

多 Agent 智能制造系统中 通信机制的研究

The Research of Communication in Multi-Agent Manufacturing System

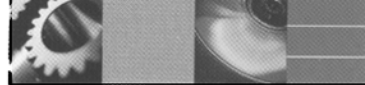


杜玉强 王明哲

(武汉华中科技大学控制科学与
工程系 430074)

摘要: 在分析 agent 和多 agent 系统特点的基础上,着重研究了基于多 agent 的智能制造系统中的通信问题,提出了对话模式作为 agent 之间的通信机制,并使用着色 Petri 网进行了校验。

关键词: agent 多 agent 系统
对话模式 着色 Petri 网



1 引言

随着信息技术的飞速发展和经济全球化趋势的不断加剧,制造业将不得不面对日益激烈的市场竞争。产品生命周期的日益缩短,更新速度的不断加快和顾客对产品需求的日趋个性化、多样化,使得制造企业的生产模式和制造组织方式逐渐由面向产品转变为面向客户、面向需求和面向服务。为了赢得竞争,企业迫切需求一种具有良好柔性、快速反应性和具有容错能力的分布式网络合作化生产制造系统。这种制造系统应当能够在具备一定成本效益优势的情况下快速适应市场变化,根据不同的产品要求,迅速、经济的建立制造过程,并能动态的对整个制造过程进行自适应、自组织、自学习、自由化和自维护。分布式人工智能领域的研究成果表明,采用agent技术建立的智能制造系统是最具潜力的发展方向。本文首先对agent和多agent系统的特点进行了介绍,然后提出了一种基于对话模式的方法来设计agent间的通信,并采用着色Petri网对其进行了校验。

2 agent和多agent系统

关于agent,至今还没有一个统一的定义,认为在没有人的干预下能够自主地完成给定任务的对象都可以称为agent。而现在比较普遍的观点是agent,应该具有三个重要的特征—自治性、适应性和协作性。所谓自治性是在没有外界(人或软件)的干预下,能够积极主动地完成相应的任务;适应性是指agent具有感知、适应外界环境的能力,并有自学习能力,通过学习不断提高自身适应变化的外界环境;协作性是多agent系统的一个重要特征,在多agent系统中,agent相互协作共同完成一个任务。因此多agent系统的能力并不由单个agent具有的能力所决定,而是由agent相互协作集体体现出来的智能所决定。agent具有目标、知识、标识等基本属性,它由通信模块、业务处理模块、推理模块、学习模块、消息传递等功能单元所组成(如图1所示),从中可以看出,agent具有面向对象技术中的类的特征,如继承、封装、消息传递等,所以在实现的时候,可以用面向对象的技术来实现agent的功能。

agent的具体组成结构如下:

- (1) 标识: 是一个agent区别于多agent系统中其他agent的一种属性特征,一般包括agent的名称、地址等;
- (2) 目标: agent不断运行所追求的具体目标,由目标确定agent的责任和义务;
- (3) 知识: 包括事实和规则,存储于agent的知识库中;
- (4) 通讯模块: 负责agent的通信,消息接受和发送,可以传递任务、运行结果和实现知识共享;
- (5) 推理模块: 根据agent的目标、知识和能力以及最新信息进行推理和决策,作用于消息处理、业务处理等,做出的决策必须有利于目标的实现
- (6) 业务处理模块: 进行业务处理,由业务处理方法构成,它是agent

实现目标的主体;

(7) 学习模块: 从agent的运行过程中总结经验,为知识库增加新的知识,提高适应环境变化的能力;

多agent系统由一组在逻辑上或者物理位置上分步的许多agent组成,它们通过网络连接,共享资源,为完成共同的任务而组成一个有组织的群体。一般认为多agent系统特别适用于那些能根据空间、时间或功能进行分解和划分的应用问题,在这些应用中采用多agent系统将带来如下优点:由于处理的并行化,系统的运行速度将加快;由于对信息的处理是在信息源附近进行的,所以对通信带宽的要求较低;由于某一个agent出错不会影响整个系统的运行,因此,系统具有较高的可靠性;由于感知、处理和动作紧密相伴,系统具有较高的反应速度。

3 agent之间的通信机制—对话模式

在多agent系统中,agent之间的协调和协作是解决问题的关键,因此,agent之间必须要通过通信来交换信息,这些信息包括目标、意图、结果和状态等。对话(conversation)是这种信息交流中的最有效最直接的一种方式。对话是指两个或两个以上的agent为了达到特定目的而传递的一系列消息。通信的必备条件是在异类的agent之间能够进行可靠和灵活的交互,并且

