

# Agent 组织结构设计的一种形式语义\*

张伟<sup>1,2</sup>, 石纯一<sup>1</sup>

<sup>1</sup>(清华大学 计算机科学与技术系,北京 100084);

<sup>2</sup>(中国科学院计算技术研究所 智能信息处理开放实验室,北京 100080)

E-mail: zwytu@263.net

http://www.tsinghua.edu.cn

**摘要:** Agent 组织是多 Agent 系统研究的重要方面,在不同的 Agent 组织形成方法中常采用显式的组织形成方法.基于 $\pi$ 演算和化学抽象机给出了组织结构设计过程的一种形式语义.把参与组织结构设计的有关 Agent 都作为化学抽象机中的化学分子,而化学抽象机的运行过程即是组织结构的设计过程.这种语义给出方式准确、直观,使组织结构设计系统易于实现,改进了 Ferber 和 Xu 关于 Agent 组织的操作语义和组织结构设计的工作.

**关键词:** MAS(multi-agent systems);Agent 组织结构;形式语义; $\pi$ 演算;抽象机

中图法分类号: TP18 文献标识码: A

在 MAS(multi-agent systems)中,组织是一种有效的求解方式.这是由于一方面 Agent 能力的有限性假设,另一方面由于现实问题的复杂性和分布性,使得采用组织的方法进行问题求解更为有效<sup>[1]</sup>.

关于组织的研究引起了越来越多的注意和重视,一个重要的研究内容是组织的形成方法和角色的定义.在诸多组织形成方法中,面向结构的组织形成方法与人类社会的产生机制相似,是一种显式的目标明确的组织形成方法.文献[2]采用面向结构的方法给出了组织结构描述以及组织结构的设计原则和设计过程,但对组织结构的形成过程缺少严格而明确的形式化描述和形式语义.Ferber 等人基于一个特定的组织模型给出了组织形成的操作语义<sup>[3]</sup>,描述了一个 Group 的创建和 Agent 加入组织的过程,由于没有采用面向结构的组织形成方法,而且局限于一个具体的组织模型,因此使该方法的一般性受到影响,另外,对于 Agent 组织的形式语义描述也不够完整.本文基于面向结构的组织形成方法,使用 $\pi$ 演算和化学抽象机,给出了组织结构设计过程的形式语义描述,从而改进了 Ferber 等人 and Xu 的工作.这一方面严格定义了完整的组织结构形成过程,避免引起歧义,另一方面也方便对组织结构形成过程的理解和系统的实现.

本文第 1 节简单给出语义描述使用的工具及抽象机.第 2 节概述了组织结构的描述和设计过程.第 3 节对组织结构的形成过程给出了语义描述.第 4 节通过化学抽象机的运行举例,说明了组织结构的设计过程.最后给出了简单的结论.

## 1 $\pi$ 演算和化学抽象机

$\pi$ 演算<sup>[4]</sup>是一种进程代数.从技术上讲, $\pi$ 演算是 CCS 的扩展,使其可以传递通道名.在 $\pi$ 演算中,每个表达式表示一个进程,代表一个可以自由并行运行的实体,进程可以有子进程.进程之间的通信是通过在命名的通道上交换信息实现的.设  $P$  和  $Q$  代表进程的元变量, $x, y, z, v$  是名字元变量.

$\pi$ 演算的句法为

\* 收稿日期: 2000-10-23; 修改日期: 2001-02-20

基金项目: 国家自然科学基金资助项目(69973023;60173011)

作者简介: 张伟(1961 - ),男,山东滕州人,博士生,副教授,主要研究领域为分布式人工智能;石纯一(1935 - ),男,河北秦皇岛人,教授,博士生导师,主要研究领域为人工智能应用基础.

$$P, Q = \mathbf{0} | (P|Q) | P; Q | x(z).P | x(y).P | C?P | \nu(v)P | A(y_1, \dots, y_n).$$

其中: $\mathbf{0}$  是一个空进程,该进程不做任何事情,通常可以忽略; $P|Q$  是两个进程的并发组合,表示  $P$  和  $Q$  并发执行; $P;Q$  表示或者  $P$  执行,或者  $Q$  执行; $x(z).P$  是一个输入前缀,其中  $x$  是输入通道, $z$  是参数,可以携带发往通道  $x$  的消息; $x(y).P$  是输出前缀,其中  $x$  是输出端口名, $y$  是输出参数,完成输出以后,进程  $x(y).P$  约减为  $P$ ; $C?P$  是一个测试进程,如果条件  $C$  成立,则进程约减为  $P$ ,否则约减为  $\mathbf{0}$ ; $\nu(v)P$  定义一个新的名字  $v$ ,该名字在  $P$  中不存在; $A(y_1, \dots, y_n)$  是一个定义的等式.

