

# 基于 Agent 的供应链模型<sup>\*</sup>

赵新宇 林作铨

(北京大学信息科学系 北京 100871)

**摘要** 供应链(SC)是一个由供应商、制造商、仓库、物流、渠道、零售商和客户所组成的网络,通过原料获取、运输、制造并最终递送到客户的活动组织到一起。软件 Agent 是可以为我们完成特定任务的具有自治行为的对象。本文试图从 Agent 模型角度对供应链进行建模,提出一种可动态加载能力的多 Agent 模型,设计了一个基于 Agent 的供应链管理系统原型,并通过实现一个供应链上定单处理流程来说明该模型所具有的良好性质。

**关键词** Agent, BDI, 供应链, 能力

## An Agent-Based Supply Chain Modeling

ZHAO Xin-Yu LIN Zuo-Quan

(Department of Information Science, Peking University, Beijing 100871)

**Abstract** A supply chain is a network of supplier, manufacturer, warehouse, logistics, channel, retailer, customer collectively responsible for procurement, manufacturing and distribution activities associated with one or more families of related products. Software agent is an object that has autonomous action and be able to accomplish specific task for us. We provide a multi-agent model in which the capability can be dynamically loaded. A prototype of agent-based supply chain management system is presented. In addition, we implement an order process of supply chain to show nice properties of this model.

**Keywords** Agent, BDI, Supply chain, Capability

## 1 引言

供应链(SC)是一个由供应商、制造商、仓库、物流、渠道、零售商和客户所组成的网络,通过原料获取、运输、制造并最终递送到客户的活动组织到一起<sup>[1-3]</sup>。协同商务<sup>[4,5]</sup>使供应链进一步扩展,使企业从单纯的以产品为中心的经营策略逐步转向网络环境下的商务主体(企业)和商务活动的全过程协调和集成,涉及整个供应链及其与相关环节之间的协同运作,给企业的商务模型及运作带来变革,实现了企业与合作伙伴间的双赢。协同商务平台及其相关技术<sup>[6,7]</sup>将企业的所有应用和数据集成到一个信息管理平台之上,并以统一的界面提供

给企业员工、客户、合作伙伴,用户能够在统一、集成的框架下访问企业业务流程、客户关系等其所需要的个性化信息。假设供应链上的企业通过协同商务平台及其相关信息系统(如 EDI 或电子商务等)联接,供应链中的一个业务流程是先从客户的订货请求开始,客户通过零售商了解企业产品目录(或产品配置表),客户确定所需产品后签订订单提交到零售商,零售商根据订单产品与企业联系;如果企业库存不足以满足需求,则准备生产订单列表中的产品,同时还需要和一级、二级供应商联系以获得零部件;供应商与原料提供商联系购买生产原料;整个业务流程需要物流企业提供物流渠道。一个典型的供应链网络<sup>[8]</sup>如图 1 所示。

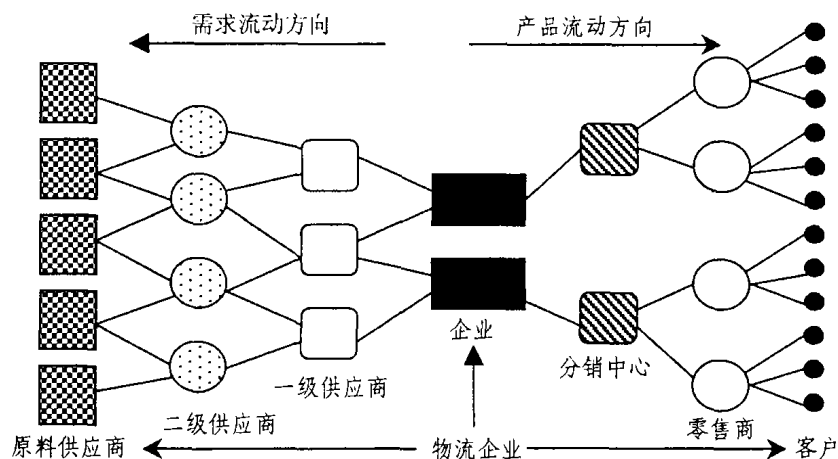


图 1 供应链网络

随着经济全球化,新技术、新产品和新需求的不断出现,购买者和供应商的关系越来越依靠服务质量、递送效率、协商

的可变性以及合作的信誉等因素,而不是传统的建立在成本基础上的关系<sup>[9]</sup>,企业面临的是一个更加复杂的生态环境;供

<sup>\*</sup> 本文受国家自然科学基金项目(编号:69925203)资助。赵新宇 博士研究生,主要研究方向:基于 Agent 的供应链管理软件,人工智能。林作铨 教授,博士生导师,主要研究方向:计算机软件,人工智能。

应链业务流程需要多个企业及其多个信息系统间的互操作, 构建于协同商务平台上的供应链系统的设计、开发和维护需要新的技术支持以适应这种环境的动态变化, 本文用到的 agent 技术就是其中一种解决方案。

