

摘 要

随着计算机和网络技术的不断发展,使得网络上商务行为越来越普遍。电子商务由于充分地利用了 Internet 的优势,使企业与企业、用户与企业之间的距离变为零,这种零距离的交易给企业带来了无限的商机。

电子商务的成功与否,协商、谈判等活动起的作用是极其重要的。协商是一件非常消耗人力、物力以及时间的事情。为了节省买卖双方由于协商所花费的大量时间,提高交易效率和降低交易成本,有必要实现电子交易的自动或半自动协商。由于 Agent 具有自治、移动、灵活等特点,能在不同的买方或卖方中协商出一个最佳的交易方案。为此本文给出了一个基于 Agent 的电子交易框架,围绕着买卖双方对商品的协商问题,讨论 B2C 电子商务活动的部分过程,其中包括用户搜索商品信息、双方的协商、交易,以及商务活动当中买卖双方对商品信息的 management 问题。框架中以 Agent 代替交易双方的参与人员,形成由卖方多 Agent 系统和买方多 Agent 系统组成的新营销模式。

由于协商本身又是一个很广泛的研究范畴,本文在基于 Agent 代替买卖双方对商品进行交易框架的前提下,设计并实现了其中的协商模型。

第一章:概述了本文研究的背景,分析了研究的意义和必要性,介绍目前本课题的研究现状,讨论分析了目前研究现状存在的问题,以及本文研究的主要内容和本文的组织结构。

第二章:详细的介绍了与文本有关的背景知识。这些知识包括电子商务技术知识、Agent 技术知识、移动 Agent 技术知识。这些背景知识是本文的论述基础。

第三章:提出了基于 Agent 协商模型。这一章内容既是本文的重点也是本文的难点章节。目前已有多种协商的理论、方法,主要是基于对策论和基于启发式的协商,但都不完善。本文主要研究双边多项目协商问题,是基于意图的协商。文章给出了相应的协商模型、协商协议、协商算法、协商流程等。

第四章:提出了基于 Agent 协商系统的体系结构。这一章内容为本文的重点章节。本章给出了由卖方多 Agent 系统和买方多 Agent 系统代替交易双方的参与人员组成的新交易模式,讨论了电子商务活动的部分过程,其中包括搜索、协商、交易等活动。

第五章:介绍了本文提出的基于 Agent 协商原型系统的实现。主要介绍了系统运行的环境、系统的 UML 图、以及实例运行部分界面图。

第六章:总结了本文的主要研究的成果,以及创新点,并指出进一步需要研究的工作。

关键词: 电子商务, Agent, 移动 Agent, 协商

Abstract

Along with the fast development of computer and network technologies, Business activities through networks are becoming more and more popular. E-business took full advantage of the power of internet, which shorten the distance between companies and customers to zero, and this zero-distance trading brought companies infinite business opportunities.

The success or failure of E-Business seriously depends on business negotiations. Negotiation is a matter that needs many resources such as persons, money and time. In order to save the time of both merchants and customers, improve trade efficiency and reduce trade cost, it is necessary to realize automatic negotiations or half-automatic negotiations during the electronic trade. Agent has the features of autonomic, mobile and flexible, which is useful to give a best trade solution during the negotiation between seller and buyer. Therefore, this thesis gives an Agent-based electronic trade framework, and discusses partial process of B2C E-Business activities about negotiations including user searching products, negotiations, trades and the management of product information. In stead of persons, agents act as traders in our framework, which forms a new commerce mode consisting of a multi-agents buyer system and a multi-agents seller system.

Because negotiation itself is also a wide research field, this thesis designed and implemented a negotiation model under the preconditions of agent-based product trade.

The 1st chapter summarized the research background of this thesis, analyzed the sense and necessity of the research, introduced the status of present research, discussed the existing problems in present research and the main contents and structure of this thesis.

The 2nd chapter introduced the relative background knowledge in detail. The knowledge includes E-Business technologies, agent technologies and mobile agent technologies. These background technologies are the basic knowledge of this thesis.

The 3rd chapter put forward the agent-based negotiation model. The content of this chapter is the key point and the difficult point of the whole thesis. There are many theories and methods of negotiation which are mainly game-based and heuristic-based negotiation and not perfect. This thesis focuses on the problems about two-sides multi-projects negotiations which is based on intention. The thesis gave the corresponding negotiation model, negotiation protocol, negotiation algorithm, negotiation process and so on.

The 4th chapter put forward the architecture of agent-based negotiation system. It is the key chapter. This chapter gave the new trade model consisting of seller multi-agent system and buyer multi-agent system instead of persons, and discussed partial process of B2C E-Business activities about negotiations including registry, negotiations, trades, etc.

The 5th chapter introduced the implementation of the agent-based negotiation prototype system including the runtime environment of the system, UML diagrams of the system and some interfaces of the system.

The 6th chapter summarized the main research results, the innovation points of this thesis, and indicated the future work.

Key words: E-Business, Agent, mobile agent, negotiation.

第一章 绪 论

1.1 引言

当代的信息技术，特别是以 Internet 为核心的网络技术的发展与应用，使社会步入了全新的网络经济时代。电子商务作为网络经济时代适应网络虚拟市场发展的企业经营管理新模式，将突破传统的商务模式，改变传统的竞争规则。电子商务（E-Business）是利用现代计算机通信网络提供的信息网络平台在网上进行的商务活动，电子商务改变了传统的买卖双方面对面的交流方式，打破了旧的经营管理模式，可以使企业在世界范围内为用户提供每周 7 天、每天 24 小时的全天候服务。电子商务由于充分地利用了遍布全球的 Internet 的优势，使企业与企业、用户与企业之间的距离变为零，这种零距离的交易给企业带来了无限的商机。

电子商务的成功与否，协商、谈判等活动起的作用是极其重要的。协商是一件非常消耗人力、物力以及时间的事情。传统商业交易需要消费者或企业对公司对买卖商品或服务的价格、交货时间、质量等交易条件进行协商。在消费者对消费者（C2C）、企业对消费者（B2C）、企业对企业（B2B）的电子商务中，为了节省买卖双方为协商完成交易所花费的大量工作和时间，提高交易效率和有效降低交易成本，有必要实现电子交易的自动或半自动协商。

由于 Agent 具有自治、移动、灵活等特点。因此基于 Agent 的电子商务技术的研究已成为现在研究的热点之一。Agent 技术是 90 年代在网络技术和 Internet 发展以及决策支持系统技术的基础上发展起来的，是具有很强的自主能动性的软件技术。但是目前基于 Agent 技术的电子商务的研究，大多数都侧重于比较购物（comparison shopping）或者为用户查找所需商品信息（例如 BargainFinder 和 Jango），或者为用户推荐相关商品信息（例如 Firefly），类似于传统网站中的搜索引擎，缺乏对现实商务活动完整过程的考虑，特别是缺乏价格协商的机制和缺乏个人的对商品属性偏好不同的机制，这已经开始限制电子商务的发展。

