

# 多 Agent 合作结构研究

李常洪, 李敏强, 寇纪淞

(天津大学系统工程研究所, 天津 300072)

**摘要:**总结了当前多 Agent 合作的研究现状, 然后从“是否完全合作”、“是否存在目标互换”、“合作双方的数量对比”等三个角度对多 Agent 合作结构进行了系统分类, 提出 16 种多 Agent 合作结构, 设计了一组多 Agent 通讯原语, 给出了多 Agent 合作的具体算法以及典型应用实例。

**关键词:**多 Agent 系统(MAS); 多 Agent 合作; 合作结构; 通信原语; 算法

**中图分类号:** TP18      **文献标识码:** A      **文章编号:** 1007-9807(2003)05-0006-06

## 0 引言

产生于分布式人工智能领域的 Agent 技术是 21 世纪人工智能相关领域研究的一个重点。Agent 真正有前途的领域是多 Agent 系统领域 (multi-agent system, MAS)。MAS 是由分布在网络上的多个 Agent 松散耦合而成的大型复杂系统, 这些 Agent 相互作用以解决由单一 Agent 的能力和知识所不能处理的复杂问题; MAS 也可以是基于提供服务 - 使用服务 (或者, 提供服务-中介-使用服务) 的市场机制, 由多个 Agent 组成的面向某一领域的大型复杂系统, 例如基于多 Agent 技术的汽车交易系统。

合作是 MAS 中最基本的特征, 整个系统比单个 Agent 具有更强的功能<sup>[1]</sup>。多 Agent 合作是 MAS 研究的一项重要内容, 其研究目标主要有三个: 通过 Agent 的合作去支持人类更好的合作; 通过 Agent 的合作去研究、解释人类社会的一些合作行为; 通过 Agent 的合作去研究解决一些复杂的优化问题。目前, 对多 Agent 合作的研究主要包括理论与应用两方面: 理论研究包括合作结构<sup>[2]</sup>、合作产生的过程<sup>[3]</sup>、合作的分类<sup>[4]</sup>、大规模自治系统的合作<sup>[5]</sup>、合作的进化<sup>[6]</sup>、协商<sup>[7,8]</sup>、联盟<sup>[9,10]</sup>等方面; 应用研究包括信息采集<sup>[11]</sup>、电子拍卖<sup>[12]</sup>、实

现网络映射的移动 Agent 的合作<sup>[13]</sup>等方面。

多 Agent 合作结构是指多个 Agent 进行合作时相互之间的关系, 它是多 Agent 合作研究的一个核心问题, 关系到多 Agent 系统的设计及实现等许多关键问题, 但很少有人对合作结构进行系统的研究。文献[2]提出两种多 Agent 合作结构: 不完全的合作结构; 完全的合作结构。其缺陷在于: 没有考虑目标互换; 没有考虑合作双方的数量关系。针对上述问题, 本文从“是否完全合作”、“是否存在目标互换”、“合作双方的数量对比”等三个角度对多 Agent 合作结构进行了系统分类, 提出 16 种多 Agent 合作结构, 设计了一组多 Agent 通讯原语, 并给出了多 Agent 合作的具体算法以及典型应用实例。

## 1 多 Agent 合作结构

文献[2]基于目标的分解 (目标  $g_1$  的分解如图 1(a) 所示) 提出两种多 Agent 合作结构: 不完全的合作结构 (图 1(b)); 完全的合作结构 (图 1(c))。  $g_i$  表示第  $i$  个目标,  $i$  表示 Agent  $i$ ,  $i_1$  要实现  $g_1$  目标, 但是, 自己无法实现。  $i_3, i_4, i_5, i_6$  可以分别独立地完成目标  $g_3, g_4, g_5, g_6$ ;  $i_2$  要实现目标  $g_2$  必须通过  $i_5$  完成目标  $g_5$ ,  $i_6$  完成目标

收稿日期: 2002-03-11; 修订日期: 2003-01-08。  
基金项目: 国家自然科学基金资助项目 (69974026; 70171002)。  
作者简介: 李常洪 (1973-) 男, 山西洪洞人, 博士生。

$g_6$  来实现. 图 1(b) 表示  $a_1$  通过把目标  $g_2, g_3, g_4$  分别授予  $a_2, a_3, a_4$  来帮助自己实现目标  $g_1$ , 但是, 由于  $a_2$  不具备实现  $g_2$  的能力, 因此这种合作是不完全的.