软件 Agent 是可以为我们完成特定任务的具有行为能力的对象<sup>[10]</sup>, 基于 Agent 的方法最近被广泛应用于企业建模、应用集成、制造规划、生产调度和过程控制中<sup>[2]</sup>。多 Agent 系统是由多个 Agent 组成的系统, Agent 之间及 Agent 与环境间通过通信、协商和协作来共同完成任务。多 Agent 系统应用于供应链管理是从 Fox 等<sup>[11]</sup>开始的, 供应链管理及协同商务概念的本质是在多决策者的环境中取得一致性, 因此采用多 Agent 系统建模是一个有效的技术途径。已经有一些工作探讨把多 Agent 模型应用于供应链管理系统中<sup>[8,12,13]</sup>。

本文试图从 Agent 模型角度对供应链进行建模。BDI 模型是 Rao 和 Georgeff<sup>[16,17]</sup>、Cohen 和 Levesque<sup>[15]</sup>以及 Bratman<sup>[14]</sup>等提出的一种重要的 Agent 认知模型, 得到广泛的应用<sup>[2,17]</sup>。供应链系统是一个复杂的、开放的和不可预测的环境, 多个系统之间需要直接或间接地进行交互, 纯粹的基于 BDI 的 Agent 模型难于满足实时性要求高、需求变化迅速的供应链系统的需求, 需要定义新的模型来支持这样的应用。我们提出一种具有可动态加载能力的多 Agent 模型, 并设计了一个基于此模型的企业供应链管理平台(以下简称 ASCM)。ASCM 中的 Agent 采用扩展 BDI 模型, 具有信念(B)、愿望(D)、主张(I)以及可以动态加载的能力组件(C)。其特征是如果 Agent 的功能在系统设计和实现时还没有确定, 只需要提供接口和标准元数据的定义, 在出现新需求的时候, 就可以依据接口的定义, 实现能力组件以满足需求而不需改动原有的系统; Agent 初始化时加载能力组件成为系统中不同类型的 Agent (类型的定义将在下文中详细介绍); Agent 在运行时可以根据其精神状态(B、D、I)从能力组件库中选择加载能力组件, 进而获得执行智能动作的能力。这种模型在 Agent 系统的分析、设计、开发和运行阶段表现出来的特征适用于对实时性、动态性要求较高的供应链系统, 对于停止运行很短时间就会造成很大损失的那些系统尤为重要。这使得基于 Agent 的供应链管理系统的开发和维护代价更低, 具有柔性、可扩展, 同时增加了系统的可重用性。

本文第 2 节给出可动态加载能力的 Agent 模型以及性质; 第 3 节讨论了 ASCM 的基于 Agent 分层次的体系结构,

重点讨论应用组件层 Agent 类型的概念; 在第 4 节, 通过一个订单实例说明该模型的应用; 在第 5 节, 与相关的基于 Agent 的供应链系统进行比较, 分析了 ASCM 系统的优势; 最后, 我们简要地进行总结和给出进一步的工作。

## 2 Agent 模型

### 2.1 基类 Agent 的定义

在 ASCM 系统中, 采用类似于面向对象的思想, 把系统中 Agent 的共同特征提取出来, 定义基类 Agent 的结构。文 [18] 只给出了反应式 Agent 和慎思型 Agent 模型的一种 BDI 结构, 与之相比, ASCM 系统加入了决定 Agent 的类型、支持系统柔性 and 动态性的能力组件(将在下节详细阐述)和系统实际运行必要的一些组件, 设计了一个通用的基于 BDI 模型的基类 Agent 结构。ASCM 系统中 Agent 的定义如下:

$$\text{Agent} = (\text{B}, \text{D}, \text{I}, \text{Tr}, \text{Re}, \text{Ev}, \text{Pr}, \text{Co}, \text{Ca})$$

其中, B: (En, St, Aq) 是 Agent 的信念模型, 由 Agent 关于所处环境的信息 En、Agent 自身的信息 St 和 Agent 相关的熟人信息 Aq 组成;

D: Agent 的愿望模型, 包括 Agent 的根据当前的信念和偏好而产生的多个愿望;

I: Agent 的意图队列, 根据当前信念选择队列中的规划进行执行。在系统中意图队列中的意图被定义为一个规划六元组(Na 规划名, PreCon 前提条件, PosCon 完成状态, Typ 类型, Pri 优先级, Exe 执行体);

Tr: 触发器, 根据 Agent 的信念和其所关心的环境的改变, 形成所要执行事件的愿望, 加入 D 中;

Re: 感知机, 通过标准格式的消息的方法接收、感知外界环境输入, 形成所要执行事件的愿望, 加入 D 中;

Ev: 评估器, 评估感知机和触发器送来的愿望执行所带来的效用, 确定该愿望的类型和优先级并对愿望执行进行规划, 形成的规划放入意图队列;

Pr: 处理机, 可以同时执行多个意图队列中的规划;

Co: 通讯组件, 负责发送和接受 Agent 之间进行协调、协商交互的消息;