为此本文提出了一种基于 Agent 协商的电子交易框架，主要围绕着企业与企业对商品的协商问题，讨论 B2C 电子商务活动的部分过程，包括用户搜索商品信息、双方的协商、交易，以及商务活动当中买卖双方对商品信息的 management 问题。框

架中以 Agent 代替交易双方的参与人员,形成由卖方多 Agent 系统和买方多 Agent 系统组成的新营销模式。

由于协商本身又是一个很广泛的研究范畴,本文主要针对网上交易这一特定的领域,以 Agent 代替买卖双方的参与人员对多商品交易进行研究,即双边多项目协商问题,设计并实现了它的协商模型,该协商模型是基于意图的协商。由于该协商模型,具有一般性,因此在应用中有着广泛的适用空间。

1.2 课题研究的现状

协商有时亦称谈判,主要涉及经济、政治、经济、军事等领域。它大量存在于人们的生活中,是一种解决分歧与争端有效手段。关于谈判谋略的研究,在我国有着悠久的历史。《孙子兵法》、《老子》及《鬼谷子》是中国古代具有深邃东方谈判谋略思想的三部代表著作,它们主要从定性的哲学高度阐述了有关东方谈判谋略的体系,其中包括谈判过程的哲学思想、谈判原则及谈判规律等等。

协商是通过相关的结构化信息的交换,形成公共观点和规划一致的过程,即协商是为达成共识而减少不一致性或不确定性的过程。协商是多 Agent 系统实现协作、冲突消解和矛盾处理的关键环节,是多 Agent 系统研究的主要内容之一。目前主要有基于对策论的协商、启发式协商和基于意图的协商。

1.2.1 基于对策论的协商

在多 Agent 系统中,协商的含意有多种理解,一种认为子问题和资源的指派是协商,另一种则认为 Agent 之间一对一直接协商。所有协商活动的目的是在一组独立工作的 Agent 间构建协作,协商策略取决于具体的 Agent。

1.2.1.1 对策论

对策论的研究是以 Von Neumann 的效用理论为基础的,研究二人零和对策。在 60 年代初,对策论风靡全球。到了 70 年代中期开始,突破传统对策论偏重完备知识、单遇、Nash 平衡的方法,将注意力转向多步对策、重复对策,以求利用交互的历史信息,弥补知识的不完备,形成当今对策论研究的热点。

对策论以效用集为基础, 如果对于任一目标 g , 使某个 Agent_i 的效用 $u(i, g)$ 最大, 却不能使另一个 Agent_j 的效用 $u(j, g)$ 最大, 则协调就成为必要的。

若 $u(X \cup Y) \leq u(X) + u(Y)$, 协作就可以实现。

一个非合作 n 人对策论由以下三个因素确定:

- (1) 局中人集 $I = \{1, \dots, n\}$;
- (2) 每个局中人有一个纯策略的有限集:

$$S = \{S(i)\} = \{s_1(i), \dots, s_{m_i}(i)\}, \quad i=1, \dots, n;$$

- (3) 每个局中人 i 有一个赢利函数 $P_i, i=1, \dots, n$.

当每个局中人选定一个策略 $s(i)$ 后, 就形成对策的一个态势 $s=(s(1), \dots, s(n), s(i) \in S$. 每个局中人得到的赢利为 $P_i = P_i(s)$. 因此, 一个非合作 n 人对策表示为: $\Gamma = [I, \{S_i\}, \{P_i\}]$

下面规定符号:

$$S_{\parallel} t(i) = (s(1), \dots, s(i-1), t(i), s(i+1), \dots, s(n))$$

定义 1: 平衡态势或平衡点。设 s^* 是非合作 n 人对策 Γ 的一个态势, 若对任一 $i \in I$ 和任一 $s(i) \in S_i, (s(i) = s_k(i), (k=1, \dots, m_i))$, 有

$$P_i(s^*_{\parallel} s(i)) \leq P_i(s^*)$$

则称 s^* 是 Γ 的一个平衡态势或平衡点。在一般对策中, 平衡点不一定存在。

定义 2: 混合策略。对每一个 $i \in I$, 设 $x(i)$ 是 i 的一个混合策略, 也就是定义在 s_i 上的一个概率分布:

$$x(i) = (x_1(i), \dots, x_{m_i}(i)),$$

$$x_k(i) \geq 0$$

$$\sum x_k(i) = 1,$$

称下 $x = (x(1), \dots, x(n))$ 为对策 Γ 的一个混合策略局势,

记 $x_{\parallel} z(i) = (x(1), \dots, x(i-1), z(i), x(i+1), \dots, x(n))$ 。

一个混合策略表示为: $\Gamma' = [I, \{x_i\}, \{P_i\}]$

定义 3：混合策略平衡点。设 $E=E_i(x)$ 是 i 在 x 局势下的期望赢利，若对每一个 $i \in I$, $x(i) \in X$, 有

$$E_i(x^* || x(i)) \leq E_i(x^*) \text{ 则称 } x^* \text{ 为混合策略下的一个平衡点。}$$

Nash 在 1951 年利用 Brower 不动点定理证明了混合策略平衡点必定存在，这种平衡点又称为 Nash 平衡点。

Rosenschein 首先用对策论方法讨论理性 Agent 协调、协作的条件，提出 Agent 的理性假设有：

- (1) 最小理性假设：假设对方行为是随机的。
- (2) 单独理性假设：假设对方也是理性的。
- (3) 唯一理性假设：假设对方的行动已知。

满足唯一理性假设必同时满足单独理性假设。Rosenschein 的理论认为：

- (1) 各个 Agent 追求本身效用的最大化。
- (2) 效用（或赢得）矩阵是公共知识。
- (3) Agent 是理性得，即满足三个理性假设。

在无通信的情况下，Agent 可以根据对方及本方效用模型，求得 Nash 平衡解，因而可以有效协调，选择适当的行为。Rosenschein 的理论 Agent 交互理论开创运用对策论解决多 Agent 协调问题。

但是该理论存在下列缺陷：

- (1) 完备知识假设，即认为各 Agent 效用矩阵为公共知识，这对一般的多 Agent 系统难以做到。
- (2) 单遇假设，即两个 Agent 之间只有一次交互。没有考虑交互历史信息的影响。
- (3) 诚实假设，即认为各 Agent 对各自的承诺完全信任，这与实际情况不相符。

1.2.1.2 面向领域的协商理论

Zlotkin 总结了以往协商理论的弱点，提出一个面向领域的协商理论。

Zlotkin 的协商理论假设：

- 各主体追求本身效用最大。
- 知识完备。
- 无历史信息。
- 目标集固定。
- 协商在两主体之间同时进行。
- 主体操作集相同。
- 世界仅当主体操作之后发生变化

