

# 支持智能体的通信原语库系统的研究与实现\*

周立 吴泉源 陈火旺

(国防科技大学计算机系,长沙 410073)

**摘要** GKD-APS 是我们基于面向智能体程序设计范例(AOP)设计并实现的一个支持智能体相关概念及智能体间消息通信的 Prolog 解释器.它同时引入了智能体、精神状态、知识、信念、承诺、能力及智能体间的消息通信等概念,并提供了对这些概念的直接支持.GKD-APS 为实现多智能体间的知识通信和合作运行,构造多智能体系统提供了基础,有效地支持了分布式知识系统的研究和开发.

**关键词** 智能体,通信,精神状态,继承,对象.

近年来,人工智能研究开始强调推理与知识表示有机结合的能动实体——智能体(agent).人工智能领域的智能体通常是指在一定环境下持续自主运行的实体.多年来,人工智能的研究主要是针对单个智能体在静态环境中的行为,人工智能系统的设计通常无需直接涉及智能体的概念.相应地,人工智能程序设计语言也没有直接支持智能体构造的程序设计概念和语言设施.但是随着知识处理、机器人和分布式人工智能等研究领域的深入发展,人们不可避免地更多涉及具有智能体个性特征的理性推理活动的研究.这类推理活动一般涉及智能体的信念、意图、偏爱、承诺、能力等精神状态(mental state),表现出“主观的、动态的”和“开放的”特点.

美国 Stanford 大学的 Yoav Shoham 教授在 1990 年明确提出了面向智能体的程序设计(Agent Oriented Programming, 简记为 AOP)的概念. AOP 是一种以计算的社会观为基础的新型程序设计范例,它以智能体的设计与构造为人工智能程序设计的核心. AOP 的出现反映了人工智能领域日益重视智能体的动态特征和智能体间交互活动的研究开发需要.研究开发 AOP 语言是适应人工智能研究上述新变化的十分重要的基础性工作,它将给人工智能的发展带来较大的影响.

为便于构造多智能体系统,更好地支持多智能体之间的知识通信和合作运行,我们根据当前国际研究发展现状和我们已有的工作基础,设计并实现了一个支持智能体的通信原语库系统——GKD-APS,它在我们已有的面向对象程序设计系统 GKD-OOPS<sup>[1]</sup>的基础

\* 本文 1994-04-04 收到,1994-07-21 定稿

作者周立,1968年生,博士生,主要研究领域为智能体、逻辑程序设计、专家数据库系统.吴泉源,1942年生,教授,博士生导师,主要研究领域为软件工程与人工智能.陈火旺,1937年生,教授,博士生导师,主要研究领域为软件工程与人工智能.

本文通讯联系人:周立,长沙 410073,国防科技大学计算机系

上,把对象演变为智能体,将对象的内部状态固定为知识、信念、承诺和能力等精神状态,智能体之间通过继承关系构成层次相关的整体,并根据 Speech act 理论丰富了对象系统中的简单消息通信,提供了由 3 种具有不同语用性质的智能体通信原语所组成的支持智能体通信的原语库。

## 1 语言的设计

### 1.1 基本概念

#### (1) 智能体(agent)

在本系统中,我们采用 Shoham<sup>[2]</sup>关于智能体的一个“高层次”的定义。

如果一个实体的状态可被视为包含了诸如知识、信念、承诺和能力等精神状态(mental state)时,该实体就是智能体。这个定义也可表述为,如果一个实体可用诸如知识、信念、承诺和能力等精神状态来描述时,那么该实体就是智能体。

在本系统定义的智能体中,其精神状态包括 4 个属性:知识、信念、承诺和能力。

#### (2) 知识(knowledge)

知识就是一个智能体所知道的、必为真的、不会再被修改的断言。换言之,如果一个智能体把某条断言作为知识放入其知识库,那么这条断言必须永远为真,该智能体也就不会再认为它的知识可能是错的,而需对其进行修改。