Ca: 能力组件, 决定 Agent 的类型, Agent 通过能力组件执行规划。

基类 Agent 结构的 UML 类图如图 2 所示。

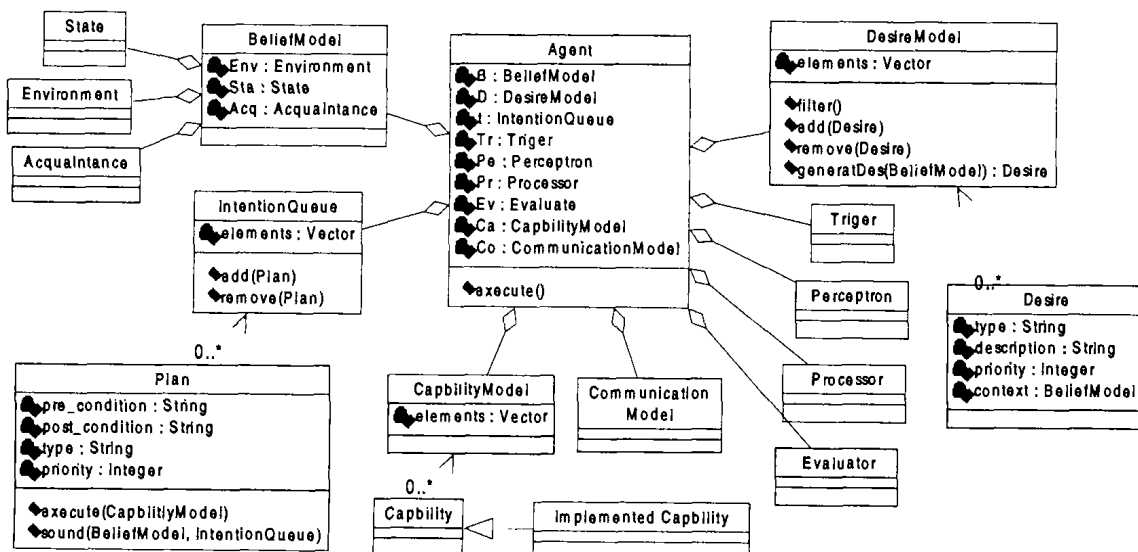


图 2 基类 BDI Agent 类图

## 2.2 可动态加载的能力组件

在上述 Agent 模型中能力组件定义为 Agent 具有的、为完成一定目标而进行的一组逻辑相关的行为的集合,包括 Agent 的内在本质属性(通信、规划等)和外在行为动作(竞价、协商等)。内在本质属性决定了 Agent 的基本功能和类型,例如通讯能力组件使 Agent 具有通讯能力;外在行为动作使 Agent 具有执行智能动作的功能,即为与其他 Agent 进行协作、协调进而完成特定目标的功能。

基于能力组件的 Agent 模型在系统的分析设计、实现和运行阶段对系统提供支持。在一个应用系统的分析设计阶段,通过分析系统需要的 Agent 类型和 Agent 具有的功能,从已设计好的能力组件库中选择能力组件,通过可视化界面进行选择、编辑。例如,Agent 的社会性要求 Agent 具有通讯能力,因此 Agent 从库中选择通讯能力组件加载;在基于 Agent 的供应链管理系统客户向零售商下订单的场景,可为客户提供查询和购买能力组件、为零售商提供分销组件,他们就可以在该场景下进行下订单的交互过程。在系统实现阶段,将能力组件的定义转换为接口,不同系统的不同开发者根据实际需要为能力组件实现不同的代码(完成不同的功能),实现的代码经编译加入能力组件库中;在系统运行阶段,不同系统的 Agent 根据需要查找、匹配、动态加载适当的能力组件,成为具有能力的 Agent。

我们采用 RDFS 对能力组件进行定义。RDFS<sup>[14]</sup>是资源描述框架的模式,是对结构化的数据进行编码、交换和重用的有效工具。能力组件定义包括(限于篇幅省略能力组件的 RDFS 定义):

CapabilityType:定义能力的类型,包括内在属性和外在属性;CapabilityID:能力的 id 号,对能力进行唯一标识;CapabilityDescription:能力描述,帮助人对能力组件的理解;Capa-

bilityKeyword:能力关键字,有助于能力的查询、匹配;CapabilityActionsList:该能力组件所能完成的动作的列表,列表内包括多个动作。

Action:动作的定义;ActionName:动作的名称;Action-Return Type:动作所产生的结果;ActionParameterList:动作所需要的参数的列表;Parameter:参数的定义;Parameter-Name:参数的名称;ParameterType:参数的类型。

使用 RDFS 定义 Agent 的能力,使用 RDF 解析器对用 RDF 描述的能力组件解析,通过可视化的表示,对定义好的能力组件进行编辑操作。与用 DAML-S<sup>[20]</sup>描述的 Web Service 类似,利用 Web Service 的匹配、合成技术<sup>[21]</sup>,Agent 就可以自动处理,发现、查询、比较、匹配、合并、加载根据 RDFS 定义、用 RDF 描述的能力组件。