在协商过程中,若存在一个双方满意的分配,可以得到一个联合规划,使协商终止的条件是:

(1)达成协议:如果 $Utility(P(i, t)) \leq Utility(P(i, t-1))$, 这一协商过程将在有限步内结束。

(2)冲突: 如果 $Utility(P(i, t)) = Utility(P(i, t-1))$, 则无法达成协议

1.2.1.3 最佳平衡协商方法

传统的协商是基于 Nash 平衡的, 它的缺点是 Nash 平衡产生多个平衡点, 对结果约束较少。Kraus 使用 Rubinstein 的最佳平衡(Perfect Equilibrium, 简称 P.E.)理论, 建立了一种基于 P.E. 的协商方法, 需要在协商的任一阶段都产生平衡, 即在协商的任一阶段, 假设 主体 A 使用 P.E. 策略, 则主体 B 除了自己的 P.E. 策略外没有更好的策略可遵循。故若有唯一的 P.E., 并假定主体要使用该策略, 则它在协商的每一阶段都只用这一策略。可证明, 存在唯一的 P.E., 在第一阶段后就可以终止协商。Kraus 还将对时间的偏好引入这一理论中。结果表明, 时间偏好可以提高协商效率, 并仅对某一主体有利。这一理论对于主体的构造和主体之间的协作很有用处, 如果为主体提供唯一的 P.E. 策略, 并通知其他主体, 则其他主体的最佳选择也是 P.E. 策略。

该理论的缺陷是:缺乏动态性特色, 应用面窄。因为尽管存在唯一的 P.E, 但如何求出仍未得到解决。

1.2.1.4 集中式协商方法

Ephrati 使用了一种集中式的协商方法, 用一个 “master agent” 或组投票机制以达成协议。在这种方法中, 一组主体的协作与组规划进程相关。Ephrati 使用一种动态的、迭代的搜索过程, 通过一组约束, 使主体递增式地构成一个最大 “社会效用” 规划。在每一步, 各主体对于组规划的下一个联合行动投票。使用这一技术, 主体无需完整地展示其偏好, 可选状态集在投票之前产生。

该理论的缺陷是: 这一方法突破了传统的 Nash 协商理论, 但其集中式方法本身不符合多 Agent 分布的宗旨。

1.2.2 基于启发式协商

启发式方法是针对对策论的缺点而展开的, 认为计算和决策是有成本的, 并以非完全方式搜索协商空间。该方法的目的是找到 “好的解”, 而不是 “最优解”。启发式方法在计算性上与对策论方法相似, 是非形式化协商模型的可计算实现。

1.2.3 基于意图的协商

Grosz 将 Agent 的信念、愿望、意图理论应用到协商中, 该方法中不使用子规划, 而是使用意图进行协商, 减少通信量。BDI 理论认为, 导致 Agent 理性行为的既不是愿望, 也不是规划, 而是信念与愿望结合产生的意图, 处于愿望与规划之间的层次。实现意图的子规划是由该意图产生, 一个意图可能对应几个子规划, Agent 进行协商时没有必要交换各个子规划, 只要交换意图。但是 Grosz 方法中假定 Agent 是完全合作关系。

1.3 存在的问题

- 对策论在刻画偏爱和协商参与者策略时很有用, 但在应用时有以下问题: 对策论假设可以刻画 Agent 对可能结果的偏爱, 但人们很难一致地定义出 Agent 对可能结果的偏爱。只有在偏爱性质明显的地方, 对策论技术

1.2.1.4 集中式协商方法

Ephrati 使用了一种集中式的协商方法, 用一个 “master agent” 或组投票机制以达成协议。在这种方法中, 一组主体的协作与组规划进程相关。Ephrati 使用一种动态的、迭代的搜索过程, 通过一组约束, 使主体递增式地构成一个最大 “社会效用” 规划。在每一步, 各主体对于组规划的下一个联合行动投票。使用这一技术, 主体无需完整地展示其偏好, 可选状态集在投票之前产生。

该理论的缺陷是: 这一方法突破了传统的 Nash 协商理论, 但其集中式方法本身不符合多 Agent 分布的宗旨。

1.2.2 基于启发式协商

启发式方法是针对对策论的缺点而展开的, 认为计算和决策是有成本的, 并以非完全方式搜索协商空间。该方法的目的是找到 “好的解”, 而不是 “最优解”。启发式方法在计算性上与对策论方法相似, 是非形式化协商模型的可计算实现。

1.2.3 基于意图的协商

Grosz 将 Agent 的信念、愿望、意图理论应用到协商中, 该方法中不使用子规划, 而是使用意图进行协商, 减少通信量。BDI 理论认为, 导致 Agent 理性行为的既不是愿望, 也不是规划, 而是信念与愿望结合产生的意图, 处于愿望与规划之间的层次。实现意图的子规划是由该意图产生, 一个意图可能对应几个子规划, Agent 进行协商时没有必要交换各个子规划, 只要交换意图。但是 Grosz 方法中假定 Agent 是完全合作关系。

1.3 存在的问题

- 对策论在刻画偏爱和协商参与者策略时很有用, 但在应用时有以下问题: 对策论假设可以刻画 Agent 对可能结果的偏爱, 但人们很难一致地定义出 Agent 对可能结果的偏爱。只有在偏爱性质明显的地方, 对策论技术出 Agent 对可能结果的偏爱。只有在偏爱性质明显的地方, 对策论技术

1.2.1.4 集中式协商方法

Ephrati 使用了一种集中式的协商方法, 用一个 “master agent” 或组投票机制以达成协议。在这种方法中, 一组主体的协作与组规划进程相关。Ephrati 使用一种动态的、迭代的搜索过程, 通过一组约束, 使主体递增式地构成一个最大 “社会效用” 规划。在每一步, 各主体对于组规划的下一个联合行动投票。使用这一技术, 主体无需完整地展示其偏好, 可选状态集在投票之前产生。

该理论的缺陷是: 这一方法突破了传统的 Nash 协商理论, 但其集中式方法本身不符合多 Agent 分布的宗旨。

1.2.2 基于启发式协商

启发式方法是针对对策论的缺点而展开的, 认为计算和决策是有成本的, 并以非完全方式搜索协商空间。该方法的目的是找到 “好的解”, 而不是 “最优解”。启发式方法在计算性上与对策论方法相似, 是非形式化协商模型的可计算实现。

1.2.3 基于意图的协商

Grosz 将 Agent 的信念、愿望、意图理论应用到协商中, 该方法中不使用子规划, 而是使用意图进行协商, 减少通信量。BDI 理论认为, 导致 Agent 理性行为的既不是愿望, 也不是规划, 而是信念与愿望结合产生的意图, 处于愿望与规划之间的层次。实现意图的子规划是由该意图产生, 一个意图可能对应几个子规划, Agent 进行协商时没有必要交换各个子规划, 只要交换意图。但是 Grosz 方法中假定 Agent 是完全合作关系。