(1) 不存在可能的矛盾和死锁;(2) 对话会伴随着agent间活动的结束而中止。为了简化agent间通信模块的设计,需要有一套通信规则:对话策略(conversation policies, CPs)来降低agent通信的复杂程度,下面将采用基于模式的方法来设计面向agent的生产系统中的通信和对话模块。

3.1 需求分析

所要设计的对话模块主要有以下要求:

- (1) 它应该是面向任务的,这是因为对话通常发生在agent需要实现特定目标或是完成特定任务时,这与一般的通信协议有很大不同。
- (2) 对话模块应当和对话的语义联系起来。

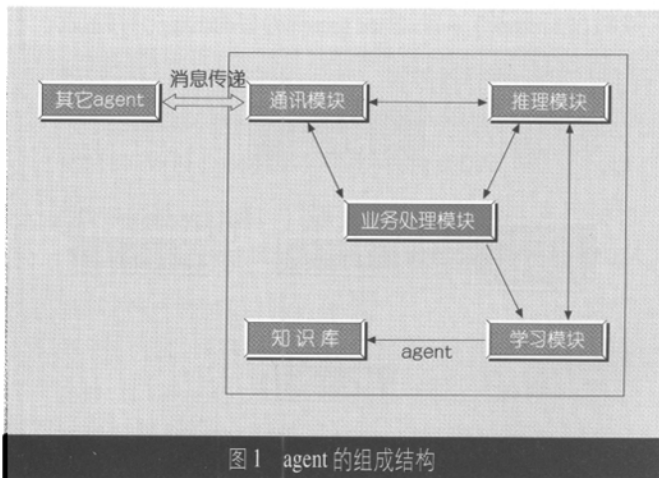


图1 agent的组成结构

- (3) 通信模块必须允许通信内容的提取。
- (4) 通信模块能提供错误修复。
- (5) 它应该具备可复用性和可扩展性。
- (6) 它能够按照需求自动装载。

对话模式的构建要从对所有对话主题确认开始,然后把单独的通信行为收集起来加以分类,形成模式。这些模式的集合可以用CPN来检查是否存在矛盾和死锁。通过分类、实例化、继承和聚合从而得到一组模式类和模式实例。图2给出了对话模式的构建过程:

3.2 对话主题的确定

对话通常集中于一个或多个主题。主题包括一系列的属性。因此,一旦知道了agent的目标或任务,我们就能定义它与其他agent的对话主题,进而定义它们的对话行为。对话的主题可以用一些变量来描述,这些变量在agent间进行对话的时候会被赋予特定的值。对话主题(Topic, TP)可以如下定义: TP=(TP_ID; ARGUMENTS)。TP_ID指的是主题的标识, ARGUMENTS是指主题中的参数列表。举例如下:

在一个按订单生产的企业中,在定单中会对到货日期、价格和交货地点进行约定。这样,后勤agent、生产计划agent和生产agent需要就不同的主题进行对话,相关的主题表述如下:

TP1: (Prod_Tar; Cust_order, Cust, Loc, DD, Price).....(1)

TP2: (Mfg_Cost; Factory_order, DD, Cost).....(2)

(1) 式中对话主题Prod_Tar描述了在后勤agent和生产计划agent之间对生产目标进行的对话:运输货物A以Price的价位在DD日送到Loc地点的客户Cust_order手中。

(2) 式中对话主题Mfg_cost描述了生产计划agent和生产agent之间对生产成本进行的对话:对于订单Factory_order在DD日完成的生产成本Cost。这些主题通过一系列的对话来确定切实可行的生产目标和最优的生产完成日期。

