

## 基于角色和协作场景的 MAS 管理模型及应用

张志勇 普杰信 冯长远

(河南科技大学电子信息工程学院 河南 洛阳 471003)

**摘要** MAS 协作管理是实现 Multi-Agent 协同工作的关键理论与技术。本文通过引入基于角色的策略和 Agent 协作场景 ACS, 提出了一种面向 MAS 的多 Agent 协作管理参考模型 MACM<sup>2</sup> (Multi-Agent Collaboration Management Model), 并给出它在基于智能体的网络文本信息检索系统中的具体应用。该模型在实际系统中解决了协作管理的复杂性问题, 提高了 Agent 的协作能力, 同时也降低了系统开销。

**关键词** MAS 角色 ACS 协作管理

### MULTI-AGENT SYSTEM MANAGEMENT MODEL BASED ON ROLE & COLLABORATION SCENE AND APPLICATION

Zhang Zhiyong Pu Jiexin Feng Changyuan

(School of Electronic and Information Engineering, Henan University of Science & Technology, Luoyang Henan 471003, China)

**Abstract** MAS collaboration management is key theory and technology realizing Multi-Agent cooperative work. The paper introduces role-based policy and Agent Collaboration Scene, and proposes a multi-agent cooperative management reference model-- Multi-Agent Collaboration Management Model (abbr. MACM<sup>2</sup>) as well as its application of Agent-based text information retrieval on web. It solved the question of collaboration management complexity, improved cooperation ability, and decreased the system's spending in actual system.

**Keywords** Multi-Agent system Role Agent collaboration scene Collaboration management

## 1 引言

Multi-Agent System 理论与技术是分布式人工智能领域中前沿研究的热点<sup>[1]</sup>, 它主要研究多个 Agent 之间如何通过动态调度、相互协调、冲突消解等协作完成任务, 并保证 MAS 的高效性、安全性和稳定性。

目前 MAS 协作的研究主要集中在 MAS 的体系结构和具体实现上, 在传统的 MAS 中 Agent 的能力和任务是单一、固定的, 然而这并不适合 MAS 的动态性、复杂性和不确定性<sup>[1]</sup>。为此文献[2]提出了一种松散型的联邦式 MAS 体系结构, 从而实现分布式问题的求解; 文献[3]基于 RBAC 模型提出了一个多 Agent 协调模型和体系架构, 以及用于 MAS 相关的函数, 但缺少对该理论模型精确的形式化描述。

文献[4]面向 WFMS 创建了一个基于角色的多 Agent workflow 模型, 并用扩展的巴克斯范式描述了该模型。由于该模型并未涉及对多 Agent 协作管理的描述, 故缺少完备性。本文将结合 MAS 动态、复杂和不确定的协作环境, 引入基于角色的策略和 Agent 协作场景 ACS, 提出一种用于 MAS 的多 Agent 协作管理模型 MACM<sup>2</sup> 来解决 MAS 中缺少对 Agent 动态协作管理的不足, 并将其用在实际系统中。

Ability)、反应性 (Reactivity) 和仁爱性 (Benevolence) 等重要特性, 具体表现在 Agent 能够感知外界环境, 并能及时做出反应, 进而又影响实际环境等一系列行为活动上。

MAS 的协作求解问题的能力超过了单个 Agent, 这是 MAS 产生的最直接的原因。多 Agent 之间没有冲突的目标, 它们依靠相互作用和准确的信息通讯展示出一种导向目标的行为, 通过协作完成目标任务。MAS 的组织结构也有完全集中式、完全分布式和混合式三种组织结构<sup>[1]</sup>。由于 MAS 的复杂性和动态性, 因此 MAS 的协作管理将对于 Agent 社会中的协调、协商、协作活动至关重要。

基于角色的策略是一种符合人类现实组织环境, 具有中性的管理策略和思想, 它最早用于基于角色的访问控制 (RBAC) 研究领域<sup>[5]</sup>。RBAC 主要包括会话、用户、角色、许可和约束等概念, 其主要优势体现在利用“角色”这个相对稳定的概念, 使得许可分配和角色相联系, 而不是直接和用户关联, 从而简化了大量繁杂的授权管理; 再者, RBAC 策略遵循安全管理中两个规范的原则“最小特权原则”和“责任分离原则”, 以及本身所具有的“数据抽象原则”, 使其更易于实现企业级的管理策略与机制。上述 RBAC 特性适合于解决大规模分布式系统中实体管理的复杂性问题。