1.3 存在的问题

- 对策论在刻画偏爱和协商参与者策略时很有用, 但在应用时有以下问题: 对策论假设可以刻画 Agent 对可能结果的偏爱, 但人们很难一致地定义出 Agent 对可能结果的偏爱。只有在偏爱性质明显的地方, 对策论技术

才比较理想，而在更复杂的偏爱情况下，就很难使用对策论；对策论产生的是许多用于特定的相互依赖的决策的高度专用模型；对策论理论模型对可计算性有完美假设，即在可行结果范围内寻求共同接受的结果是不需要计算量的。而且还假设所有 Agent 拥有完全的信息。但在现实中这些假设是不成立的。

- 启发式协商方法是基于现实的假设，可以用于更广泛的领域。启发式协商方法也存在它的缺陷：启发式协商方法采用了近似的理性概念，也没有搜索整个可能解空间，因而模型常选择次优解；模型需要大量的评估，特别是通过仿真和经验分析来评估，因为在各种条件下不能精确预测系统和 Agent 将如何行为。

1.4 本文研究的内容

目前已有多种协商的理论、方法，主要是基于对策论和基于启发式的协商，通过分析知道还都不完善。由于协商是一件非常消耗人力、物力以及时间的事情，为了节省买卖双方为协商完成交易所花费的大量工作和时间，提高交易效率和有效降低交易成本，本文给出了一个基于 Agent 的电子交易框架，围绕着企业与用户对商品的协商问题进行研究。因为 Agent 具有自治、灵活、移动等特点，Agent 在市场中自动寻找合适的买主或卖主，采用市场交易机制进行协商，在买主或卖主中协商出一个最佳的交易，因此本文的研究具有实际的研究意义。本文在基于 Agent 的电子交易框架下，讨论 B2C 电子商务活动的部分过程，其中包括用户搜索商品信息、双方的协商、双方交易，以及在商务活动中的对商品信息的管理问题。框架中以 Agent 代替交易双方的参与人员，形成由卖方多 Agent 系统和买方多 Agent 系统组成的新营销模式。买方和卖方系统分别是一个多 Agent 系统，其中包括搜索 Agent、决策 Agent、协商 Agent、管理 Agent，交易 Agent，还有一个用户需求库和商品信息库。其中搜索 Agent、协商 Agent、交易 Agent 是移动 Agent。

协商 Agent 功能是进行协商。Agent 根据参与协商 Agent 的数目，协商可以分为双边协商和多边协商；根据协商主题的数目，可以分为单主题协商和多主题协商。

本文主要研究两个 Agent 利用协商协议就多个主题进行的协商——双边多主题协商，并给出了相应的协商模型、协商协议和协商算法以及协商流程。

本文中 Agent 的思维状态采用 BDI 模型(信念、愿望、意图)不同的 Agent 有不同的目标，也就是基于意图的协商。由于 Agent 置于特定环境下能感知环境，能代表其设计者或使用者完成一系列目标而且能灵活、自治地运行的计算实体或程序。在第三章基于 Agent 的协商模型讨论的协商问题中，本文作出如下基本假设：

- 假设 1 Agent 是自利的(追求自身效用最大化)。
- 假设 2 Agent 具有有限理性。
- 假设 3 Agent 具有不完备信息(比如不知道对方的偏好信息等)。
- 假设 4 时间因素对协商双方是宝贵的。
- 假设 5 协商双方有诚意通过协商达成一致。(协商中没有欺骗行为)。

由于该协商模型具有一般性，因此在存在广泛的适用空间。

1.5 本文的结构

本文主要论述了基于 Agent 的电子协商系统的设计与实现的研究，其中包括系统的体系结构的设计、协商模型设计、以及原型系统的实现等。

第一章：概述了本文研究的背景，分析了研究的意义和必要性，介绍目前本课题的研究现状，讨论分析了目前研究现状存在的问题，以及本文研究的主要内容和本文的组织结构。

第二章：详细的介绍了与文本有关的背景知识。这些知识包括电子商务技术知识、Agent 技术知识、移动 Agent 技术知识。这些背景知识是本文的论述基础。

第三章：提出了基于 Agent 协商模型。这一章内容既是本文的重点也是本文的难点章节。目前已有多种协商的理论、方法，主要是基于对策论和基于启发式的协商，但都不完善。本文主要研究双边多项目协商问题，是基于意图的协商。本章给出了相应的协商模型、协商协议、协商算法、协商流程等。

第四章：提出了基于 Agent 协商系统的体系结构。这一章内容为本文的重点章节。本章给出了电子商务活动的部分过程，其中包括搜索、协商、交易等活动。并以多 Agent 代替交易双方的参与人员，形成由卖方 Agent 和买方 Agent 组成的

新交易模式。

第五章：介绍了本文提出的基于 Agent 协商原型系统的实现。主要介绍了系统运行的环境、系统的 UML 图、以及实例运行部分界面图。

第六章：总结了本文的主要研究的成果，以及创新点，并指出进一步需要研究的工作。

本文的研究内容及组织结构如图 1.1 所示。第一章是本文的概述；第二章介绍了本文研究所需的背景知识；第三章提出了基于 Agent 协商系统的体系结构；第四章提出了基于 Agent 的协商模型（第三章、第四章是本文的重点和难点章节，是本文的核心部分）；第五章是基于 Agent 协商系统原型系统的实现部分，给出了相应的系统的 UML 模型图，以及实例运行部分界面；第六章是论文的总结与展望。

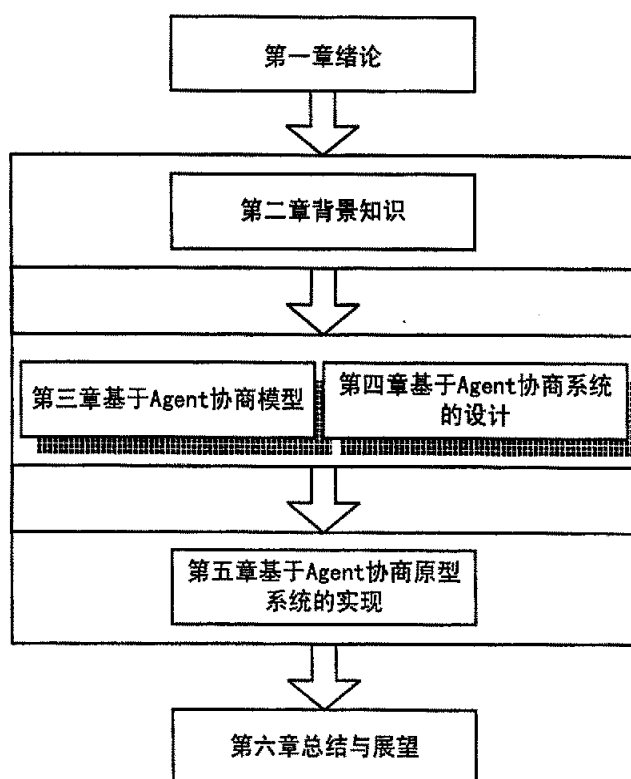


图 1.1 本文的组织结构图

1.6 小结

本章是一个综述性的章节，为本文的研究工作做了简单的介绍，可以分为五部分。第一部分:研究背景的介绍，分析了研究的意义和必要性。第二部分:介绍目前本课题的研究现状。第三部分:讨论分析了目前研究存在的问题。第四部分:本文的研究主要内容。第五部分:本文的组织结构。