由上可知,对话主题只是在较抽象的层次上定义了信息的交互,并没有讨论例如通信时机等具体问题。

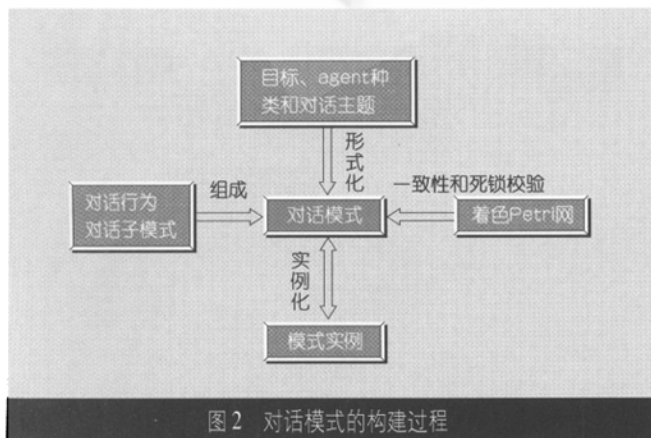


图2 对话模式的构建过程

3.3 对话模式的定义

对话模式是面向目标的对话方法和模块化的对话策略。

定义: Conv_schema = (Goal, Task, Topics, Agent types, Sub_Schemata, Acts, States, Flows)其中 Goal: 参与对话的agent要实现的目标; Task: 参与对话的agent要执行的任务;

Types={Agt1, Agt2, ..., Agtm} (m>0) 参与对话和提供所需服务的agent的类型;

Sub-Schemata = {Sub_Schema1, Sub_Schema2, ..., Sub_Schemas} (s>=0) 组成对话模式的子模式;

Acts={Act1, Act2, ..., Actn} (n>0) 所有参与对话的agent的行为的集合;

States = {Sta1, Sta2, ..., Stao} (o>0) 每个agent的对应状态的变化;

Flows = Acts × States ∪ States × Acts: Acts和States之间的关系;

TPs = {Tp1, Tp2, ..., Tpk} (k>0) 对话主题的集合;

对话模式分为原子模式和复合模式两类。复合模式可以是几种模式的组合,而原子模式则不可以。举一个原子模式的例子:agent A对agent B有一个业务请求:REQUEST,可以表示为对话模式:REQUEST=(Goal, Task, Topics, (A,B), (request, continue, repair, req_ack, req_repair, acknowledge, cancel), (submitted, failed, succeeded), (content)).

表示的意思为两类agent A和B, A向B请求的对话过程分为七个子模式:发出请求(request),继续等待(continue),请求修理(repair),接受请求(req_ack),接受修理请求(req_repair),确认请求(acknowledge),对话结束(cancel)。对话的状态有三种:提交(submitted),失败(failed),成功(succeed)。

3.4 对话模式的CPN建模

Petri网是描述具有分布、并发、异步特征的离散事件动态系统的有效工具。它用四种元素为系统模型建模:库所(Place)、变迁(Transition)、弧(Arc)和令牌(Token),它用库所、变迁、弧的连接标识系统的静态结构,通过变迁的激发和令牌的移动描述系统的动态行为。着色Petri网(Colored Petri Nets, CPN)作为一种高级的Petri网模型,通过对库所和令牌加以颜色类型的标定,对变迁和弧加上发生条件的约束,使得建模功能大大增强。正是因为着色petri网的以上优点,所以采取它作为对话模式的建模分析工具。

3.4.1 原子模式的构建

对话模式中的行为或子模式可以用CPN中的变迁表示,模式的状态用库所表示,消息用令牌来持有。下面是模式的构建步骤:

第一步:根据对话主题区分agent的种类、属性和状态;

第二步:对于每一个agent的种类,模式中的每一种行为用变迁来表示,同一种agent对应的多个行为的变迁水平排列;

第三步:在变迁之间添加库所和弧。通过这种方式,为每个agent之间发生的信息交换建立了模型;

第四步:为对话主题添加被状态库所描绘的信息交换;

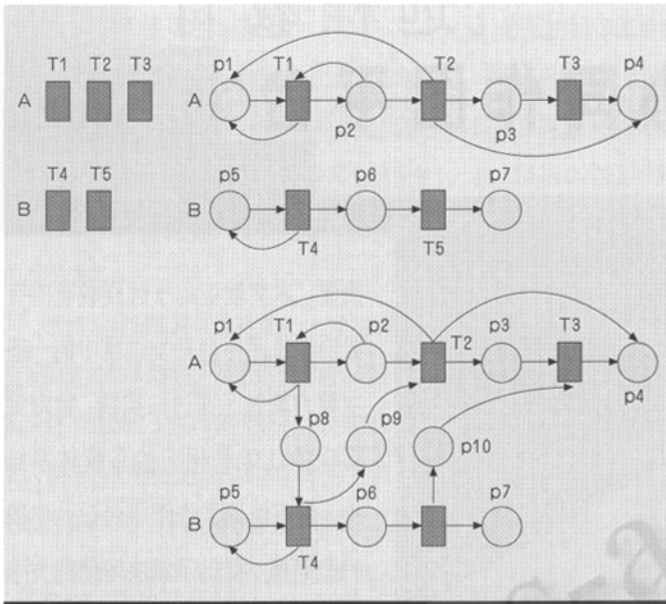
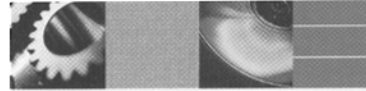