等价表达式可以互相代换而不影响计算,用一致关系符“ $\approx$ ”表示.对于并发运算,“ $|$ ”操作符相应的子表达式与先后顺序无关,因此有等价关系:

$$\begin{aligned} P|Q &\approx Q|P, \\ (P|Q)|R &\approx P|(Q|R). \end{aligned}$$

$\pi$ 演算的操作语义通过与 $\lambda$ 演算相似的一组约减规则给出.主要规则描述了进程之间如何通过通道间发送消息进行通信:

$$x(y).P | x(z).Q \approx P[y/z]Q.$$

其他规则描述了进程的并发性,即当进程是并发关系时,约减关系不变:

$$\begin{aligned} P|R &\approx Q|R \text{ if } P \approx Q, \\ \nu(v)P &\approx \nu(v)Q \text{ if } P \approx Q, \\ P &\approx Q \text{ if } P \approx P' \text{ and } Q \approx Q' \text{ and } P' \approx Q'. \end{aligned}$$

本文对组织结构设计的描述使用了 Picol 语言,是在基本 $\pi$ 演算的基础上增加了表达式、算术运算符和关系运算符等语言成分实现的,文献[3]给出了 Picol 语言的句法和语义.

化学抽象机<sup>[5]</sup>简称 CHAM(chemical abstract machine),可以用来给出 $\pi$ 演算的操作语义.化学抽象机是由 Banatre 和 Le Metayer 的  $\Gamma$  语言发展而来,是借用了化学比喻的一种抽象机,为描述并行计算和并发语言的操作语义提供了形式化框架.

化学抽象机以化学溶液表示, $S = \{m_1, \dots, m_k\}$ ,  $m_i (i=1, \dots, k)$  是一个化学分子.通过定义溶液中的分子和化学反映规则可以实现一台化学抽象机,分子根据有关的抽象语法规则定义.化学抽象机的运行遵循反应规则和化学定律.反应规则为

$$\{m_1, \dots, m_k\} \rightarrow \{m'_1, \dots, m'_k\} \text{ if condition}(m_1, \dots, m_k).$$

表示如果溶液中出现左面的分子并且满足一定条件,那么溶液中的分子被右面的分子替代.化学抽象机的化学定律,形式化地描述了溶液中的自由化学反应:

$$\begin{aligned} S &\rightarrow S' \\ S \oplus S'' &\rightarrow S' \oplus S'' \end{aligned}$$

其中 $\oplus$ 表示两种溶液的合并.

## 2 组织的形成

组织可以认为是 Agent 的集合和 MAS 的分布式问题求解机制,也是多 Agent 系统社会性的一个重要的方面.研究工作主要集中在组织的形成和演化等问题上.

Agent 组织是 MAS 研究的 3 个主要问题(模型、组织、交互语言)之一,组织的形成和演化问题是基于 Agent 的计算和 Agent 合作问题求解的关键<sup>[6]</sup>.目前,关于 Agent 组织的形成方法有:(1) 基于协商的合同网协议,当一个 Agent 发现自己面临的问题难以独立求解或者合作求解更有效时,通过与其他 Agent 的协商,将其部分或全部任务委托其他 Agent 来完成,在他们之间形成顾客与服务员关系型组织;(2) 基于依赖关系的社会推理,是通过 Agent 之间依赖关系的社会推理,发现与其目标有依赖关系的其他 Agent,与他们形成不同形式的合作组织;(3) 基于对策论的联盟形成,通过联盟效用的计算与分配,依据 Agent 个体、联盟和群体理性原则,在 Agent 之间形成不同的联盟;(4) 基于价格调控的市场机制,通过市场价格的调整,Agent 形成供求平衡下的买卖关系;(5) 面向结构的组织形成.

多 Agent 在问题求解中表现的组织形态就是组织结构.现在,对组织形态越来越多地是采用显式的表示方式<sup>[7]</sup>.面向结构的 Agent 组织形成方法与人类社会的组织产生机制相符合,是一种目标明确的组织形成方法.

一个为了完成一组目标的组织结构为

$$OS ::= \langle OS\_ID, ROLE, RELATION, O\_GOAL, MANAGER \rangle.$$

其中  $OS\_ID$  是组织结构标识符,  $ROLE$  是非空的角色集合,  $RELATION$  是角色关系集合,  $O\_GOAL$  是非空的组织目标集合,  $MANAGER$  是非空的组织管理者集合.