第二章 背景知识

随着网络技术的不断发展,电子商务越来越显示出灵活的特性,已经成为一种不可缺少的新型商务工作模式。协商是电子商务中极其重要的部分,如果电子交易采用自动或半自动协商,就可以节省买卖双方由于协商所花费的大量时间,提高交易效率和降低交易成本。因为 Agent 具有自治、移动、灵活等特点,所以采用 Agent 技术能在节省时间和提高交易效率这方面起到辅助性的作用。

90 年代,随着计算机网络、计算机通信等技术的发展,对于 Agent 的研究成为分布式人工智能研究的一个热点。下面本章对电子商务技术、Agent 技术、移动 Agent 技术做个介绍。

2.1 电子商务技术

电子商务源于英文 ELECTRONIC COMMERCE, 简写为 EC。电子商务是利用现有的计算机硬件设备、软件和网络基础设施,在按一定的协议连接起来的电子网络环境下从事各种各样商务活动的方式。其内容包含两个方面,一是电子方式,二是商贸活动。对于电子商务概念的科学理解应包括以下几个基本方面:

- (1) 电子商务是整个贸易活动的自动化和电子化。
- (2) 电子商务是利用各种电子工具和电子技术从事各种商务活动的过程。其中,电子工具是指计算机硬件和网络基础设施;电子技术是指处理、传递、交换和获得数据的多技术集合。
- (3) 电子商务渗透到贸易活动的各处阶段,因而内容广泛,包括信息交换,售前售后服务、销售、电子支付、运输、组建拟企业、共享资源等等。
- (4) 电子商务的参与者包括消费者、销售商、供货商、企业雇员、银行或金融机构以及政府等各种机构或个人。

2.1.1 电子商务应用类型分类

- (1) 企业间的电子商务(简称为 B2B 模式)

利用网络和相关技术从事商业活动,并通过上下游企业之间的整合,增加竞争力。这就是通常所讲的 Extranet(对应于企业内部的 Intranet)。Extranet 是

企业与企业之间,为了整合资源,增加上下游企业的联系,减少信息交互成本,所构建的网络。这种模式被认为是中国特别是中国网上交易最容易实现的模式,主要原因在于一方面在进入 WTO 之后,政府放松对进出口商品的控制之后,这种模式可以避开传统的落后经销系统,使中国的企业和国外的企业连接起来,加快企业的国际化步伐,另外一方面,企业对企业的交易,在信用度恢复的情况下,支付的方式也相对容易解决。此外,还有一些便利条件包括企业间的交易数量大,金额高,物流成本小,在配送方面已经有比较成熟的渠道;在购买方式、支付手段上也已经非常成熟,同时由于是企业对企业的交易,不存在个人消费观念的障碍。

(2) 企业与消费者之间的电子商务(简称为 B2C 模式)

这是一种企业对消费者的经营模式,企业通过便利的购物渠道,向消费者提供定制的产品与服务,这就是众所周知的互联网(Internet)。现在不仅仅可以通过互联网购物,还可以根据个人不同的需求,提供定制的产品和个性化的服务。在早期多数的网站里,包括门户网站,都采用了这种零售方式。因为,对于门户网站而言,他们有丰富的客户群,而这些客户大多数又都是个人。B2C 是把注意力转换为网站盈利的主要模式,这也证明了“眼球经济”或“注意力经济”并非完全是泡沫。现阶段,B2C 的模式主要是靠网络公司在运作,不久的将来,那些传统的有较高信誉度、已经成为名牌的零售商贸企业也一定会参与 B2C 的行业中,比如北京的西单商场、翠微集团都在做这方面的努力。这种零售的优势在于它采取了直销的方式,消除了若干中间环节,减少因为中间环节而增加的消费品交易成本,使得消费者可以获得更优惠的价格。

(3) 消费者与消费者之间的电子商务(简称为 C2C 模式)

这个模式的特点是消费者与消费者讨价还价。这种模式商业运作的最大难点和问题在于个人信誉及信用监督的问题。因为这是一种投机性的交易形式,如果没有很好的监督机制,多数拍卖就会演变成光打雷不下雨,等到达成一致价格而真正需要支付时购买方却销声匿迹了。因此,支持 C2C 的电子商务网站除了建立一个拍卖交易平台,让消费者相互去讨价还价,还需要建立第三方保证机制,或者利用委托收款的方式代替保证机制(雅宝网站采用的模式),来维持交易的诚信。拍卖交易平台通常完成这样的功能:以英式拍卖、集体议价、标价求购等多

种方式完成多客户类型、多交易形式、多拍卖品类型的交易过程，平台本身不直接介入交易，只通过提供交易平台和相关服务，收取交易佣金。同时，还可以利用 AI 的技术来完成客户的竞价请求，也就是实现自动协商过程，即利用计算机来完成客户的委托请求。

(4) 企业与政府之间电子商务(简称为 B2G 模式)

企业对政府的电子商务模式。这种模式主要包括：一是政府通过网上服务，为企业创造良好的电子商务空间，如网上办公、网上报税、网上报关、网上审批等。二是政府上网采购，为企业提供大量的商机，如国家工程的竞标，如政府部门大宗公共产品的采购等。这种政府在网上发布信息，企业上网竞标的形式是一种透明度高的政府采购行为。此外，B2G 有助于政府改进工商企业管理模式，提高工商管理部门办事效率、促进政府机关勤政、廉政建设、维护消费者的权益等等。

2.1.2 电子商务系统的流程

在一个商务系统的流通体系中，一般包含如下四部分。如图 2.1 所示。

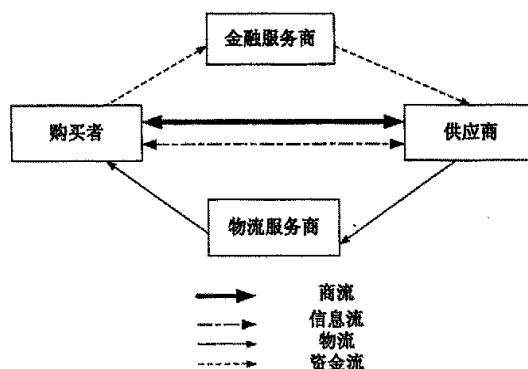


图 2.1 电子商务系统的流程

(1) 信息流：它包括商品信息的提供、促销行销、技术支持、售后服务等内容，也包括诸如询价单、报价单、付款通知单等商业贸易单证，还包括交易方的支付能力、支付信誉等。