## 2 Multi-Agent 协作与基于角色的策略

智能主体 Agent 具有自治性 (Autonomy)、社会性 (Social

收稿日期:2005-03-26。教育部科学技术重点项目 (03081); 河南省自然科学基金基础研究项目 (200410464004)。张志勇, 讲师, 主研领域: RBAC 与访问控制, 智能 MAS。

### 3 MACM<sup>2</sup> 模型及其形式化描述

#### 3.1 MACM<sup>2</sup> 模型的构建

MACM<sup>2</sup> 是在基于角色的策略基础上通过引入 Agent 角色和动态的 Agent 协作场景(ACS)概念而提出的一种面向 MAS 协作管理的参考模型。基于角色的策略简化了 Multi-Agent 社会的协作管理,并且 Agent 可以通过在不同 ACS 中角色激活来动态更替新角色,从而获得新角色的协作能力,解决了传统 Agent 能力的单一性问题。同时在 ACS 动态环境上实施协作约束,又提高了 MAS 的安全性。该模型主要由主体 Agent、角色、社会、任务、对象、能力、约束、ACS 等部分构成,如图 1 所示。

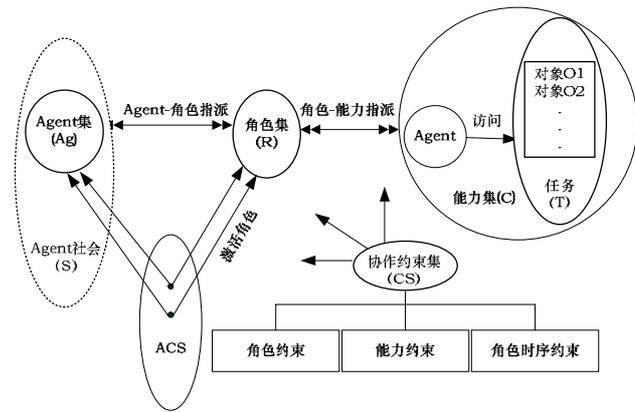


图 1 MACM<sup>2</sup> 参考模型

#### 3.2 MACM<sup>2</sup> 模型形式化描述

结合集合论和谓词逻辑形式化描述 MACM<sup>2</sup> 参考模型如下:

定义 Agent 主体集 Ag、角色集 R、社会集 S、任务集 T、对象集 O、能力集 C、协作约束集 CS 和 Agent 协作场景集 ACS。

**定义 1 (Agent)** Agent 是具有知识、信念、意向和期望等因素的实体,它能自主地感知环境的变化,并对此做出反应,从而进一步影响环境,是一个具有自治性和社会性的主体。Agent 可以表达为由 AID、感知器、信息处理器、通信组件、目标组件、效应器和协作场景等部分组成的七元组。

$$\forall ag (ag \in Ag) ag = \{ aid, sensor, processor, communication, aim, effector, acs \}$$

**定义 2 (角色 Role)** 角色是具有一定职责和能力的抽象实体,承担某个角色的 Agent 则成为该角色的一个实例,即角色可以看作是一组具有相同能力 Agent 的集合,它们两者是对多对的关系。

$$\forall r (r \in R) r = \{ ag1, ag2, \dots, agn \mid agi \in Ag \}$$

**定义 3 (社会 Society)** 社会是为同一任务目标由多个 Agent 或多个角色构成的组织单位。

$$\forall s (s \in S) s = \{ ag1, ag2, \dots, agn \mid agi \in Ag \}$$

$$\text{或 } \forall s (s \in S) s = \{ r1, r2, \dots, rn \mid ri \in R \}$$

**定义 4 (激活角色 Active Role)** Agent 在一个特定 ACS 中所启用的单个角色,用 AR(acs)表示在某个 ACS 中激活的角色。

$$\forall acs (acs \in ACS) (ar(acs) \in R(ag))$$

**定义 5 (ACS)** ACS 是多个 Agent 协同工作的动态场景环境,即协作上下文。它包括协作时间、空间、协作伙伴和工作对象等。Agent 工作是通过 AR 和 ACS 之间建立起映射函数 agentwork 来进行的。

$$\forall acs (acs \in ACS) acs = \{ time, space, parter, object \}$$

$$agentwork(ag,ar): Ag \rightarrow ACS$$

**定义 6 (角色指派 Role Assignment)** 根据协作任务为 Agent 指派适当的角色,使其具有一定的能力。角色指派关系 RA 是 Agent 与角色之间多对多的关系。

$$RA \subseteq Ag \times R$$

$$R(ag) = \{ r \mid \text{指派给某个 agent 主体 ag 的角色集, } ag \in Ag \}$$

**定义 7 (任务 Task)** 任务是一组协作 Agent (Agent 社会) 协同工作的目标。

**规则 1 (Agent 社会-任务)** 任意一个 Agent 社会有且仅有一个任务目标。谓词 t(s)表示社会 s 的任务。

$$\forall s (s \in S) \exists t1, t2 (t1, t2 \in T) t1(s) \wedge t2(s) \Rightarrow t1 = t2$$