#### (3) 信念(belief)

信念是一个智能体自认为正确的断言。这条断言可能为真,也可能为假,如果有新的证据产生,就需要对智能体原有的信念进行修正。即智能体的信念是非单调的。

#### (4) 承诺(commitment)

承诺在本系统中具体表现为承诺规则。承诺规则描述了智能体在作出某项承诺时所必须满足的条件。

一旦智能体 a 对智能体 b 作出了承诺,那么智能体 a 只有在以下条件之一满足时才会取消其已作出的承诺:智能体 a 已实现它所作出的承诺;智能体 a 发现它已无能力再完成已作出的承诺;智能体 b 不再要求智能体 a 去完成已对它作出的承诺,并且智能体 a 已知这一情况。

#### (5) 能力(capability)

为了能判断其行为可能产生的结果,智能体必须具有关于它能够做什么的有关信息,这些相关信息就由智能体的能力描述给出。

### 1.2 继承与多重继承

在 GKD-APS 中,我们为—组智能体构造一个“组智能体”,该智能体的知识、信念、承诺和能力分别是组内各智能体的“公共知识”、“公共信念”、“公共承诺”和“公共能力”,即“组智能体”的精神状态属性可被组内所有智能体继承。

GKD-APS 中实现的继承是多重继承,一个智能体可以同时属于多个不同的“智能体组”,其所属智能体组的“组智能体”构成一个“组智能体”链表。在求解过程中,当通过一条继承链求解失败时可采用深度优先的规则,通过第二级继承智能体回溯完成其它继承链的搜索。一般来说,多重继承主要用来让智能体可以继承不同组智能体的精神状态。

### 1.3 对象与智能体

GKD-APS 是在我们的面向对象程序设计系统 GKD-OOPS 的基础上实现的, 它们二者既有差别又有密切联系. 表 1 总结了 GKD-APS 与面向对象程序设计系统 GKD-OOPS 的关系.

表 1 对象与智能体的对比

	面向对象程序设计系统	支持智能体的通信原语库系统
基本组成单元	对象	智能体
用以定义基本单元状态参数	无限制	知识、信念、承诺和能力等
计算过程	消息传送和方法响应	消息传送和方法响应
消息类型	无限制	通知、询问和请求等

GKD-OOPS 是由对象组成, 各个对象之间可以进行消息通信, 每个对象的内部都有自己的方法来处理接收到的消息. 支持智能体的通信原语库系统 GKD-APS 对该框架进行了特例化, 它固定了对象(现称智能体)的状态(现称精神状态), 使之包含知识、信念(关于其自身、世界和其它智能体)、承诺和能力等精神属性, 它们都有各自精确定义的语法. 该系统内的计算包括智能体间的通知(inform)、询问(inquire)和请求(request), 它大大丰富了面向对象程序设计系统中的通信机制, 这个思想来源于言语行为(speech act)理论.

## 2 通信原语

### 2.1 消息发送

向一个智能体发送的消息类型有以下 3 种:

inform(agent, fact)

inquire(agent, goal)

request(agent, action)

其中 agent 是接收消息的智能体, fact、goal 和 action 是向 agent 发送的消息, inform、inquire 和 request 是系统提供的智能体通信原语.

inform 的语义为在 agent 的信念库中加入一条新的信念 fact, 并进行相应的信念维护.

inquire 的语义为请求 agent 在其知识信念环境中对 goal 进行证明. 若证明成功, 则把一致化后的 goal 送入发出该“询问”消息的智能体的信念库, 并进行相应的信念维护. 智能体的知识信念环境就是该智能体所具有的及其可继承到的所有知识和信念.

request 的语义为要求 agent 在其能力环境中证明 action. 在开始证明之前, 必须先检查在 agent 中与 action 相应的承诺规则的条件是否满足, 而承诺规则中条件的满足与否, 跟 agent 当前所具有的知识信念密切相关. 智能体的能力环境包括该智能体本身的及其可继承到的所有能力描述.

消息可作为一条命令发送给某个智能体, 也可作为任何程序子句的子目标, 在程序运行过程中实现多个智能体间的知识共享和合作运行.