(2) 商流：主要指商品在购、销之间进行交易和商品所有权转移的运动过程，具体指商品交易的一系列活动。

(3) 货币流：主要指货币的转移过程，包括付款、转帐等过程。

(4) 物流：是指物质实体（商品或服务）的流动过程，具体指运输、储存、配送、装卸、保管、物流信息管理等。对于大多数的商品和服务而言，物流仍然要通过物理方式实现。

2.1.3 电子商务交易的基本程序

电子商务交易应遵循的基本程序包括以下十个方面。如图 2.2 所示

(1) 商品报价请求。客户方向供货方提出商品报价请求，说明想购买的商品信息。

(2) 商品的报价。供货方向客户方回答该商品的报价，说明该商品的报价信息。

(3) 商品订购单。客户方向供货方提出商品订购单，说明初步确定购买的商品信息。

(4) 商品订购单的应答。供货方向客户方所提出的商品订购单的应答，说明有无此商品及规格型号、品种、质量等信息。

(5) 商品订购单变更请求。客户方根据应答提出是否对商品订购单有变更请求，说明最后确定购买商品信息。

(6) 商品运输说明。客户方向供货方提出商品运输说明，说明商品的运输工具、运输方式和交货地点等信息。

(7) 商品发货通知。供货方向客户方发出商品发货通知，说明商品的运输公司、发货地点、运输设备、包装等信息。

(8) 商品收货通知。客户方向供货方发回商品收货通知，报告商品收货信息。

(9) 汇款通知。交易双方收发汇款通知，买方发出汇款通知，卖方报告收款信息。

(10) 电子发票通知。供货方向客户方发送商品的电子发票，在买方收到商品，卖方收到货款并出具电子发票后，完成全部交易。

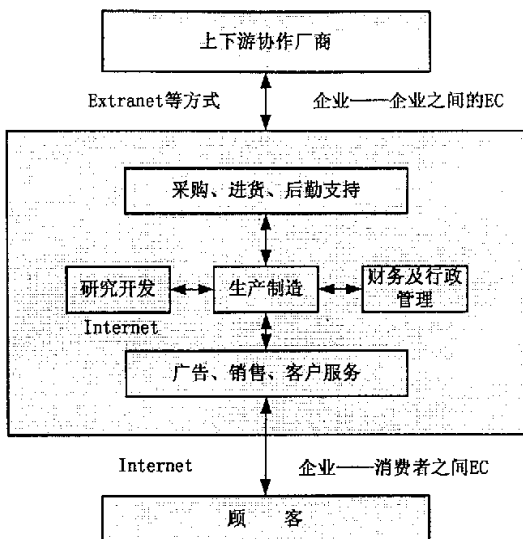


图 2.2 Internet/Intranet/Extranet 网上电子商务示意图

2.1.4 电子商务中商品直销的流程

电子商务中商品直销是指消费者和生产者，或者是需求方和供应方直接利用网络形式所开展的买卖活动。这种买卖交易的最大特点是供需直接见面，环节少、速度快、费用低。其流转程式可以用图 2.3 加以说明。

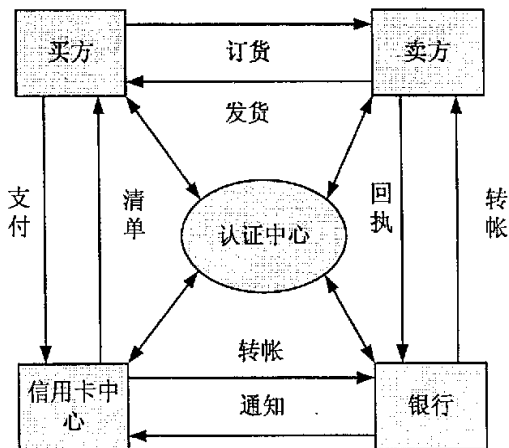


图 2.3 电子商务中商品直销流程图

网络商品直销过程可以分为以下六个步骤：

- (1) 消费者进入 Internet，查看企业和商家的主页。
- (2) 消费者通过购物对话框填写姓名、地址、商品品种、规格、数量、价格。

(3) 消费者选择支付方式，如信用卡，也可选用借记卡、电子货币或电子支票等。

(4) 企业或商家的客户服务器检查支付方服务器，确认汇款额是否认可。

(5) 企业或商家的客户服务器确认消费者付款后，通知销售部门送上门。

(6) 消费者的开户银行将支付款项传递到消费者的信用卡公司，信用卡公司负责发给消费者收费单。

2.1.5 电子商务中商品中介交易的流程

电子商务中商品中介交易是通过网络商品交易中心，即虚拟网络市场进行的商品交易。在这种交易过程中，网络商品交易中心以 Internet 网络为基础，利用先进的通信技术和计算机软件技术，将商品供应商、采购商和银行紧密地联系起来，为客户提供市场信息、商品交易、仓储配送、货款结算等全方位的服务。其流转程式可以用图 2.4 加以说明。

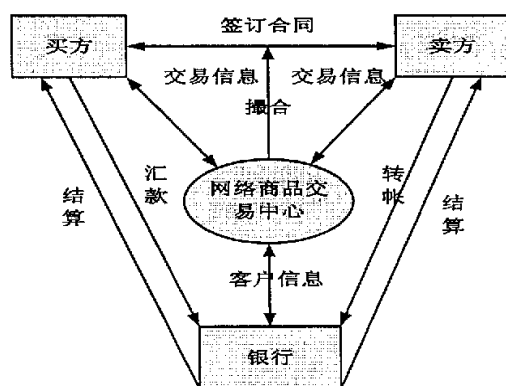


图 2.4 电子商务中商品中介交易流程图

● 网络商品中介交易的流转程式可分为以下几个步骤：

(1) 买卖双方将各自的供应和需求信息通过网络告诉给网络商品交易中心，网络商品交易中心通过信息发布服务向参与者提供大量的、详细准确的交易数据和市场信息。

(2) 买卖双方根据网络商品交易中心提供的信息，选择自己的贸易伙伴。网络商品交易中心从中撮合，促使买卖双方签订合同。

(3) 买方在网络商品交易中心指定的银行办理转账付款手续。

(4) 网络商品交易中心在各地的配送部门将卖方货物送交买方。

● 通过网络商品中介进行交易具有许多突出的优点：

(1) 网络商品中介为买卖双方展现了一个巨大的世界市场。

(2) 网络商品交易中心可以有效地解决传统交易中，“拿钱不给货”和“拿货不给钱”两大难题。

(3) 在结算方式上，网络商品交易中心一般采用统一集中的结算模式，即在指定的商业银行开设统一的结算账户，对结算资金实行统一管理，有效地避免了多形式、多层次的资金截留、占用和挪用，提高了资金风险防范能力。