以上两种结构过于简单, 不能包括以下情况: 一对多合作, 多对一合作, 多对多合作, 目标互换

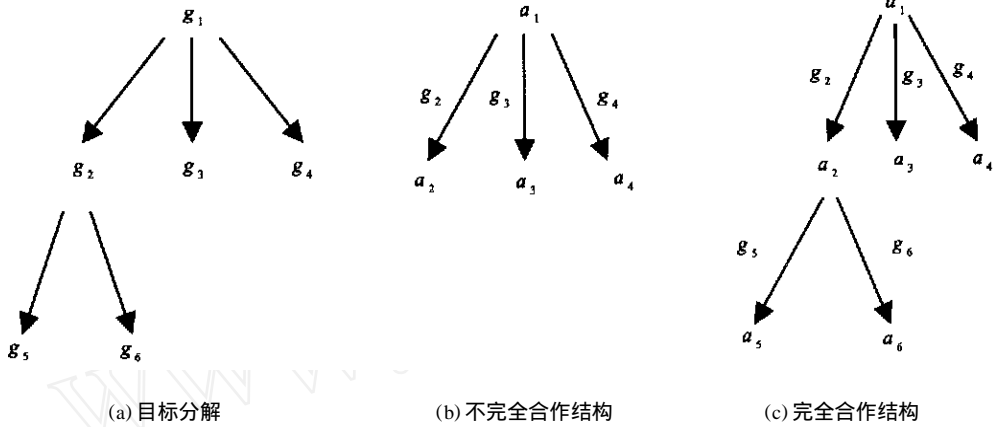


图 1 目标分解及合作结构

多对一合作: 具有相同目标的 Agent 群体(例如, 为了减少合作成本、提高合作效率), 通过与单个 Agent 的合作来实现相同的目标.

目标互换合作: 两个 Agent 分别把对方的目

合作.

一对多合作(多对多合作): 对于一个不可再分解的目标(单个 Agent 的目标, 或 Agent 群体的目标), 有时需要一组 Agent 共同去完成, 或者 Agent 群体去完成比单个 Agent 完成更加高效、高收益;

标当作自己的目标来实现;

综合以上的分析, 本文根据是否完全合作、合作双方的数量, 是否存在目标互换三个属性, 提出 16 种合作结构, 如表 1 所示.

表 1 16 种合作结构

				合作双方数量			
				一对一	一对多	多对一	多对多
目标互换	存在	完全合作	是	存在目标互换的 一对一完全合作	存在目标互换的 一对多完全合作	存在目标互换的 多对一完全合作	存在目标互换的 多对多完全合作
			否	存在目标互换的 一对一非完全合作	存在目标互换的 一对多非完全合作	存在目标互换的 多对一非完全合作	存在目标互换的 多对多非完全合作
	不存在	完全合作	是	不存在目标互换的 一对一完全合作	不存在目标互换的 一对多完全合作	不存在目标互换的 多对一完全合作	不存在目标互换的 多对多完全合作
			否	不存在目标互换的 一对一非完全合作	不存在目标互换的 一对多非完全合作	不存在目标互换的 多对一非完全合作	不存在目标互换的 多对多非完全合作

## 2 多 Agent 合作的通讯原语

言语行为(Speech Act)理论是多 Agent 研究领域为多 Agent 交互建立模型的基础. Speech Act 理论认为, 虽然自然语言的所有可能使用方式是无限的, 但言语所能执行的动作类型的数目是有限的. 言语行为可与语言所含的具体内容相分离, 并被分类以供研究使用. 本文提出了一组用于实现多 Agent 合作的言语行为作为通讯原语.

符号约定:

$i$  表示 Agent  $i$ ,  $(k_1, k_2, \dots, k_n)$  表示 Agent

联盟 ( $n > 1$ ) 或单个的 Agent ( $n = 1$ ),  $g_m^i$  表示  $i$  的第  $m$  个目标.