作为例子我们用能力组件方法描述 Middle Agent 提供的服务。Middle Agent<sup>[22]</sup>是构建多 Agent 系统的一类重要的 Agent,具有接受其它 Agent 向其注册地址信息、提供查询服务的能力。如图 3 所示(限于篇幅只进行部分定义)。根据图 3 定义,该能力组件的唯一 ID 号为 0001,以与其他能力组件进行区别;类型为内在属性(INTRINSTIC),加载该能力组件的 Agent 可以接受系统中 Agent 向其注册的地址信息,为完成该动作,需要两个输入参数:注册 Agent 的 ID 和地址信息。系统生成基类 Agent 并根据上述 RDF 描述转换为能力组件的 Java 实现,基类 Agent 通过 Java 的类加载机制,加载能力组件,实现能力组件运行时动态加载的特性,保证系统再次开发和维护时可以不间断运行。一个动态加载 MiddleAgent 能力组件的部分代码如下:

```
ClassLoader loader = new CapabilityLoader(host, CapabilityName);
Object middleAgent = loader.loadClass("MiddleAgent", true).
    newInstance();
middleAgent.getDeclaredMethod("register", null).invoke(middleAgent, null);
```

<rdf:Description rdf:about="http://is.pku.edu.cn/2003/03/MiddleAgentCapability#MiddleAgentCapability">
<rdf:type rdf:resource="http://is.pku.edu.cn/2003/03/capability#Capability"/>
<rdfs:comment>A capability model for the middle agents</rdfs:comment>
<ca:CapabilityType>CapabilityType.CAPABILITYTYPE_INTRINSTIC</ca:CapabilityType>
<ca:CapabilityID>0001</ca:CapabilityID>
<ca:CapabilityDescription>The capability to store capability registrations and answer request by other agents</ca:CapabilityDescription>
<ca:CapabilityKeyword>MiddleAgent:register:registerCapability:findCapability:getAgentAddress</ca:CapabilityKeyword>
<ca:CapabilityActionsList>
<ca:ActionList><rdf:li rdf:resource="http://is.pku.edu.cn/2003/03/MiddleAgentCapability#Action1"/>..... </ca:ActionList>
</ca:CapabilityActionsList>
</rdf:Description>
<rdf:Description rdf:about="http://is.pku.edu.cn/2003/03/MiddleAgentCapability#Action1">
<rdf:type rdf:resource="http://is.pku.edu.cn/2003/03/capability#Action"/>
<rdfs:comment>accept the Provider Agent's registration: first is the agent's address: ip or something else</rdfs:comment>
<ca:ActionName>register</ca:ActionName>
<ca:ActionReturn Type>int</ca:ActionReturn Type>
<ca:ActionParameterList>
<ca:ParameterList><rdf:li rdf:resource="http://is.pku.edu.cn/2003/03/MiddleAgentCapability#Action1-Parameter1"/>
<rdf:li rdf:resource="http://is.pku.edu.cn/2003/03/MiddleAgentCapability#Action1-Parameter2"/>
</ca:ParameterList></ca:ActionParameterList>
</rdf:Description>
<rdf:Description rdf:about="http://is.pku.edu.cn/2003/03/MiddleAgentCapability#Action1-Parameter1">
<rdf:type rdf:resource="http://is.pku.edu.cn/2003/03/capability#Parameter"/>
<ca:ParameterName>aid</ca:ParameterName><ca:ParameterType>AgentID</ca:ParameterType>
</rdf:Description>
<rdf:Description rdf:about="http://is.pku.edu.cn/2003/03/MiddleAgentCapability#Action1-Parameter2">
<rdf:type rdf:resource="http://is.pku.edu.cn/2003/03/capability#Parameter"/>
<ca:ParameterName>aAddress</ca:ParameterName><ca:ParameterType>AgentAddress</ca:ParameterType>
</rdf:Description></rdf:RDF>

图3 Middle Agent 能力组件

由基类 Agent 生成的 Agent 实例不具有任何能力组件,若要赋予其特定的企业角色、具有企业流程的业务知识,可按

照企业角色能力组件的定义,实现企业角色应具有的业务流程相关知识的能力组件,Agent 加载能力组件后,就具有供应

链领域业务流程的知识和操作能力,对应供应链上的企业角色,完成供应链的业务流程。

### 2.3 Agent 模型运行解释器

类似文[16]我们给出一个上述 Agent 模型的运行解释器,目的是在抽象的层次上解释该 Agent 模型的运行机制,并缩小与现实系统的距离。这个抽象运行解释器包括文[16]中的 Agent 的信念 B、愿望 D、意图 I,还引入了能力组件 Ca、通讯组件 Co、触发器 Tr、感知机 Re、评估器 Ev、处理机 Pr 与实现相关的组件。运行解释器描述如下:

```

CBDI-interpretor
Initialize-state();
Do
  B::= update-belief(Re,Co,B);
  D::= deliberate(Tr,B,D,I,Ca);
  I::= evaluate(Ev,B,D,I,Ca);
  Ca::= load-capabilities(Ca,Co,I);
  execute(Pr,B,D,I,Ca);
  get-new-external-events(Re,Co);
  drop-success-attitudes(B,D,I,Ca);
  drop-impossible-attitudes(B,D,I,Ca);
  unload-unnecessary-capabilities(B,D,I,Ca);
Until quit
    
```

