过程逐渐实现自动化的有效途径。本文在介绍 Agent 的基本概念的基础上,分析利用面向对象的技术构造一个多 Agent 系统的方法,提出了一个基于多 Agent 的知识管理系统的体系结构,并讨论相关的构造技术。

1 Agent 的基本概念

1.1 Agent 的定义

正如在现实世界中人们通过形成一个群体相互协调行为以达到某种目的一样,一个多 Agent 系统是由多个可计算的 Agent 组成的,通过多个 Agent 的相互协作来解决需要求解的问

题。其中,每个 Agent 被认为是一个物理的或抽象的实体,不仅能作用于其自身,同时还可影响环境,操纵环境的部分表示,并且能与其它 Agent 通讯。为了达到此目的,系统中的每个 Agent 必须具备知识和能力来决定其自身的行为,并且使自己能够更好地适应环境。在非常复杂的情况下,由于单个 Agent 的功能有限或者和其它的 Agent 会有冲突,它必须在基于观察到的信息的基础上,有效地感知外界环境的影响,进而有能力对其它 Agent 的行为作出合理的预测,并以此来计划自己的动作。

目前,对于 Agent 还没有一个统一明确的定义。英国的 Agent 理论专家 Wldridge 和 Jennings 给 Agent 作出的定义是:Agent 是一个自主的程序,它能基于其对环境的理解,有能力控制自己的决策的行为,以追求达到一个或多个目标。它具有以下几个基本特征:

(1)自主性:Agent 是一种独立的主体,可将数据、过程和通信设施等封装于一体。Agent 根据状态自主地控制其行为,在没有人或其它程序介入时自主地运行和执行操作;

(2)应激性:Agent 能感知其所处的外部环境,并对环境发生的变化作出及时的响应;

(3)能动性:Agent 主动表现出目标驱动的行为,能自行选择合适时机采取适宜的动作;

(4)社会性:Agent 通过某种通信语言与其它 Agent (也可能与人)进行交流,特别是有能力积极参与到社会活动中,如协同问题求解或协商,以达到它们的目标。因此 Agent 应具有通信能力和协作协商能力,Agent 之间存在着相互依赖、相互制约的关系,它们的行为既有局部效应,又有全局效应;

(5)开放性:Agent 应具有进化能力,即能够在推理活动中适应外部环境的要求,动态地扩充、限制和修正自身的局部知识状态。

1.2 Agent 面向对象的表示

从形式化的角度并结合面向对象的技术,Agent 可用一个 5 元组表示:

$$\text{Agent} = (\text{ID}, \text{Mental}, \text{Rule}, \text{Action}, \text{Interface}),$$

其中 ID 为 Agent 的唯一标识符;Mental 为 Agent 的心态类,包含其能力、信念、承诺等精神状态。能力体现了它对自身能力的意识,信念表示得到的事实,而承诺则体现了与其它 Agent 间的关系。Agent 接收到的消息可以作为事实以信念的形式进行处理。

Rule 是 Agent 的行动规则类。对于知识 Agent 而言,它体现了对该领域知识进行推理的控制;而对于功能 Agent 而言,它体现了与各知识 Agent 间协作的知识,以及对用户请求的分解、对返回结果的综合。

Action 是 Agent 的执行动作类,体现了 Agent 的实际能力。在知识 Agent 中,它主要是负责使用推理机进行知识推理。在功能 Agent 中,它是负责对知识的综合处理。

Interface 类包括了 Agent 与用户或其它 Agent 之间进行交互的接口。

按照以上定义,可以将一个 Agent 的行为能力简单地概括为感知能力、对自身的控制能力和活动能力,其中各个 Agent 在问题求解过程中所担任角色可以归纳为解释、分类、转换和控制等等,所采用的问题求解方法可以包括传统的软件技术和基于知识的处理技术,甚至是由特定的人来亲自来处理。