通讯原语: request ( $i, (k_1, k_2, \dots, k_n), g_m^i, [arguments]$ )

描述:  $i$  向  $k_1, k_2, \dots, k_n$  请求帮助实现目标  $g_m^i$ , arguments 是可选参数, 例如 date. 若 date 存在, 则表明实现目标有时间限制, 否则没有时间限制.

cncl.request( $a_i, (k_1, k_2, \dots, k_n), g_m^i$ )

描述:  $i$  向  $k_1, k_2, \dots, k_n$  发出关于  $g_m^i$  合作

请求取消的通知.

$sk(i, k, ?, g_m^i)$

描述:  $i$  向  $k$  发出寻求关于  $g_m^i$  的合作的请求.

$reply(k, i, (r_1, r_2, \dots, r_n, (e_1^1, e_2^1, \dots, e_{m_1}^1), (e_1^2, e_2^2, \dots, e_{m_2}^2), \dots, (e_1^s, e_2^s, \dots, e_{m_s}^s)), g_m^i)$

描述:  $k$  回答  $i$  提出的合作查询请求,  $(e_1^1, e_2^1, \dots, e_{m_1}^1), (e_1^2, e_2^2, \dots, e_{m_2}^2), \dots, (e_1^s, e_2^s, \dots, e_{m_s}^s)$  表示只有 Agent 联盟中的 Agent 共同合作才能完成  $g_m^i$ ,  $r_1, r_2, \dots, r_n$  表示能单独完成  $g_m^i$  的 Agent.

$receive\_firm(j, i, sk(g_m^i) \text{ or } cancel\_request(g_m^i))$

描述:  $j$  向  $i$  发出收到合作取消通知(或查询结果)的信息.

$accept(j, i, g_m^i \text{ or } exchange(g_m^i, g_m^j))$

描述:  $j$  接受  $i$  提出的  $g_m^i$  或者是关于目标交换的请求  $exchange(g_m^i, g_m^j), exchange(g_m^j, g_m^i)$  表示  $i$  帮助  $j$  实现  $j$  的目标  $g_m^j$ ,  $j$  帮助  $i$  实现  $i$  的目标  $g_m^i$ .

$reject(j, i, g_m^i \text{ or } exchange(g_m^i, g_m^j))$

描述:  $j$  拒绝  $i$  的  $g_m^i$  请求或  $exchange(g_m^i, g_m^j)$  建议.

$confirm(i, j, g_m^i \text{ or } exchange(g_m^i, g_m^j))$

描述:  $i$  向  $j$  确认按照  $g_m^i$  或者按照  $exchange(g_m^i, g_m^j)$  合作形成.

$with\_notice(j, i, g_m^i \text{ or } exchange(g_m^i, g_m^j))$

描述:  $j$  向  $i$  发出正在考虑  $i$  提出的  $g_m^i$  或者  $exchange(g_m^i, g_m^j)$  的合作请求.

$propose(j, i, exchange(g_m^i, g_m^j))$

描述:  $j$  向  $i$  提出合作建议  $exchange(g_m^i, g_m^j)$ , 其中,  $g_m^i$  为合作提出者的目标,  $g_m^j$  是合作对象的目标. 在实际的系统设计当中,  $g_m^i (g_m^j)$  可以是  $j (i)$  禁止或要求  $i (j)$  同某些 Agent 进行合作(这种方式可以看作一种惩罚或激励机制).

$coalition\_request(i, (i, k_1, k_2, \dots, k_n), g_m^i)$

描述:  $i$  关于目标  $g_m^i$  向  $k_1, k_2, \dots, k_n$  发出

结成联盟的邀请.

$accept\_coalition(k_1, i, (i, k_1, k_2, \dots, k_n), g_m^i)$

描述:  $k_1$  接受  $i$  关于目标  $g_m^i$  结成联盟  $(i, k_1, k_2, \dots, k_n)$  的邀请.

$reject\_coalition(k_1, i, (i, k_1, k_2, \dots, k_n), g_m^i)$

描述:  $k_1$  拒绝  $i$  关于目标  $g_m^i$  结成联盟  $(i, k_1, k_2, \dots, k_n)$  的邀请.

$notice\_coalition(i, (i, k_1, k_2, \dots, k_n), g_m^i)$

描述:  $i$  通知  $k_1, k_2, \dots, k_n$  联盟形成.