根据运行解释器,Agent 通过感知机 Re 和通讯组件 Co 感知外界环境,更新信念 B;通过触发器监视信念变化,并根据当时精神状态(B、D、I)和具有的能力组件 Ca 形成新的愿望 D;然后由评估器 Ev 根据 B、D、I、Ca 评估这些愿望是否值得执行,经评估的愿望形成规划 I;处理器 Pr 负责分配和调度能力组件完成这些规划;在处理机执行计划的过程中感知机使用通讯组件继续接受外部的事件,Agent 更新精神状态和能力组件;根据 Agent 对意图的不同承诺策略<sup>[16]</sup>和 Agent 现有的能力,去掉已经满足的愿望和成功执行的计划,去掉不切实际的愿望和不可实现的计划,去掉不必要的的能力组件。

### 3 ASCM 体系结构

ASCM 系统体系结构采用分层结构,分为系统层、域独立组件层和应用组件层。系统层由 Java 虚拟机构成,ASCM 系统基于 Java 语言,建立在 Java 虚拟机之上,因此保证了 ASCM 系统能在不同操作系统和硬件平台上运行,具有平台无关性;域独立组件层由与具体应用领域无关的一些组件和基础设施组成,提供基本的数据连接、组件通讯、组件查找等基本服务;应用组件层包含供应链管理逻辑,提供组成供应链各个部分的实体库,不同企业根据其在供应链上的不同位置初始化属于自己的角色。由此 ASCM 具有平台和数据库的无关性,三层体系结构保证系统开发者可以只关心应用领域层的供应链管理逻辑,符合企业实际应用需要,适应未来软件发展趋势。

ASCM 的应用层组件由具有不同能力组件的 Agent 组成,根据能力组件的不同,系统中的 Agent 分为三类:

- 结构 Agent 是描述供应链中的实体,例如供应商、制造企业、分销中心、零售商和运输商等,用 SAgent 表示;
- 控制 Agent 用来定义供应链中管理产品流动的各种控制策略,例如需求、供应、信息和物料的流动等,用 CAgent 表示;
- 接口 Agent 为系统中的 Agent 与外界交互提供接口和方法,用 IAgent 表示。

将 ASCM 系统应用于供应链后结构如图4所示,首先抽取供应链上的实体间的交互过程为能力组件,Agent 加载能力组件后成为具有业务知识的 Agent,代表供应链上相应的实体完成相应业务流程;控制 Agent 实现供应链上实体间的

交互控制策略,协调结构 Agent 间交互过程;接口 Agent 负责提供结构 Agent 和控制 Agent 与外界通讯的接口。

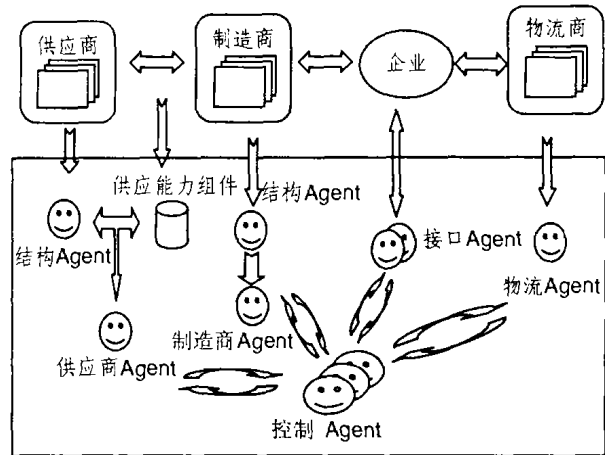


图4 ASCM 系统结构

可以为一个 Agent 加入供应能力组件使之成为供应商结构 Agent,可以为它加入市场营销策略组件使之成为营销控制 Agent,也可以为它加入消息中继组件使之成为中继接口 Agent。在供应链业务流程发生改变的时候可以按照需求完成新的能力组件的开发,在必要的时候动态加载这个能力组件,增加了系统的可扩展性。

ASCM 中组成供应链的每个企业可以根据自己的业务流程进行建模,定义业务流程描述,系统根据不同企业和他们定义的不同业务流程描述生成对应于现实世界的不同实体和业务逻辑,ASCM 系统的应用为企业高层的战略决策提供数据分析和参考支持,改善供应链中不合理的流程,提高生产效率,降低运营成本,增强了供应链上企业之间以及企业信息系统之间的信息交互。

### 4 一个订单流程

我们通过供应链系统中一个订单驱动的典型业务流程来说明 ASCM 系统所具有的性质。其中涉及到的 Agent 类型和能力组件如表1所示。

客户向零售商查询产品并下订单的流程如下:

1. 客户 SAgent Cu 通过零售商客户服务 IAgent I 查询产品;
2. SAgent Cu 填写产品的型号、规格、数量等信息,提交订货单;
3. SAgent Cu 确认订单后由 IAgent I 产生订单 CAgent O,并负责确定订单内容、协调修改订单、跟踪管理与监控订单执行全过程状态;
4. CAgent O 根据自身信念(产品型号和数量等信息)向核心企业总部 CAgent Co 发送要货请求;
5. CAgent Co 根据 SAgent A 的地址信息和配送算法选择最近的配送中心 SAgent D1;
6. SAgent D1 获得请求后与其库存控制 CAgent V 协调,如果库存足够并满足库存控制策略则转到9 否则继续;
7. SAgent D1 向总部 CAgent C 发送缺货信息,请求要货;
8. 总部 CAgent C 重新使用配送算法查找距离近配送中心 SAgent D2,转6(如果所有配送中心均不能满足要求,则从核心企业总部发货,假设总部能及时从供应商处取得产品);

表1

类型	名称	具有的能力组件
结构 Agent	客户 SAgent Cu	请求产品信息 Cu-1
	核心企业 SAgent Co	提供产品信息 Co-1
		订单管理 Co-2
		分销计划 Co-3
	配送中心 SAgent D	库存管理 D-1
运输商 SAgent T	物流运输 T-1	
控制 Agent	定单 CAgent O	协调、跟踪、监控订单执行 O-1
	库存控制 CAgent V	自动补货 V-1
接口 Agent	零售商客户服务 IAgent I	提供产品信息界面 I-1

9. 产生运输 SAgent T 根据总体策略、服务等级提出优

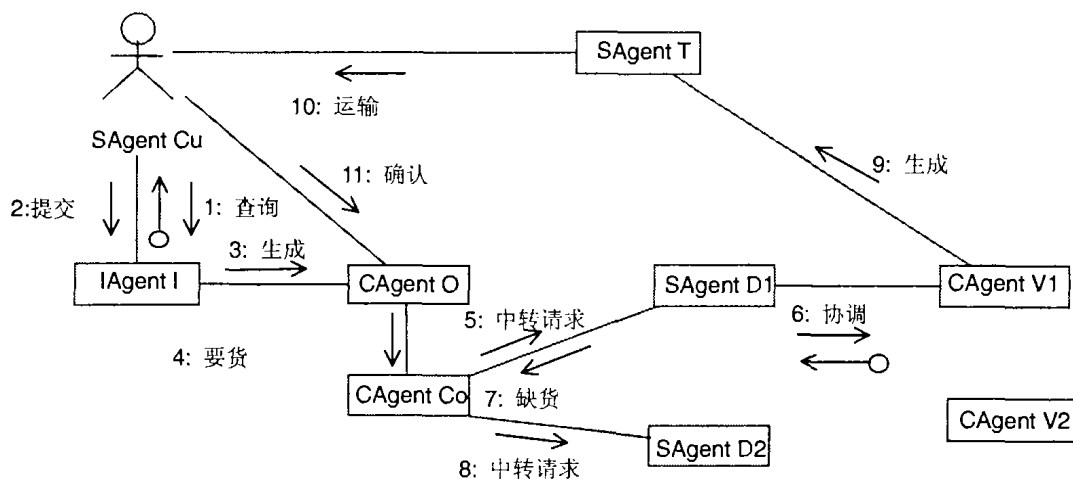


图5 订单流程的 UML 描述

## 5 与相关工作比较

在 ISCM<sup>[7]</sup>系统中,供应链被映射为一个由中心协调和一些合作的 Agent 组成的网络,Agent 通过通信协调语言 COOL(CO-Ordination Language)<sup>[24]</sup>进行协作和协商,主要研究了 Agent 在供应链背景下的协调机制.Kasbah<sup>[25]</sup>是一个有关电子商务的基于 Web 的多 Agent 系统,系统中的 Agent 种类分为两种:买方 Agent 和卖方 Agent,研究重点电子市场中 Agent 的交互和竞争.MetaMorph II<sup>[13]</sup>应用基于分布式代理框架并以协调代理为中心的混合结构,通过核心企业将合作伙伴、供应商、客户动态地和供应链中各自的协调代理集成在一起,提出了混合多代理框架系统结构.Swaminathan<sup>[8]</sup>提出并区分了供应链中两类不同的元素:结构元素和控制元素,提出结构 Agent 和控制 Agent 的概念,使用这两类 Agent 对供应链进行建模.AARIA<sup>[26]</sup>是为军用制造而开发的工厂模拟调度多 Agent 系统,系统中的 Agent 具有共同可重用的部分和相同的结构,可以根据环境的变化加入所需要的 Agent,但是 Agent 的类型在系统设计实现的时候就确定了,实现的代码也不易更改。

以上工作和系统已应用于供应链的模拟和管理,它们侧重于供应链背景下 Agent 的协调、交互和对供应链的模拟、集成,实现的系统具有一定的可重用性,但 Agent 的类型、所代表的供应链上的实体以及 Agent 做出的智能行为的策略是在系统设计时就确定的,不能根据供应链业务流程的改变而做相应变化,因此它们对供应链具有动态性考虑不够,系统灵活性和扩展性尚显不足。与以上工作相比,ASCM 扩展 BDI

化模型和具体运输计划;