2 一个基于多 Agent 的知识管理系统的体系结构

从系统的层次上分析,多 Agent 系统中存在 2 种层次:即 Agent 间进行协作的层次、Agent 内通过组件合作形成能力的层次。Agent 间的协作层是通过共享本体论和相互协商,结成动态

的开放的 Agent 协作联盟来实现的，它是在语义基础上的高层协作，是面向跨部门的联合目标、强调大范围内大粒度软构件的松耦合集成。在 Agent 内部，通过应用组件/软总线技术将不同的组件进行合成，实现 Agent 的一种能力，它是面向 Agent 的一项专门目标、强调小范围内小粒度软构件的紧耦合集成。

基于多 Agent 系统的理论，图 1 给出了一个基于多 Agent 的知识管理系统的体系结构。该系统按功能设计了 3 种 Agent：即功能 Agent、知识 Agent 和接口 Agent。每一个领域知识的处理都用一个知识 Agent 来完成，而每一种定性或定量的预测处理方法都有以 Agent 的功能性模块的方式实现。这样每个知识 Agent 都有自己的多种不同的预测处理方法来满足不同任务的需要，并具有新知识的获取与应用的能力、与其它 Agent 通讯合作的能力。

整个系统的知识是分层次进行分布存储的。对于仅涉及某个领域的知识，存储于相应的知识 Agent 中，而对于涉及多个领域知识进行合成的高层知识，就存储于相应的功能 Agent 中。功能 Agent 的知识是面向功能应用、面向用户需求的，一项新功能的开发，实际上是对应于功能 Agent 新知识的增加。而知识 Agent 主要是关于某个领域或某一类的知识进行有针对性的组织管理、存储积累和应用提取，从而可以使得领域知识获取与应用的效率大大提高，也有利于相关的知识应用方法的设计与实现。

接口 Agent 的设置主要是为了利用已有的各领域的专家系统，通过基于 KQML 语言的通信方式与其它 Agent 进行交互，实现协作。它在整个系统中的主要功能与知识 Agent 基本一致。接口 Agent 可以设置新的功能模块来对已有的这些领域知识进行处理，以满足要求。因此，在基本上不用作大改动的情况下，就可以应用这些已有的领域专家系统的知识。

系统数据库的信息包括各种 Agent 的注册信息，以及系统运行中所需要的一些资料。它对于所有 Agent 都是可访问的。

在系统中，每个 Agent 都可以与其它 Agent 进行合作，利用多 Agent 之间的有机合作可以实现定性方法与定量方法的综合集成。而随着该系统的不断运行，各个 Agent 将不断地获取知识提高自己的能力，从而使整体系统的智能程度和反映速度等性能得到大大提高。

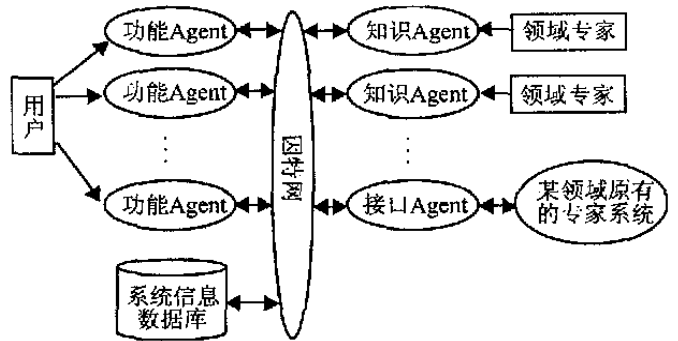


图 1 一种基于多 Agent 的知识管理系统的体系结构

3 多 Agent 系统的构造技术

3.1 多 Agent 系统的分析设计

目前，没有一种成熟的、现成的面向 Agent 设计（AOP）方法供我们对多 Agent 的应用系统系统进行分析设计，也没有一个基于 AOP 的软件开发环境供我们来描述用户需求、构造多 Agent 模型、刻划 Agent 特征，更没有基于 Agent 的编程语言供我们进行系统实现和测试。经过分析，我们认为面向对象方法与面向 Agent 要求在大体上是一致的，因此利用面向对象方法来分析设计面向多 Agent 的应用系统是可行的。其中的原因如下：