$notice\_propose(i, (i, k_1, k_2, \dots, k_n), exchange(g_m^i, g_m^*))$

描述:  $i$  通知  $k_1, k_2, \dots, k_n$  合作对象  $co$  关于交换目标的提议.

$notice\_success(i, (i, k_1, k_2, \dots, k_n), g_m^i, success \text{ or } fail)$

描述:  $i$  通知  $k_1, k_2, \dots, k_n$  合作成功或失败.

### 3 多 Agent 合作结构实现算法

符号约定:

$A_m^i$  表示 Agent  $i$  的知识库中关于实现目标  $g_m^i$  的可选合作对象集合, 该集合中的元素是全序的, 其顺序表示合作的优先性;

$M_i$  表示 Agent  $i$  的知识库中关于信息 Agent 或中介 Agent 的查询对象集合, 该集合中的元素是全序的, 其顺序表示查询对象选取的优先性;

$B_m^i$  表示 Agent  $i$  的知识库中关于实现目标  $g_m^i$  的可选合作对象集合, 该集合中的元素是无序的, 该集合用来存放从信息 Agent 所获取的合作对象信息;

$co$  表示选取的合作对象;  $qo$  表示选取的查询对象;  $g^{co}$  表示合作对象  $co$  提出的交换目标.

$return(bj, num1, num2)$ :  $bj$  表示合作对象的来源;  $bj = 1$  表示合作对象来源于  $A_m^i$ ,  $bj = 2$  表示合作对象来源于  $B_m^i$ .  $return(bj, num1, num2)$  表示  $bj = 1$  则返回到算法行  $num1$ ,  $bj = 2$  则返回到算法行  $num2$ .

update.  $A(\text{co}, A_m^i)$  表示根据合作对象的排序规则把合作成功的对象  $\text{co}$  在  $A_m^i$  中的顺序进行更新. 合作对象排序规则可以考虑与  $\text{co}$  合作成功的次数, 与  $\text{co}$  合作失败的次数, 与  $\text{co}$  合作的平均收益等方面的内容.

update.  $M(\text{qo}, M_i)$  表示根据查询对象的排序规则把查询成功的对象  $\text{qo}$  在  $M_i$  中的顺序进行更新. 查询对象排序规则可以考虑向  $\text{qo}$  查询成功的次数,  $\text{qo}$  返回的信息质量(信息质量可以考虑返回的信息中的合作对象与  $i$  合作成功的次数, 以及平均收益等)

**算法1** 一对一算法( $\text{co}$  为单个的 Agent,  $i$  单独与  $\text{co}$  合作)

- (1) 若  $i$  自己能实现目标  $g_m^i$ , 则算法结束;
- (2) 按照优先顺序从  $A_m^i$  中选取一个尚未选取过的合作对象  $\text{co}$ , 若选取成功, 则把  $\text{co}$  作选取标记, 令  $b_j = 1$ , 执行(5), 否则执行(3);
- (3) 从  $B_m^i$  中任选一个合作对象  $\text{co}$ , 若选取成功, 则令  $b_j = 2, B_m^i = B_m^i - \{\text{co}\}$ , 执行(5), 否则执行(4);
- (4) 按照优先顺序从  $M_i$  中选取一个尚未选取过的查询对象  $\text{qo}$ , 若选取成功, 则把  $\text{qo}$  作选取标记, 执行(12), 否则  $g_m^i$  不能实现, 算法结束;
- (5) request( $i, \text{co}, g_m^i, [\text{rguments}]$ );
- (6) 等待消息;
- (7) 若超过系统等待时间或收到消息 reject( $\text{co}, i, g_m^i$  or exchange( $g_m^i, g^{\text{co}}$ )), 则 update.  $A(\text{co}, A_m^i)$ , update.  $M(\text{qo}, M_i)$ , return( $b_j, (2), (3)$ );
- (8) 若收到消息 with. notice( $\text{co}, i, g_m^i$  or exchange( $g_m^i, g^{\text{co}}$ )), 则返回(6)
- (9) 若收到消息 accept( $\text{co}, i, g_m^i$  or exchange( $g_m^i, g^{\text{co}}$ )), 则 accept. affirm( $i, \text{co}, g_m^i$ , or exchange( $g_m^i, g^{\text{co}}$ )), update.  $A(\text{co}, A_m^i)$ , update.  $M(\text{qo}, M_i)$ , 算法结束;
- (10) 若收到消息 propose( $i, \text{co}, \text{exchange}(g_m^i, g^{\text{co}})$ ), 则判断是否能够进行目标交换: 若可以进行目标交换, 则 accept. affirm( $i, \text{co}, \text{exchange}(g_m^i, g^{\text{co}})$ ), update.  $A(\text{co}, A_m^i)$ , update.  $M(\text{qo}, M_i)$ , 算法结束, 否则执行(11).
- (11) 若  $i$  有新的提议, 则 propose( $\text{co}, i, \text{exchange}(g_m^i, g^{\text{co}})$ ), 返回(6), 否则, cancel. request( $i, \text{co}, g_m^i$ ), update.  $A(\text{co}, A_m^i)$ , return( $b_j, (2), (3)$ );