**定义 8 (能力 Capability)** 能力是完成协作目标所应具备的特权、许可和条件。它可以是实际的文件读写和执行能力,也可以是系统定义的抽象的能力。Agent 角色可以看作是有限能力的集合,而完成某一任务目标需要多个 Agent 能力,任意一个 Agent 能力属于且仅属于某一个任务。这里定义函数 task 来表示能力和任务之间多对一的映射关系。

$$\forall r (r \in R) r = \{ c1, c2, \dots, cn \mid ci \in C \} task(c): C \rightarrow T$$

**定义 9 (能力指派 Capability Assignment)** 根据角色的职能(或承担责任)为其指派访问对象所应具备的适当工作能力。能力指派关系是角色、对象、能力之间多对多的关系,记为 PA=(R,O,C)。

$$PA \subseteq R \times O \times C$$

$$PA(r,o,c) = \{ pi \mid \text{指派给角色 r 关于对象 o 的能力 c, } r \in R, o \in O, c \in C \}$$

#### 3.3 基于 MACM<sup>2</sup> 的协作约束规则

为了有效地进一步增强 MAS 中多 Agent 协作能力和角色的管理,在 MACM<sup>2</sup> 参考模型中引入了协作约束,这保障了协作过程的安全性和高效性。该模型中的协作约束主要依据 RBAC 模型中的约束特征而创建的,它包括 MAS 管理基本策略和可自定义的规范约束规则等两类。其中前者主要包括“最小特权原则”和“特权分离原则”;后者则是依据角色或能力特征来生成和实施互斥角色约束规则或互斥能力约束规则。以下是主要规则的形式化描述:

**定义 10 (协作约束规则 Collaboration Constraint Rule)** Multi-Agent 协作约束规则由主角色、次角色和约束特征三元组构成。

$$cc = \{ mainrole, secondrole, character \}$$

**定义 11 (约束集 Constraint Set)** MAS 协作约束规则的有限集合,表示为 CS。

$$CS = \{ cc1, cc2, \dots, ccn \mid n > 0 \}$$

**定义 12 (互斥角色 Mutex Role)** 互斥角色约束是指在协作管理中不能同时指派给同一 Agent 的两个角色。

$$Mutex\_Role(ri) = \{ rj \mid rj \text{ 和 } ri \text{ 满足角色互斥, } i \neq j \}$$

**定义 13 (互斥能力 Mutex Permission)** 互斥能力约束是指在协作管理中不能同时指派给同一 Agent 的两个能力。

$$Mutex\_Capability(ci) = \{ cj \mid cj \text{ 和 } ci \text{ 满足能力互斥, } i \neq j \}$$

**规则 2 (互斥角色约束 Mutex Role Constraint)** 任意一组互斥角色都不能同时指派给任意一个 Agent 所构成的约束关系。

$$\forall ag, r1, r2 (ag \in Ag, r1 \in R, r2 \in R) r1 \in Mutex\_Role(r2) \Rightarrow R(ag) \in r1 \wedge R(ag) \in r2 = F$$

规则 3 (互斥能力约束 Mutex Permission Constraint)

任意一组互斥能力都不能同时指派给任意一个 Agent 所构成的约束关系。

forall r, c\_i, c\_j (r in R, c\_i in C, c\_j in C) c\_i in Mutex\_Capability(c\_j) => PA(r, o, c\_i) ^ PA(r, o, c\_j) = F

规则 4 (角色时序约束 Role Temporal Constraint) RTC

是 Agent 在特定 ACS 中角色激活的时序关系, 这种关系满足偏序性, 记为 "□"。

forall ag, ri, rj (ag in Ag, ri, rj in R) (ri in R(ag) ^ rj in R(ag) -> ri □ rj)
forall ag, ri, rj, rk (ag in Ag, ri, rj, rk in R) (ri □ rj ^ rj □ rk -> ri □ rk)