组织结构的设计过程一般由 5 个步骤完成:(1) 确定组织结构的目标;(2) 任务分解树的合并;(3) 组织结构初始化;(4) 角色的合并;(5) 计算角色的收益.本文在给出组织结构设计的形式语义时,对文献[2]中的组织结构设计过程做了如下修正,这是为了便于组织结构设计的描述和实现.

- (1) 设计组织结构框架,作为组织结构设计的基础.
- (2) 由不同 Agent 的共同目标出发,确定组织目标.
- (3) 根据组织目标进行任务分解,划分出不同的子任务.
- (4) 由子任务设计组织角色.

### 3 组织结构设计的形式语义

面向结构的组织形成的主要概念有角色、角色目标、组织目标等,可使用 Picol 语言对组织结构的设计过程进行形式语义描述.

为简化描述,设 Agent  $a$  和 Agent  $b$  有部分共同求解目标,欲通过组织管理者 Manager 进行组织结构设计(本文假定组织结构设计统一由 Manager 完成),多个 Agent 的情况可以很方便地通过扩展实现.

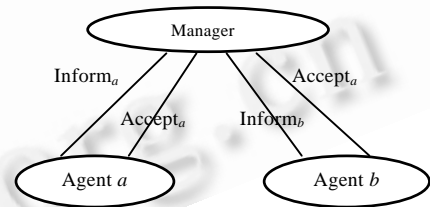
Agent  $a$  有目标集  $G_a$ , Agent  $b$  有目标集  $G_b$

$$G_a = \{goal_{a1}, goal_{a2}, \dots, goal_{c1}, goal_{c2}, \dots\},$$

$$G_b = \{goal_{b1}, goal_{b2}, \dots, goal_{c1}, goal_{c2}, \dots\}.$$

其中  $goal_{c1}, goal_{c2}, \dots$  是 Agent  $a$  和 Agent  $b$  的共同目标.

在进行组织结构设计时, Agent  $a$  和 Agent  $b$  分别向 Manager 发送组织结构设计请求 request, Manager 根据请求首先设计一个空的组织结构框架并通知 Agent  $a$  和 Agent  $b$ , Agent  $a$  和 Agent  $b$  收到组织结构框架设计成功的消息后,向组织设计者发送各自的求解目标,供 Manager 进行任务分解,进一步完成组织目标和组织角色的设计. Manager, Agent  $a$  和 Agent  $b$  之间的命名消息通道如图 1 所示. 组织结构的设计过程是 manager, agent<sub>a</sub> 和 agent<sub>b</sub> 的并行运行过程.



管理者, 通知, 接受.

Fig. 1  
图 1

在  $\pi$  演算中引入了一个新的操作原语 new, 用来表示创建一个空的组织结构或者组织角色, 分别表示为 new os 和 new roles, new 由参数带回一个标识符. 创建后的角色存放在角色字典中.

Milner 给出了基本  $\pi$  演算的形式语义<sup>[4]</sup>, Ferber 用化学抽象机给出了 Picol 语言的操作语义<sup>[3]</sup>.

组织结构设计的语义可描述如下:

```

SYSTEM =def
  v(accepta, acceptb, informa, informb)manager|agenta|agentb
manager(msg) =def
  manager(msg)
  match(msg)
  create_org_str
  v(o_goal, o_task)create_str.create_org_goal(o_goal).
  task_decomp(o_goal, o_task).create_roles(o_task)
    
```

```

create_org ...
create_str=def
  accepta(r1)|acceptb(r2).
  (r1 and r2=REQUEST)?
  (new os(os_id). informa(os_id,OK)| informb(os_id,OK)
  : informa(fail)| informb(fail))
create_org_goal(o_goal)=def
  (accepta(Ga)|acceptb(Gb)).(Ga Gb ∅)?
  o_gaol(Ga Gb):o_goal(fail)

```

其中 $\emptyset$ 表示空集合。

Agent 组织角色的创建基于组织目标.通过把组织目标分解成各个子任务,并把相同或相似的子任务合并得到需求数量,由角色承担不同的子任务完成组织角色的创建.组织目标的分解比较困难,需要相关领域的专门知识,已经超出了本文的讨论范围,这里只是假设任务的分解由进程  $\text{task\_decomp}(o\_goal,o\_task)$  实现,其中  $o\_goal$  和  $o\_task$  分别表示组织目标集合和任务分解后组织的子任务集合。

每个子任务由一个角色完成,由角色功能实现.子任务的大小会影响组织形成时 Agent 对角色的承担,这需要在组织形成的过程中对子任务和角色进行调整.这是一个必然的过程,可以使角色功能和 Agent 的能力相匹配。