### 2.2 通信原语的语用性质

随着面向智能体的程序设计(Agent-Oriented Programming)的提出,智能体的行为特征和交互活动成为 AOP 语言关心的重要问题.在这种情况下,AOP 语言不仅涉及语法、语义性质,还涉及语用(Pragmatic)性质.美国哲学家 C. Morris 把语用学定义为研究“符号与符号解释者的关系”,如果一项研究明确地涉及语言使用者,我们就把它归入语用学的领域.语用学已经应用于人机对话和人工智能.J. McCaathy 在他提出的 Elephant 2000 程序设计语言中采用了许多言语行为的概念.Shoham 在 AOP 语言 AGENT0 中把智能体的通信行为规定为通知(inform)和请求(request)两类言语行为<sup>[2]</sup>.E. Werner 在研究智能体的合作框架中也采用语用学的概念描述智能体间的相互通信的含义<sup>[3]</sup>.根据言语行为的理论,我们在 GKD-APS 中提出了 3 类通信原语:通知(inform),询问(inquire)和请求(request).

言语行为是关于智能体之间通信行为的语用学概念.言语行为理论基于以下假设:人类交际的基本单位不是句子或其它表达手段,而是完成一定的行为,例如:“通知”、“请求”、“询问”、“命令”、“感谢”、“道歉”等.言语行为的特点是说话人通过说一句或若干句话来执行一个或若干个上面列举的行为,这些行为的实现还可能给听话人带来某些后果.英国哲学家 J. L. Austin 提出的言语行为三分说<sup>[4]</sup>,即“以言指事(locutionary act)”,“以言行事(illocutionary act)”和“以言成事(perlocutionary act)”对分析智能体之间的通信行为很有价值.考虑以下的“问询处”具体场景,就不难理解言语行为三分说的基本思想.

旅客 passenger 询问服务员 conductor 从长沙到北京的火车有几趟. passenger 提问本身涉及“以言指事”:关于长沙至北京的列车车次.接到 passenger 的询问后,引发 conductor 的下列行为:①查看自己的知识库中是否记录有从长沙至北京的列车车次;②如果有,则告诉 passenger 所有的相关车次;如果没有,则回答 passenger“对不起,不知道”.行为①、②构成询问的“以言行事”,行为①、②产生的效果,即做出相应的回答,构成询问的“以言成事”.

J. McCaathy 在 Elephant 2000 的设计中提出:程序的输入输出规范和完成规范分别对应用程序的“以言行事”和“以言成事”.我们认为,采用言语行为三分说有助于全面认识智能体之间的交互行为,特别有助于理解 GKD-APS 中的通信原语:

- 通信原语中蕴含的信息构成消息的“以言指事”.
- 通信原语引发的消息传递和原语方法的执行构成消息的“以言行事”.
- 通信原语执行后产生的效果构成消息的“以言成事”.

我们认为,特定的言语行为取决于其在“以言指事”、“以言行事”和“以言成事”3 方面的具体属性.例如,通信原语“通知(inform)”就是从以下 3 个方面进行定义:

“以言指事”方面:通知原语中蕴含了陈述性断言.

“以言行事”方面:被通知者把通知中的陈述性断言记录在自己的知识库.

“以言成事”方面:被通知者得到了通知中的陈述性断言.

### 3 主要算法描述

GKD-APS 的主要算法包括语法分析算法、归结算法及 3 个通信原语的实现算法.

#### (1) 语法分析算法

首先识别输入的文件是否是智能体描述文件.若是定义智能体的文件,则对智能体说明进行分析,建立一个智能体及其关系的结构,然后对文件中定义的智能体的知识、信念、承诺

规则及能力描述进行语法分析,建立它们的内部结构.若不是智能体描述文件,则将其定义的内容加入 user 模块中. user 模块是在系统启动时由系统自动创建的.该算法完成了智能体的识别,智能体的静态输入及智能体存储器的动态改变.