(12) sk( $i, \text{qo}, ?, g_m^i$ );

(13) 等待;

(14) 若超过系统等待时间, 则返回(4);

(15) 收到 reply( $\text{qo}, i, r_m^i, g_m^i$ ): 若  $r_m^i =$ , 则返回(4), 否则 receive. affirm( $i, \text{qo}, \text{sk}(g_m^i)$ ),  $B_m^i = B_m^i - r_m^i$  返回(3)

**算法2** 一对多算法( $\text{co}$  为 Agent 联盟,  $i$  单独与  $\text{co}$  合作)

假设  $\text{co} = (k_1, k_2, \dots, k_m)$ ,  $m \geq 2$ . 以算法1为基础, 当  $\text{co}$  中有负责的 Agent, 例如  $k_1$ , 则  $i$  只向  $k_1$  请求合作, 不考虑  $k_1$  如何同联盟中的 Agent 进行通讯、交易; 当  $\text{co}$  中没有负责的 Agent, 则  $k_1$  需要向  $\text{co}$  中所有的 Agent 发出合作请求, 并与所有的 Agent 达成协议, 否则, 合作就没有形成.

**算法3** 多对一算法( $\text{co}$  为单个的 Agent,  $i$  同其他 Agent 结成联盟, 然后与  $\text{co}$  合作) 联盟规则: 谁发起联盟, 谁负责联系、负责目标交换.

以算法1为基础, 假设  $i$  通过联盟( $i, k_1, k_2, \dots, k_n$ ) 与  $\text{co}$  合作, 则

collition. request( $i, (i, k_1, k_2, \dots, k_n), g_m^i$ )  
如果  $\forall j \in \{1, 2, \dots, n\}$ , 都存在 accept. collition( $k_j, i(i, k_1, k_2, \dots, k_n), g_m^i$ ) 则联盟形成. 然后由  $i$  代表其结成的联盟向  $\text{co}$  发出合作请求; 当不存在目标交换时, 若合作成功, 则 notice. success( $i(i, k_1, k_2, \dots, k_n), g_m^i, \text{success or fail}$ ); 若有提议, 则  $i$  代表  $k_1, k_2, \dots, k_n$  来考虑 accept, propose 或者 reject.

**算法4** 多对多算法( $\text{co}$  为 Agent 联盟,  $i$  先同其他 Agent 联盟, 然后与  $\text{co}$  合作)

联合算法1、算法2、算法3, 可实现算法4.

算法1~4中都有可能非完全合作的情况: 例如  $\text{co}$  自身不能完成目标  $g_m^i$ , 但  $\text{co}$  却同意与  $i$  合作, 然后再寻找合作对象, 去完成  $g_m^i$ , 这种情况类似于期货交易.