10. SAgent T 负责将产品运输到客户 SAgent Cu 处;

11. SAgent Cu 通过 IAgent I 向 CAgent O 确认到货,同时 CAgent O 被归档。

订单流程的 UML 描述如图5所示。订单流程中1-3步基于 Web 的方式实现,多在企业信息系统前端以 B/S 结构实现,以后各步采用多层体系结构实现供应链业务流程,当企业业务流程改变时,例如如果企业库存管理策略由配送中心分别管理库存改变为供应商管理库存(VMI),核心企业统一管理库存,在上述要货过程中只需根据需要更新能力组件的实现,并由核心企业 SAgent Co 加载库存管理能力组件 D-1,就可以适应新的业务流程,而不需要改写整个系统。

Agent 模型,加入可动态加载的能力组件,在基于 Agent 的系统分析、设计、开发和运行阶段支持供应链的动态性和实时性,同时 ASCM 的三层体系结构也保证了系统的灵活性和可扩展性。

**结论和进一步工作** 在本文中我们提出一种具有可动态加载能力组件的多 Agent 模型,给出了模型中 Agent 的定义、能力组件的定义和应用以及 Agent 模型的运行解释器。设计了一个基于此模型的企业供应链管理平台 ASCM,根据供应链中的实体、策略和实际需要,将 ASCM 中的 Agent 分为三类,具有共同的基类 Agent,运行中 Agent 可以动态加载 RDF 描述的能力组件到自己的代码中,执行不同的动作、完成不同的功能。通过给出一个供应链中订单处理业务流程说明,该多 Agent 模型在供应链建模和设计中的应用。

以下几个问题是我们今后研究的重点:

- (1) Agent 如何在用 RDF 表示的能力组件库中快速查找、匹配、合成多个能力组件以完成特定目标;
- (2) 多 Agent 环境中 Agent 完成目标时对任务的分解和 Agent 间的协调机制;
- (3) 在加入能力组件的情况下扩展 Agent 通讯语言,使得关于能力组件的查询和协调在语义层得以实现。

软件 Agent 的体系结构在理论上的研究还不成熟,但是 Agent 技术已在实际应用中取得了显著的成果,并且被证明具有广阔的发展潜力。

## 参考文献

1 Barbuceanu M, Fox MS. The Information Agent: An Infrastruc-

- ture Agent Supporting Collaborative Enterprise Architectures. In: Proc. of Third Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Morgantown, West Virginia, IEEE Computer Society Press, 1994
- 2 Parunak H V D. Practical and Industrial Applications of Agent-Based Systems. Environmental Research Institute of Michigan (ERIM), 1998
  - 3 Barbuceanu J, Fox J S. The design of a coordination language for multi-agent systems. In: Intelligent Agents, LNAI Vol. 1193. Heidelberg: Springer-Verlag, 1996. 341~356
  - 4 Bond, Bruce. C-Commerce: The New Arena for Business Applications, 1999, Gartner Group Research. <http://www.gartner.com>
  - 5 Phillips, Charles, Meeker. The B2B Internet Report-Collaborative Commerce, 2000, Morgan Stanley Dean Witter Internet Research
  - 6 Behrens S. Electronic Collaborative Commerce An Overview of Enabling Technologies, 2000. <http://citeseer.nj.nec.com/behrens00electronic.html>
  - 7 Fou J. Web Services and Collaborative Commerce Collaborate or Die. <http://www.webservicesarchitect.com/content/articles/fou01.asp>
  - 8 Woodridge M, Jennings N R. Intelligent Agent: Theory and Practice. The Knowledge Engineering Review, 1995, 10(2): 115~152
  - 9 Roboam M, Fox M S. Organization modeling as a platform for Multi-agent Manufacturing Systems. In Computer Applications in Production and Engineering: Integration Aspects. Holland: Elsevier Science Publishers. p581~591
  - 10 Fox M S. The Integrated Supply Chain Management Project. <http://www.eil.utoronto.ca/iscm/index.html>
  - 11 Maturana F, Shen W, Norrie D H. MetaMorph: an adaptive agent-based architecture for intelligent manufacturing. International Journal of Production Research, 1999, 37(10): 2159~2173
  - 12 Swaminathan M. Modeling Supply Chain Dynamics: A Multi-Agent Approach. Decision Sciences, 1997, 29(3): 607~632
  - 13 Helper S. How much has really changed between U. S. manufac-