已创建的角色统一放入角色字典中.在创建过程中如果遇到相同或相似的组织角色则进行合并,并记录组织角色的需求数量,供具体组织形成时参考使用.对角色字典的操作由相应的操作原语完成,这里不再给出语义描述。

```

create_roles(o_task)=def
  (o_task=∅)? informa(OK)|informb(OK);
  (o_task ∅)? (new role)|create_roles(o_task-r)

```

Agent  $a$  的 $\pi$ 演算描述为

```

Agenta=def
  accepta(REQUEST).informa(r1,r2).((r2=OK)?
  accepta(Ga):stop).
  informa(r).(r=OK)? manager(create_org(r1))

```

其中  $\text{create\_org}$  表示组织结构创建成功以后,Agent  $a$  启动该进程建立具体的 Agent 组织.如果 Agent  $a$  和 Agent  $b$  没有共同目标,则组织结构设计失败,系统终止.除了命名通道和有关变量不同以外,Agent  $b$  与 Agent  $a$  的演算完全相同,这里不再重复。

## 4 举例

足球队的组织结构设计。

设 Agent  $a$  和 Agent  $b$  要踢足球,如果已经存在一个足球队,则 Agent  $a$  和 Agent  $b$  可以根据自己的能力,通过承担相应的角色加入足球队.如果没有足球队,则需要建立一个足球队,这首先要进行足球队的组织结构设计.现假设没有足球队存在,Agent  $a$  和 Agent  $b$  经过协商,发现有成立足球队的共同目标,并由 Agent  $a$  负责设计足球队的组织结构.Agent  $a$  和 Agent  $b$  分别有目标  $G_a=\{\dots,goal_c,\dots\}$ ,  $G_b=\{\dots,goal_c,\dots\}$ ,  $goal_c$  表示成立足球队的共同目标。

Agent  $a$  和 Agent  $b$  进行足球队组织结构设计的过程如下

```

agenta=def
  acceptb(Gb).
  (Ga Gb=∅)? informb(∅,fail);

```

$$(G_a \ G_b = \emptyset)? (\text{new os}(team\_id, G_a \ G_b). \text{inform}_b(os\_id, OK). \\ \text{task\_decomp}(G_a \ G_b, o\_task). \text{create\_roles}(o\_task)$$

$$\text{agent}_b =_{\text{def}}$$

$$\text{accept}_b(G_b). \text{inform}_b(r1, r2). (r=OK)? \text{create\_org}(r1): \text{stop}$$

化学抽象机的运行给出了足球队组织设计的约减过程:

$$\{ | \text{agent}_a, \text{agent}_b | \}$$

$$\{ | \text{accept}_b(G_b). (G_a \ G_b = \emptyset)? \text{inform}_b(\emptyset, \text{fail}); \\ (G_a \ G_b = \emptyset)? (\text{new os}(team\_id, G_a \ G_b). \text{inform}_b(team\_id, OK). \\ \text{task\_decomp}(o\_goal, o\_task). \text{create\_roles}(o\_task), \\ \text{accept}_b(G_b). \text{inform}_b(r1, r2). (r2=OK)? \text{create\_org}(r1): \text{stop}) | \}$$

$$\{ | (\text{goal}_c = \emptyset)? \text{inform}_b(\emptyset, \text{fail}); \\ (\text{goal}_c \ \emptyset)? (\text{new os}(team\_id, G_a \ G_b). \text{inform}_b(team\_id, OK). \\ \text{task\_decomp}(\text{goal}_c, o\_task). \text{create\_roles}(o\_task), \\ \text{inform}_b(r1, r2). (r2=OK)? \text{create\_org}(r1): \text{stop}) | \}$$

$$\{ | (\text{new os}(team\_id, \text{goal}_c). \text{inform}_b(team\_id, OK). \\ \text{task\_decomp}(\text{goal}_c, o\_task). \text{create\_roles}(o\_task), \\ \text{inform}_b(r1, r2). (r2=OK)? \text{create\_org}(r1): \text{stop}) | \}$$

$$\{ | \text{inform}_b(team\_id, OK). \text{task\_decomp}(\text{goal}_c, o\_task). \text{create\_roles}(o\_task), \\ \text{inform}_b(r1, r2). (r2=OK)? \text{create\_org}(r1): \text{stop}) | \}$$

$$\{ | \text{task\_decomp}(\text{goal}_c, o\_task). \text{create\_roles}(o\_task), \\ (OK=OK)? \text{create\_org}(team\_id): \text{stop}) | \}$$