## 5 应用实例

无论是在科研还是在商业领域, 多 Agent 合作都有着广泛的应用前景. 本文给出多 Agent 合作在电子商务领域的一个应用实例: 基于多 Agent

合作的商品交易系统. 如图 2 所示,  $p_i$  与  $b_u$  的商品交易属于一对一合作,  $p_i$  与  $(b_{t_1}, b_{t_2}, \dots, b_{t_m})$  的商品交易属于一对多合作,  $(p_{k_1}, p_{k_2}, \dots, p_{k_n})$  与  $b_u$  的商品交易属于多对一合作,  $(p_{k_1}, p_{k_2}, \dots, p_{k_n})$  与  $(b_{t_1}, b_{t_2}, \dots, b_{t_m})$  的商品交易属于多对多合作. 多个买方 Agent 联合起来与卖方 Agent 讨价还价或者通过授权给中间 Agent  $m_j$  (middle agent) 与卖方 Agent 讨价还价进行商品交易, 由于批量购买从而降低了购买价格, 有益于购买方; 多个卖方 Agent 联合起来解决缺货问题(没有这种货物或这种货物数量不够), 降低了库存成本, 并提高了卖方的信誉, 有益于卖方, 同时, 缩短了买方的购物时间, 也有益于买方. 通过中间 Agent 进行商品交易在一定程度上可以保护买卖双方的隐私.

下面以购买图书为例, 给出一个简单的“一对一合作”的交易场景, 以通讯原语来表示交易过程:

sk( $p_j, m_j, ?, \text{buy-book}(\text{“信息论”}, \text{“机械工业出版社”}, \text{“1999”})$ )

reply( $m_j, p_i, (r_1, r_2, \dots, r_n), \text{buy-book}(\text{“信息论”}, \text{“机械工业出版社”}, \text{“1999”})$ )

request( $p_i, r_2, \text{buy-book}(\text{“信息论”}, \text{“机械工业出版社”}, \text{“1999”})$ )

accept( $p_j, r_2, \text{exchange}(\text{buy-book}(\text{“信息论”}, \text{“机械工业出版社”}, \text{“1999”}))$ )

pay(book\_price, 29)

accept( $r_2, p_i, \text{exchange}(\text{buy-book}(\text{“信息论”}, \text{“机械工业出版社”}, \text{“1999”}))$ )

pay(book\_price, 23.2)

accept(ffirm( $p_i, r_2, \text{exchange}(\text{buy-book}(\text{“信息论”}, \text{“机械工业出版社”}, \text{“1999”}))$ )

pay(book\_price, 29)

下面解释上述交易过程:  $p_i$  想购买“机械工业出版社”于 1999 年出版的《信息论》, 但是不知道从何处购买, 于是  $p_i$  向中间 Agent  $md_j$  查询,  $md_j$  告诉  $p_i$  从  $r_1, r_2, \dots, r_n$  中的任何一个 Agent 都可以买到;  $p_i$  随机选择一个供应商  $r_2$ , 并向  $r_2$  提出购买请求,  $r_2$  返回该书的价格, 但是  $p_i$  请求打折,  $r_2$  接受请求, 交易确定.

### 5 结束语

本文对多 Agent 合作结构进行了系统的研究, 提出了 16 种多 Agent 合作结构, 这些合作结构反映了可能存在的 16 种合作场景, 但是, 这些合作结构是否能覆盖所有的合作现象还需要证明. 本文提出的关于实现多 Agent 合作的通讯原语及其算法便于实现, 但对于实际问题, 还需要进一步完善与扩充. 惩罚与激励机制是多 Agent 合作中的一种反馈机制, 本文在 Agent 的目标中初步考虑了惩罚与激励机制, 下一步工作将研究如何通过设计有效的惩罚与激励机制去促进多 Agent 的合作.