首先，随着应用领域的扩大，面向对象方法的表达能力也在不断在增强，因此可以逐步地满足面向 Agent 系统的应用需要。比如，最近主动对象（active object）的概念被引入面向对象方法中，用来描述那些不需要接收消息就能主动执行的对象。在应用面向对象方法分析

设计面向 Agent 的系统时,就可以用主动对象来描述具有主动性行为的 Agent 对象了。

其次,在面向 Agent 的应用中,可以从应用的需要出发,对系统中 Agent 的特性进行取舍,不必体现 Agent 的全部特性。比如,在我们设计的系统中,Agent 的可移动性对于系统的应用功能而言,意义不大,因此没有必要考虑对 Agent 可移动性的面向对象描述和设计。

第三,对于系统中必需体现的一些 Agent 特性,可以在面向对象方法与 Agent 特征之间作一个折衷,用一种面向对象扩充的方式进行设计实现,能满足系统需要即可。比如,如果系统中需要体现 Agent 能自动感知环境的应激性,用面向对象方法无法实现,但我们可以让 Agent 定时对其感兴趣的环境因素进行检测,并及时作出响应。

因此,我们认为,对于基于 Agent 的应用系统,可以用扩展的面向对象方法进行分析、设计、实现,这在一般情况下是可以达到系统需求的。同时,面向对象方法也是目前最成熟、最适合于对面向 Agent 系统进行分析设计的方法。

3.2 多 Agent 系统的建模策略

UML 语言 (Unified Modeling Language) 是一种可以用于对大型系统进行建模的统一建模语言,它不仅支持面向对象的分析和设计,还支持从需求分析开发的软件开发的过程,可以为任何具有静态结构和动态行为的系统进行建模。我们在开发多 Agent 系统时,选用了 UML 语言来进行建模设计,并采用以下全局的系统设计策略:

(1) 概念化。利用用例图 (use case diagram) 来分析问题,确定用户需求和技术解决方案,实现对问题作一个初始化的分割;

(2) 分析。首先对系统需求进行分析,通过黑盒来描述系统外部行为,利用用户可理解的方式来构建 UML 模型;然后通过完整性检测或手工模拟来对系统模型进行验证;最后得到能正确地反映系统需求的分析模型;

(3) 系统设计。制定关于系统实现的高层的全局决定和结构;

(4) 对象设计。首先通过将高层操作扩展成可行的操作来细化分析模型。然后确定一定的算法和数据结构,其中大多数设计决定应能扩展成为独立于语言的方式。最后得到逻辑上正确的实现并逐步转换成设计模型;

(5) 编程实现。将设计映射到具体的语言实现,例如用 Java 语言开发出可用的软件。

3.3 多 Agent 系统的建模实现

在多 Agent 系统建模设计实现时,需要重点解决以下问题:

(1) Agent 的主动行为能力的实现。通过在面向对象方法中引入主动对象的概念来解决。这样,我们在系统建模时就可以用对象表达问题域中事物的主动行为和系统中的每个主动任务,在系统的设计实现阶段,对象的主动服务可以被实现为一个能并发执行的、主动的程序单位,比如进程或线程。

(2) Agent 的协作协商能力的实现。在对象建模阶段和系统设计阶段,给 Agent 设计一些专门的接口,使 Agent 间能建立一个动态的、松散的协作关系。Agent 通过接口与外部、与其它 Agent 进行联系。Agent 间的联系可以是同步的,也可以是异步的。每个 Agent 使用相同的 KQML 消息原语,使用相同名称的接口进行处理。而且,在系统分析设计时,要把 Agent 的知识和能力,以本体论的形式进行描述,在实现时可以用数据库的形式表示。