- turers and their suppliers? Sloan Management Review, 1991, 32(4): 15~28
- 14 Bratman M, et al. Cambridge: Harvard University Press, 1987
  - 15 Cohen P R, Levesque H J. Intention is choice with commitment. Artificial Intelligence, 1990, 42(2-3): 213~261
  - 16 Rao A S, Georgeff M. BDI Agents: from theory to practice. In: Proc. of the First Intl. Conf. on Multi-Agent Systems (ICMAS'95), San Francisco, CA, 1995. 312~319
  - 17 Rao A S, Georgeff M. Modeling rational agents within a BDI-architecture. In: Proc. of the Second Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR'91), 1991. 473~484
  - 18 Vidal J M, Buhler P A, Huhns M N. Inside an Agent. IEEE Internet Computing, 2001, 5(1): 82~86
  - 19 Resource Description Framework (RDF) Schema Specification. <http://www.w3.org/TR/PR-rdf-schema>
  - 20 DARPA Agent Markup Language. DAML-S 0.9 Beta, 2003. <http://www.daml.org/services>
  - 21 van der Aalst W M P. Don't go with the flow: Web Services composition standards exposed. issue of IEEE Intelligent Systems, 2003. 1~2
  - 22 Sycara K, Decker K, Williamson M. Middle-Agents for the Internet. In: Proc. of the 15th Joint Conf. on Artificial Intelligence, 1997. 578~583
  - 23 Finin T, Fritzson R, McKay D, McEntire R. KQML as an Agent Communication Language. In: Proc. of the 3rd Intl. Conf. on Information and Knowledge Management (CIKM'94). USA: ACM Press, 1994. 456~463
  - 24 Chavez A, Maes P. Kasbah: An agent marketplace for buying and selling goods. In: Proc. PAAM'96, London, UK, 1996. 75~90
  - 25 Parunak H V D, Baker A D, Clark S J. The AARIA agent architecture: An example of requirements-driven agent-based system design. In: Proc. of the First Intl. Conf. on Autonomous Agents (ICA'97), 1997. 482~483

(上接第7页)

数字视频用户。收端进行了 UCL 解析、下载控制、基于 UCL 的语义代理实验研究。

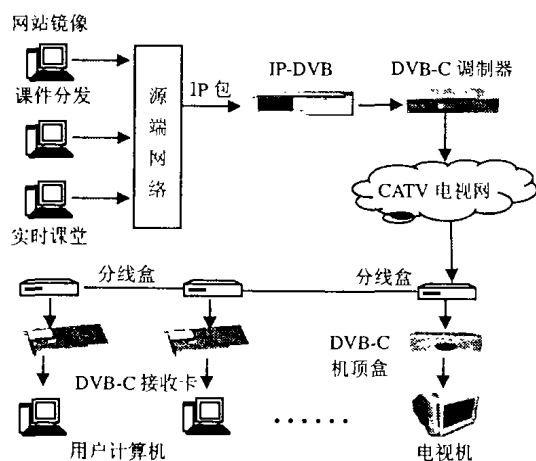


图3 实验网络

**结论** 通过本课题的研究,可以得出以下几个主要结论:

(1)建立广播网络有明确的应用背景,市场对此有十分紧迫的需求;

(2)实验结果表明,按照信息共享理论和大规模并播技术构造的无冲突的广播型网络在理论上是可行的,是信息资源在时间、空间、内容的高度整合;

(3)并播技术与 UCL 技术在普通 DVB-C 网络传输是兼容的。如果更换 QAM 调制器,本实验结果容易推广到 DVB-S(卫星传输)、地面无线传输 DVB-T、DMB-T 等网络;

(4)UCL 技术对于消除信息垃圾和消除语义鸿沟十分有效,为用户主动服务成为可能;

(5)广播网络在经营上容易做到独立和可持续发展。

广播型网络中收端的进一步研究,如用户兴趣图谱建立,用户肖像模型建立和基于 UCL 的语义代理等问题在国家 863 项目和国家自然科学基金项目资助下正在有序地进行。

本项目的研究和本文的成文整个过程中都得到了李幼平院士的精心指导和无私的帮助,文中引用了李幼平院士的很多观点,特在此表示感谢!

## 参考文献

- 1 李幼平. 共享信息的第二类网络[J]. 中国工程科学, 2002, 4(8): 8~11
- 2 Foster I. the Grid: computing without bounds [J]. Scientific American, 2003, 4: 79~85
- 3 Foster I. The Grid: A New Infrastructure for 21st Century Science [J]. Physics Today, 2001, 2(55): 42~47
- 4 高杨, 李幼平. UCL 理念及其系统设计[J]. 电视技术, 2001, 2(224): 38~41
- 5 马建国, 李幼平, 等. 国家规模远程教育平台实验研究[J]. 中国远程教育, 2002, 7(186): 38~40
- 6 李幼平, 马建国, 等. 国家教育平台[J]. 数据广播, 2002, 2(16): 1~5
- 7 高杨. 互补结构的信息共享系统[D]. [博士论文]. 北京理工大学, 2000
- 8 马卫东, 李幼平. 数据广播传输体系结构研究[J]. 计算机工程与应用, 2001, 24: 93~96
- 9 Shannon C E. A Mathematical Theory of Communication [J]. Bell Syst. Tech. J. 1948, 27: 379~423 (Part I), 623~656 (Part II)
- 10 Berners-Lee T. The Semantic Web [J]. Science American, 2001(5): 21~24
- 11 Barabási L-L, Bonabeau E. Scale-Free Networks [J]. Science American, 2003(5): 50~59
- 12 ETSI. Digital Video Broadcasting, DVB specification for data broadcasting [S]. EN 301 192 V1. 2. 1 (1999-06)
- 13 ETSI. Digital Video Broadcasting, Implementation guidelines for Data Broadcasting [S]. TR 101 202 V1. 1. 1 (1999-02)