4 MACM² 在智能信息检索系统中的应用

本文所设计和实现的基于 Agent 的文本信息检索系统处于用户和网络信息资源之间, 根据用户的需求在网上搜索信息资源, 按照设定的规则进行过滤, 并以一定的优先方式提供给用户。由于单个 Agent 的任务功能过于集中、单一, 不利于多任务的并发执行和系统效率的提高, 因此采用多 Agent 协作作为完成智能信息检索的主要策略。该系统的协作管理是基于 MACM² 参考模型而构建的, 它按照 Agent 所承担的不同任务和能力, 具体定义了用户接口 Agent 角色、信息处理 Agent 角色和信息收集 Agent 角色等来实现对 Multi-Agent 的协作管理, 完成诸如 Agent-角色指派、角色-能力指派、协作约束生成和实施等工作, 从而使得多 Agent 共同完成用户兴趣学习、信息收集、信息过滤以及信息显示与反馈等功能。本系统中的 Agent 可以通过激活不同角色和其他 Agent 并行协同工作, 提高了系统效率并且解决了传统 Agent 任务单一的不足, 从而又减少了 Agent 的开销。其系统架构如图 2 所示:

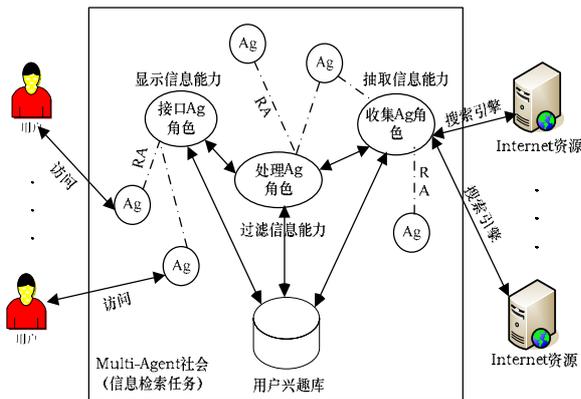


图 2 基于 Agent 和 MACM² 的网络文本信息检索系统架构

基于 MACM² 多 Agent 协作管理的基础上, 整个信息检索系统中多 Agent 协作过程如下:

- (1) 用户接口 Agent 获得用户提交的查询关键词或语句, 并提交给信息处理 Agent;
(2) 信息处理 Agent 对查询语句进行处理, 从本地信息库查找出相关的文档给用户, 或将处理结果传送给信息收集 Agent;
(3) 信息收集 Agent 携带处理过的查询请求借助通用搜索引擎进行相关搜索, 获取相关网页的链接存放在 URL 队列缓冲区;
(4) 信息收集 Agent 访问 URL 队列缓冲区, 取出链接地址将相关文本信息下载至本地, 由信息处理 Agent 参照用户的个性化模型进行过滤, 将过滤结果按照兴趣主题进行分类、索引

并存储到本地 WEB 信息库, 并在需要的时候提交给用户接口 Agent;

(5) 用户接口 Agent 将查找到的信息提供给用户, 收集用户可能提交的反馈信息;

(6) 信息处理 Agent 根据用户的反馈信息更新、完善用户的兴趣模型。

5 结论

本文所给出的 MACM² 是一种解决 Multi-Agent 协作管理复杂性问题较完备的参考模型, 通过 Agent 角色间的关系易于实现主从式、联邦式或独立式多 Agent 组织结构。该模型解决了基于智能体网络文本信息检索系统中多 Agent 协作管理的复杂性问题, ACS 和角色约束的引入提高了 Agent 协作能力, 同时也增强了 MAS 的安全性。对于 Multi-Agent 间如何自主委托角色以达到更有效地资源共享和协同工作, 需要做进一步的研究。

参考文献

[1] 王俊松, 崔世刚, "Multi-Agent 技术及应用", 《计算机工程与应用》, 2003, 39(18): 61~62, 66.
[2] 任明, 王成道, "基于联邦结构的多 Agent 协作", 《华东理工大学学报》, 2004, 30(3): 311~314.
[3] A. Omicmi, A. Ricci, M. Viroli, RBAC for organisation and security in an agent coordination infrastructure, Electronic Notes in Theoretical Computer Science, 2005, 128: 65~85.
[4] 赖会霞, 张仕, "基于角色的多 Agent workflow 模型", 《计算机应用》, 2004, 24(12): 316~318.
[5] R. Sandhu, E. Coyne, H. Feinstein, Role-based access control models, IEEE Computer, 1996, 29(2): 38~47.