### (2) 归结算法

它将求解的目标与其动态求解环境中的子句按深度优先搜索及最左归结法则进行一致化匹配,求得目标中的所有变量值,该过程包括两级回溯.按照面向智能体程序设计中的继承概念,每个目标求解过程中的各子目标都必须从接收消息的智能体开始求解;若求解失败,则使用其“组智能体”的知识库、信念库、承诺规则库、或能力库中的相应信息.

动态求解环境是指在程序运行过程中动态确定的求解目标的模块环境,求解过程中所能运用的知识就是这一模块中的所有子句,而与其它未使用的模块无关.但在程序执行过程中可以通过继承关系共享其它智能体中的信息.

### (3) “通知”原语实现算法

“通知”原语 inform 用于对智能体信念库的动态改变.当一个智能体  $A_1$  向另一智能体  $A_2$  发送一条“通知”消息,  $A_2$  接到该条消息后,先检查自己的信念库是否与所通知的内容冲突,若无冲突,则把所通知的内容作为新信念插入信念库中,否则不采取任何动作.

### (4) “询问”原语实现算法

“询问”原语 request 用于一个智能体通过对另一个智能体的咨询,以达到扩充自身信念的目的.即当智能体  $A_1$  向智能体  $A_2$  发出“询问”消息后,  $A_2$  就在自己的知识信念环境中证明  $A_1$  所询问的内容,若证明成功,则把一致化后的询问内容插回到  $A_1$  的信念库中,成为  $A_1$  的一条新信念,并对  $A_1$  的信念库进行维护;若证明失败,则不采取任何动作.

### (5) “请求”原语实现算法

当一个智能体需要另一智能体协助完成某一动作时,即可借助“请求”原语来实现.智能体  $A_1$  向智能体  $A_2$  发出“请求”消息,在  $A_2$  的承诺规则得到满足并作出相应承诺的前提下,  $A_2$  才把  $A_1$  所请求的内容送到自己的能力环境中证明.若  $A_2$  的承诺规则得不到满足,或者其能力环境当前无法实现已作出的承诺,则通知用户“已拒绝作出承诺”或“承诺无法兑现”.

## 4 结 论

我们设计并实现的 GKD-APS 从总体上看是由智能体组成,智能体之间可以通过通信原语进行通信,从而实现非相关智能体之间的相互联系,相关智能体之间可通过继承关系共享“组智能体”的精神状态属性.每个智能体的精神状态都是由 Prolog 子句来描述,这样既保持了逻辑语言的描述性特征,表达能力和清晰的语义,又吸取了面向智能体范例的知识表示方法和丰富自然的通信机制,不仅易于表达人类专家的知识,模拟人类专家的精神状态,而且便于实现多智能体间的知识通信和合作运行. GKD-APS 增强了知识的表达能力与推理能力,支持了分布式人工智能的研究.

## 参考文献

- 1 Zhou Li, Wu Quanyuan. GKD-OOPS: a new object-oriented logic programming system. The 3rd International Conference on System Integration, Sao Paulo, Brazil, IEEE Computer Society Press, 1994. 75-86.

- 2 Shoham Y. Agent oriented programming. *Artificial Intelligence*, 1993, **60**:51—92.
- 3 Werner E. Cooperation agents, a unified theory of communication and social structure. In: Gasser L, Huhns M, eds, *Distributed Artificial Intelligence* Pitman, London, 1989, **2**:3—25.
- 4 何自然. 语用学概论. 长沙:湖南教育出版社, 1988.

## THE RESEARCH AND IMPLEMENTATION OF AGENT COMMUNICATIVE PRIMITIVE BASE SYSTEM

Zhou Li Wu Quanyuan Chen Huowang

*(Department of Computer Science, National University of Defense Technology, Changsha 410073)*

**Abstract** This designs and implements an agent-oriented prolog interpreter——GKD—APS, which is in accordance with agent-oriented paradigm and supports communicating between agents and the concepts such as agent, mental states, belief, commitment and ability. GKD—APS provides the basis for building multi-agent systems and cooperation and knowledge communication among multiple agents. It effectively supports the research and development of distributed knowledge systems.

**Key words** Agent, communication, mental state, inheritance, object.