图3 对话模式 REQUEST 的构建

第五步：建立外部接口：

下面对话模式 REQUEST 进行构建：

T1 = {Request, Req_ack, Continue, Repair}

T2 = {Continue, Repair, Cancel}

T3 = {Cancel, Ack}

T4 = {Req_Repair, Repair, Ack}

T5 = {Reply}

3.4.2 复合模式的构建

复合模式有很多种构建方法，如分层的CPN表示法和面向对象的分析和设计方法都可以。复合模式可以通过对一些原子模式的整合而得到。整合的方法主要有两种：聚合和连接。聚合这种方法能够充分利用着色Petri网的特性进行仿真。例如把REQUEST、REPLY和ACK三种原子模式聚合为复合模式：info-obtaining, CPN的图形分析如图4所示：

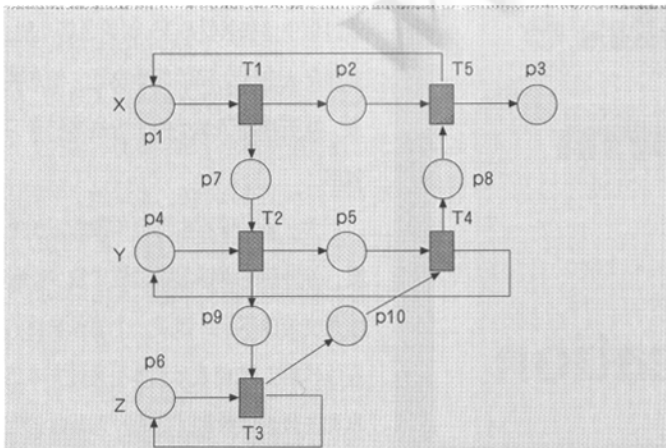


图4 对话模式 info-obtaining 的构建

变迁中的颜色集定义如下：

T1 = T2 = REQUEST("intent", "data_info.")

T3 = T4 = REPLY("intent", "result_info.")

T5 = ACK(result_info)

库所中的颜色集定义如下：

C(p1) = C(p2) = C(p7) = { <Cus_Order, LogisticsAgent>, <Fac_Order, Production PlanningAgent>, <Mfg_Order, Factory Agent> }

C(p5) = C(p9) = { <LAC1, Cus_Order>, <LAC2, Fac_Order>, <Mfg_Order, Factory Agent> }

>}

C(p10) = { <Production Planning Agent>, <Product Model>, <Feature Model> }

C(p3) = { <Logistics Agent, Production PlanningAgent>, <Production Planning Agent, ProductModel>, <Production Planning Agent, FeatureModel> }

C(p8) = { <Production PlanningAgent>, <Product Model>, <Feature Model> }

C(p4) = { <LAC1>, <LAC2>, <LAC3> }

C(p6) = { <KA1>, <KA2>, <KA3> }

4 结束语

在多agent智能制造系统中，问题的解决是一个渐进的多步的过程。其中每一步都存在agent之间的对话。本文着重分析了系统中agent之间通信的特点，提出了一种基于对话模式的通信机制，并用着色Petri进行了验证，得到了较满意的结论。

参考文献

- 1 Chung, K. T. and C. H. Wu (1997). Dynamical Scheduling with intelligent agent. Metra Application Note 105, Metra, Palo Alto, CA.
- 2 Lin, F.R., G.w.Tan, and M.J.Shaw. 1999. Multiagent enterprise modeling. Journal of Organizational Computing and Electronic Commerce, 9(1):7-32.
- 3 Bayard R J. Info S Leuth: agent — based semantic integration of information: in open and dynamic environments. Proceeding of the ACM SIGMOD Conference on Management of Data. 1997: 195-205 .
- 4 Brussel, H. V., J. Wyns, P. Valckenaers, L.Bongaerts, & P. Peeters (1998). Referencearchitecture for holonic manufacturing systems: PROSA. Computers in Industry, Vol.37, pp.255-274.
- 5 刘大有、杨鲲、陈建中. Agent 研究现状和发展趋势. 软件学报. 2000; 11 (3): 315-321. 6 王强、曾祥云. 要想ERP成功. 先作BPR. 《环球管理》. 2001年第12期. 2001.