(3) Agent 的推理和规范模型、自学习模型,可以在 Agent 的建模设计时,以组件的形式

(下转第 163 页)

- 2 Radziszowski S P. Small Ramsey numbers. *The Electronic Journal of Combinatorics*, 2000, DS1, 7, 1~36.
- 3 Kalbfleisch J G. Chromatic graphs and Ramsey's theorem [Ph D thesis]. University of Waterloo, January 1966.
- 4 Exoo G. On two classical Ramsey numbers of the form $R(3, n)$. *SIAM Journal of Discrete Mathematics*, 1989, 2, 488~490.
- 5 Kreher D L, Li Wei, Radziszowski S P. Lower bounds for multi-colored Ramsey numbers from group orbits. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 1988, 4, 87~95.
- 6 Robertson A. Some results in Ramsey theory [Ph D Thesis]. Department of Mathematics, Temple University, 1999.
- 7 McKay B D, Radziszowski S P. Subgraph counting identities and Ramsey numbers. *Journal of Combinatorial Theory, Series B*, 1997, 69, 193~209.
- 8 Su W L, Luo H P, Zhang Z Y et al. New lower bounds of fifteen classical Ramsey numbers. *Australasian Journal of Combinatorics*, 1999, 19, 91~99.
- 9 Exoo G. Some new Ramsey colorings. *The Electronic Journal of Combinatorics*, 1998, #R29 (5), 1~5.
- 10 Luo H P, Su W L, Li Q. New lower bounds of classical Ramsey numbers $R(6, 12)$, $R(6, 14)$ and $R(6, 15)$. *Chinese Science Bulletin*, 1998, 43, 10, 817~818.
- 11 Su W L, Luo H P, Li Q. New lower bounds of classical Ramsey numbers $R(4, 12)$, $R(5, 11)$ and $R(5, 12)$. *Chinese Science Bulletin*, 1998, 43, 6, 528.
- 12 Su W L, Luo H P, Li Q. Lower bounds for multicolor classical Ramsey numbers $R(q, q, \dots, q)$. *Science in China (Series A)*, 1999, 42, 10, 1019~1024.
- 13 Su W L, Luo H P, Li Q. New lower bounds for seven classical Ramsey numbers $R(k, l)$. *Journal of Systems Science and Mathematical Sciences*, 2000, 20 (1): 55~57.
- 14 Su W L, Luo H P, Shen Y Q. New lower bounds for classical Ramsey numbers $R(5, 13)$ and $R(5, 14)$. *Applied Mathematics Letters*. 1999, 12 (6): 121~122.
- 15 Wu K, Su W L, Luo H P. New lower bounds for 8 classical multicolor Ramsey numbers. *Journal of Guangxi Academy of Sciences*, 2000, 16 (2): 63~164.

(责任编辑: 蒋汉明)

(上接第 148 页)

进行描述。同时, 对于 Agent 的行为也要进行细化, 也有组件的形式描述。这样, 既可以便于系统资源的重用, 同时又有利于系统的更新换代。而近年发展迅速的、以 CORBA 和 DCOM 为代表的软构件/软总线技术, 则为异质组件的开发与“即插即用”提供了规范。

4 结语

本文仅仅对多 Agent 系统的构造技术进行了框架性的研究, 工作只是初步的, 仍有很多的问题需要解决。应该指出的是, 面向多 Agent 系统的体系理论和相关软件技术的发展, 将会对计算机应用领域产生深刻的影响。开展面向多 Agent 系统的技术研究是很有意义的。

参考文献

- 1 史忠植. 高级人工智能. 北京: 科学出版社, 1998.
- 2 甘雯. 基于 Multi-Agent 的农业专家系统在 Internet 上的系统设计. 南宁: 广西大学计算机与信息工程学院, 2000.

(责任编辑: 黎贞崇)