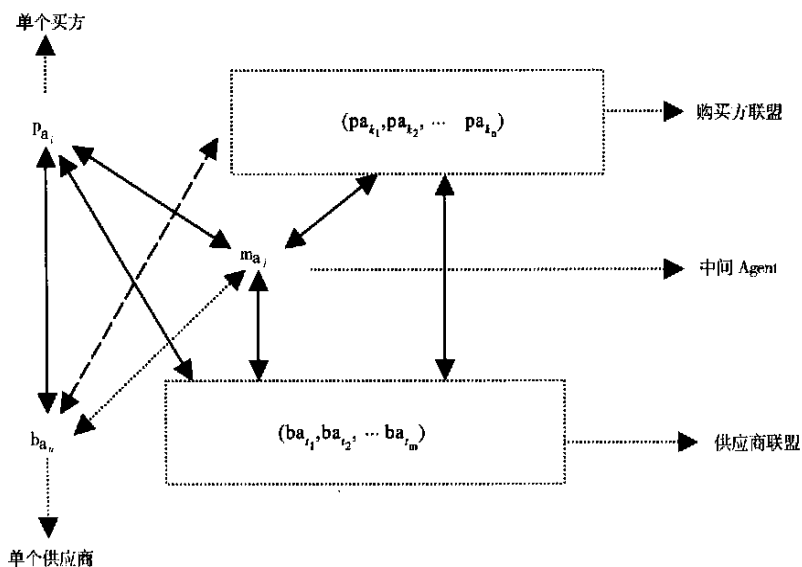


图 2 基于多 Agent 合作的商品交易示意图

## 参考文献:

- [1]d 'Inverno M, Luck M. Making and breaking engagements: An operational analysis of agent relationships in multi agent systems: Methodologies and applications[R]. Lecture Notes in Artificial Intelligence, Zhang and Lukose (eds.), pages 48—62, Berlin: Springer-Verlag, 1997
- [2]d 'Inverno M, Luck M, Wooldridge M. Cooperation Structures[C]. 15th International Joint Conference on Artificial Intelligence, AUG 23 - 29, 1997. IJCAI-97-Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 1997. 600—605
- [3]Wooldridge M, Jennings N R. Formalizing the Cooperative Problem Solving Process[C]. Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence (IWDAI- 93), Lake Quinalt, WA, July 1994
- [4]Castelfranchi C, Miceli M, Conte R. Limits and Levels of Cooperation: Disentangling Various Types of Prosocial Interaction, Decentralized AI- 2, Yves Demazeau and Jean-Pierre Muller (Eds.), Elsevier Science Publisher B. V., Holland, 1991
- [5]Shehory O, Kraus S, Yadgar O. Emergent cooperative goal-satisfaction in large-scale automated-agent systems[J]. Artificial Intelligence, 1999, 110(1): 1—95
- [6]Axelrod R. The Evolution of Cooperation[M]. USA: BasicBooks, 1984
- [7]Parunak H V D. Characterizing Multi-agent Negotiation[R]. Working Group Summary. In international Workshop on Multi-Agent Systems, IWMAS-98, 1998
- [8]Beer M, d 'Inverno M, Luck M, *et al.* Negotiation in multi-agent systems[J]. Workshop of the UK Special Interest Group on Multi-Agent Systems (UKMAS 98), DEC, 1998. KNOWL ENG REV 1999, 14 (3): 285—290
- [9]罗 翊,石纯一. Agent 协作求解中形成联盟的行为策略[J]. 计算机学报, 1997, 20(11): 961—965
- [10]Shehory O, Kraus S. Methods for task allocation via agent coalition formation[J]. Artificial Intelligence Journal, 1998, 101(1 - 2): 165—200
- [11]Sycara K, Zeng D. Cooperative Intelligent Software Agents. Robotics Institute Technical Report # CMU-RI-TR-95-14. <http://www.cs.cmu.edu/~softagents/publications.html>
- [12]Ito T, Fukuta N, Shintani T, *et al.* BiddingBot: A Multiagent Support System for Cooperative Bidding in Multiple Auctions 4th International Conference on MultiAgent Systems[C]. 2000 Fourth International Conference on Multi-agent Systems, 2000. 399—400
- [13]Nelson Minar, Kwindla Hultman Kramer, Pattie Maes. Cooperating Mobile Agents for Mapping networks[EB/OD]. <http://www.media.mit.edu/~nelson/research/routes/>

## On cooperation structure of multi-agent

LI Chang-hong, LI Min-qiang, KOU Ji-song

Institute of Systems Engineering, Tianjin University, Tianjin 300072, China

**Abstract:** This paper begins with an introduction of the current status of research on the multi-agent cooperation, then sixteen cooperation structures are provided according to the three dimensions, the completely of cooperation objective exchanging of both sides exists, and the number of both sides. Furthermore, a set of communication primitives is designed for multi-agent cooperation. Finally, algorithms of realizing the cooperation structures that present in this paper and a typical example on multi-agent cooperation was described.

**Key words:** multi-agent system; multi-agent cooperation; cooperation structure; communication primitive; algorithm