$$\{ | \text{task\_decomp}(\text{goal}_c, o\_task). \text{create\_roles}(o\_task), \\ (OK=OK)? \text{create\_org}(team\_id): \text{stop}) | \}$$

$$\{ | \text{task\_decomp}(\text{goal}_c, o\_task). \text{create\_roles}(o\_task), (\text{create\_org}(team\_id)) | \}$$

$$\{ | (\text{create\_org}(team\_id)) | \}$$

Agent  $a$  和 Agent  $b$  联合进行组织结构设计的过程,通过化学抽象机的运行直观而准确地表示出来.这里,任务分解和角色创建都是 Agent  $a$  的内部过程,不再进一步给出.基于目标和任务分解知识,  $goal_c$  可以分解成不同的子任务,如进攻、防守、中场接应等,  $\text{create\_roles}(o\_task)$  根据这些子任务递归地进行角色的创建,创建后的角色由  $\text{create\_roles}$  存入角色字典中.

## 5 结 语

基于不同的方法和理论,产生了不同的组织结构和组织的形成方法.本文对于显式的、面向结构的组织形成方法,用  $\pi$  演算定义了组织结构的形成过程,给出了严格的形式语义,尤其是引入了化学抽象机,可以把组织结构的设计过程用化学抽象机的运行直观地表示出来,改进了 Ferber<sup>[3]</sup>和 Xu<sup>[2]</sup>的关于 Agent 组织结构设计和组织形成的操作语义方面的工作.本文采用的方法还可以应用于描述其他 MAS 的形成和求解过程.

组织的形成和演化问题是 MAS 研究的主要内容之一,对于面向结构的组织的形成和求解,需要进一步研究组织的形成和演化机制等问题,并用形式语义学的方法给出严格的可操作的语义定义.另外,本文提出的方法还需要进一步完善,使之更适用于基于 Agent 组织的问题求解.

## References:

- [1] Horling, B., Benyo, B., Lesser, V. Using self-diagnosis to adapt organizational structures. Technical Report, 1999-64, Department of Computer Science, UMASS, 1999.

- [2] Xu, Jin-hui, Zhang, Wei, Shi, Chun-yi. A structure-oriented method for agent organization formation. In: Shi, Zhong-zhi, Faltings, B., Musen, M., eds. Proceedings of Conference on Intelligent Information Processing. Beijing: Publishing House of Electronics Industry, 2000. 251~258.
- [3] Ferber, J., Gutknecht, O. Operational semantics of multi-Agent organization, In: Muller, J.P., ed. Proceedings of the 6th International Workshop on Agent Theories, Architectures and Languages. Heidelberg: Springer-Verlag, 1999.
- [4] Milner, R., Parrow, J., Walker, D. A calculus of mobile processes (I&II). Information and Computation, 1992,100(1):1~77.
- [5] Berry, G., Boudol, G. The chemical abstract machine. Theoretical Computer Science, 1992,1996(1):217~248.
- [6] Jennings, N.R. On agent-based software engineering. Artificial Intelligence, 2000,117(2):1429~1436.
- [7] Xu, Jin-hui, The research on the Agent model and coalition mechanism [Ph.D. Thesis]. Beijing: Tsinghua University, 2000 (in Chinese).

#### 附中文参考文献:

- [7] 徐晋晖.多 Agent 模型与联盟机制研究[博士学位论文].北京:清华大学,2000.

## A Formal Semantics of Agent Organization Structure Design\*

ZHANG Wei<sup>1,2</sup>, SHI Chun-yi<sup>1</sup>

<sup>1</sup>(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China);

<sup>2</sup>(Laboratory of Intelligent Information Processing, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080, China)

E-mail: zwytu@263.net

http://www.tsinghua.edu.cn

**Abstract:** Agent organization is an important research topic of MAS (multi-agent systems). The explicit expressions of Agent organization structure are more widely used in the implementation of Agent organizations. A formal semantics of processes of Agent organization structure design is given in this paper based on  $\pi$ -calculus and the chemical abstract machine (CHAM). Components associated with the organization structure design are taken as molecules of CHAM and use the executions of the CHAM to express procedures of Agent organization structure formation. This formal semantic method is both accurate and direct. It is convenient to implement Agent organization structure design and improves the work of Ferber and Xu about implementation and semantics of Agent organization and organization structure.

**Key words:** MAS; Agent organization structure; formal semantics;  $\pi$ -calculus; abstract machine

\* Received October 23, 2000; accepted February 20, 2001

Supported by the National Natural Science Foundation of China under Grant Nos.69973023, 60173011