2.2 Agent 技术

进入 20 世纪 90 年代以来，在各种计算机文献和众多公司的技术发展规划中，Agent 正日渐成为使用频率最高的词汇之一。

Agent 技术的诞生和发展是人工智能技术 (AI) 和网络技术发展的必然结果。从 20 世纪 60 年代起，传统的 AI 技术开始致力于对知识表达、推理、机器学习等技术的研究，其主要成果是专家系统。专家系统把专业领域知识与推理有机的组合在一起，为应用程序的智能化提供了一个低级而实用的解决办法。作为人工智能的一个分支，AI 计划理论的研究成果使应用程序有了初步的面向目标和特征，即应用程序具有了某种意义上的主动性；而人工智能的另一个分支—决策理论和方法则使应用程序具有了自主判断和选择的行為的能力。人工智能围绕着知识所进行的广泛研究和应用正逐步形成一门新的学科，这就是知识工程，它涉及的知识的获取、存储和管理等许多课题。所有这些技术的发展加快了应用程序智能化的进程。

随着网络技术的发展，多个应用程序间相互作用的模式正从单一的集成式系统向分布式系统演化。一个在物理上和地理上分布的应用程序之间通信与合作的网络地层基础结构正逐渐建立起来。分布式对象技术（如 CORBA 或 DCOM 技术）则进一步使分布且异构的应用程序之间能以一种共同的方式提供和获得服务，实现了在分布式状态下的“软”集成。

智能化和网络化的发展促成了 Agent 技术的发展，Agent 技术正是为解决复杂、动态、分布式智能应用而提供的一种新的计算手段。

2.2.1 Agent 定义

目前并不存在一个被普遍接受的 Agent 的定义，事实上，对这个问题尚有争论，并存在不同的看法。尽管把自治性作为 Agent 的核心概念已经达成了普遍的共识，但除此之外很少有一致的看法。

Wooldridge 和 Jennings 在总结了前人在 Agent 领域的一些工作后认为，可以从狭义和广义两个方面去理解 Agent 的特性。

(1) Agent 弱概念

这是从广义的角度来规定 Agent 的特性。几乎所有被称为 Agent 的软件或硬件系统都具有以下的特性：

- 自治性 (Autonomy)：Agent 运行时不直接由人或者其它东西控制，它对它们自己的行为和内部状态有一定的控制权。
- 社会能力 (Social Ability) 或称可通信性：Agent 能够通过某种 Agent 通信语言 (agent communication language) 与其它 Agent 进行信息交换。
- 反应能力 (Reactivity)：即对环境的感知和影响。无论 Agent 生存在现实的世界中（如机器人、Internet 上的通讯 Agent、用户界面 Agent 等）还是虚拟的世界中（如虚拟商场中的 Agent、交互式游戏中的 Agent 等），它们都应该可以感知它们所处的环境，并通过行为改变环境。
- 自发行行为 (Pro-activeness)：传统的应用程序是被动地由用户来运行的，而且机械地完成用户的指令；而 Agent 的行为应该是主动的，或者说自发的。Agent 感知周围环境的变化，并作出基于目标的行为 (goal-directed behaviour)。

在这种定义下，最简单的 Agent 就是具有上述特性的一个计算机进程，这个进程可以简单到只是个具有某种智能的子程序，能够与别的 Agent 交换消息。Agent 弱概念使 Agent 不仅仅只应用于人工智能领域，而且广泛地应用在诸如人机界面、通信、并行工程、软件工程、搜索引擎等。因此很多计算机软件都可以纳入 Agent 的范畴里来，例如处理 Internet 事务，帮助用户处理 E-mail 的助理 Agent 等。

Marvin Minsky 从多 Agent 系统的社会智能的角度给出 Agent 的一个定义：“这些进程我们称之为 Agent，每个 Agent 本身只会做一些简单的事情，

但当我们用特定的方法将这些 Agent 组成一个 Agent 群 (society), 就产生了真正的智能。” Minsky 的定义显然也是一种广义的定义。

(2) Agent 强概念

对某些研究者, 尤其是人工智能的研究者来说, Agent 除了应具有上面这些特性以外, 还应该具有某些通常人类具有的特性。例如知识、信念、意图、承诺等心智状态。有的学者甚至提出了有情感的 Agent。

当前对强概念 Agent 的研究主要集中在理论方面。例如, Shoham 提出的面向 Agent 编程 (AOP Agent-Oriented Programming) 使用的就是强概念的 Agent 定义: “一个 Agent 是这样一个实体, 它的状态可以看作是由信念 (belief)、能力 (capability)、选择 (choice)、承诺 (commitment) 等心智构件 (mental component) 组成。”

(3) Agent 的其它属性

- 长寿性 (Longevity): 传统应用程序在用户需要时激活, 不需要时或者运算结束后停止。Agent 与之不同, 它应该至少在“相当长”的时间内连续地运行。
- 移动性 (Mobility): Agent 可以从一个地方移动到另一个地方而保持其内部状态不变。Agent 可以携带数据和能够在远处执行智能指令。
- 推理能力 (Reasoning): Agent 可以根据其当前的知识和经验, 以理性的、可再生的方式推理或推测。
- 规划能力 (Planning): 根据目标、环境等的要求, Agent 应该至少对自己的短期行为作出规划。虽然程序设计人员可以提供一些常见情况的处理策略, 但这些策略不可能覆盖 Agent 将遇到的所有情况。所以, Agent 应该有生成规划的能力。
- 学习和适应能力 (Learning and Adaptability): Agent 可以根据过去的经验积累知识, 并且修改其行为以适应新的环境。另外, 有些学者还提出 Agent 应该具有自适应性、个性等特性。
- 诚实 (Veracity): 假定 Agent 不会故意提供错误信息。
- 善意 (Benevolence): 假定在 Agent 之间不会存在相互冲突的目标, 且 Agent 总是尽力去帮助其它 Agent。

- 理性 (Rationality): 假定 Agent 总是尽力去实现自己的目标。

2.2.2 理性主体 (BDI 主体)

1987 年, Bratman 从哲学上对行为意图 (intention) 的研究对人工智能产生了广泛的影响。他认为只有保持信念 (belief)、愿望 (desire) 和意图 (intention) 的理性平衡, 才能有效地解决问题。

Belief——信念, 主体对环境的基本看法。

Desire——愿望, 主体想要实现的状态, 即目标。

Intention——意图, 目标的子集。

BDI 主体模型可以通过下列要素描述:

- 一组关于世界的信念;
- 主体当前打算达到的一组目标;
- 一个规划库, 描述怎样达到目标和怎样改变信念;
- 一个意图结构, 描述主体当前怎样达到它的目标和改变信念。

Rao 和 Georgeff 提出了一个简单的 BDI 解释器 (Rao 和 Georgeff 1992)

BDI-Interpreter

initialize-state();

do

options := option-generator(event-queue, B, G, I);

selected-options := deliberate(options, B, G, I);

update-intentions(selected-options, I);

execute(I);

get-new-external-events();

drop-successful-attitudes(B, G, I);

drop-impossible-attitudes(B, G, I);

until quit

2.2.3 Agent 结构

2.2.3.1 基本结构

Agent 可以看成是一个黑箱，通过传感器感知环境，通过效应器作用环境。主体的传感器有眼、耳、鼻以及其它器官，而手、腿、嘴、身体可以看成效应器。Agent 通过字符串编码作为感知和作用。如图 2.5 所示。

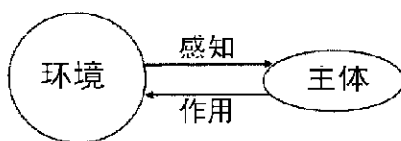


图 2.5 黑箱软件主体

Agent 不仅要与环境交互作用，更主要是处理和解释接受的信息，达到自己的目的。图 2.6 给出了 Agent 的工作过程。

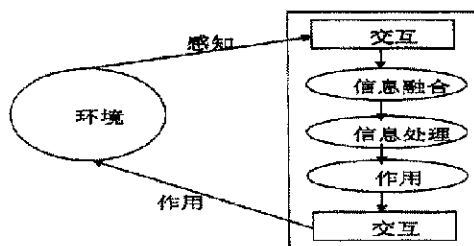


图 2.6 Agent 的工作过程

一个 Agent 骨架程序如下：

```
function Skeleton-Agent(percept) return action
static: memory /* 主体的世界记忆 */
memory ← Update-Memory(memory, percept)
action ← Choose-Best-Action(memory)
memory ← Update-Memory(memory, action)
```

```
return action
```

2.2.3.2 慎思 Agent 结构

慎思 Agent (deliberative agent), 又称认知 Agent (cognitive agent), 是一个显式的符号模型, 图 2.7 为慎思 Agent 结构。其特点是包括环境和智能行为的逻辑推理能力, 是一个基于知识的系统。慎思 Agent 具有内部状态的主动软件, 具有知识表示、环境表示、问题求解、具体通信协议等。该体系结构在 DAI 中占主导地位。

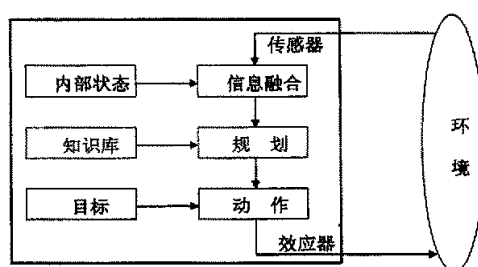


图 2.7 慎思 Agent 结构

慎思 Agent 程序如下:

```
function Cognitive-Agent(percept) returns action
    static: environment, /* 描述当前世界环境 */
    kb, /* 知识库 */
    environment ← Update-World-Model(environment, percept)
    state ← Update-Mental-State(environment, state)
    action ← Decision-Making(state, kb)
    environment ← Update-World-Model(environment, action)
    return action
```

2.2.3.3 反应 Agent 结构

反应 Agent (reactive agent) 是不包含用符号表示的模型, 图 2.8 为反应 Agent 结构图。其特点是反应 Agent 中包含了感知内外状态变化的感知器, 一组

对相关事件做出反应的过程，一个依据感知器信息激活的控制系统。该系统在分布式系统中占主导地位。

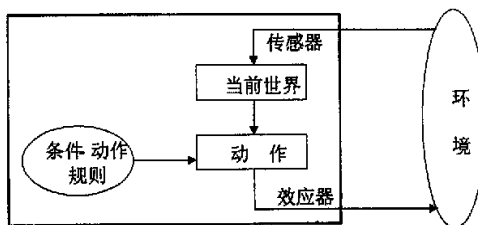


图 2.8 反应 Agent 结构图

反映 Agent 程序如下：

```
function Reactive-Agent(percept) returns action
    static: rules, /* 一组条件-动作规则 */
    state ← Interpret-Input(percept)
    rule ← Rule-Match(state, rules)
    action ← Rule-Action[rule]
    return action
```

2.2.3.4 混合式 Agent 结构

混合式 Agent 结构模型的特点是在一个 Agent 中包含两个或多个子系统，一个是慎思子系统，一个是反应子系统。而且是分层组织的，前者建立在后者之上。混合式 Agent 结构模型克服了前两种模型功能不够全面、结构不够灵活的缺点，给出了具有实时反应目标指导下自主行动及合作等综合能力的 Agent 模型。

2.2.4 通信方式

Agent 必须彼此能用多 Agent 进行分布式问题求解，集成在一个系统中的 Agent 必须彼此能通信和协作。通信是协作的基础。图 2.9 为多 Agent 中的通信和协议。通信方法可以分成黑板系统和消息\对话系统。

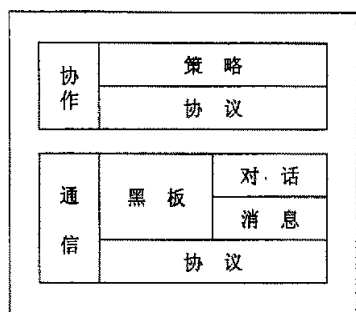


图 2.9 多 Agent 中的通信和协议

2.2.4.1 黑板系统

黑板系统是传统的人工智能系统和专家系统的议事日程的扩充，通过使用合适的结构支持分布式问题求解。在多 Agent 系统中，黑板是指一个可供 Agent 发布信息、公布处理结果和获取有用信息的共享区域，Agent 之间可以交换信息、数据和知识。开始一个 Agent 在黑板写入信息项，然后可为系统中其它 Agent 使用。Agent 可以在任何时候访问黑板，看看有没有新的信息到来。它并不需要阅读所有信息，可以采用过滤器抽取当前工作所需要的信息。Agent 必须在访问授权中心站点登录，在黑板系统中 Agent 之间不发生直接通信，每个 Agent 独立地完成它们答应求解的子问题。

黑板可以用在任务共享和结果共享系统中，它提供了一种比较灵活、迅速和高效的通信方式，但由于 Agent 的个性不同，知识库及计算处理能力有异，要求它提供灵活的公用信息表示机制，另外它的事件驱动性还要求集中控制机构，这使得其实现和运行代价较大。

2.2.4.2 消息传递(Message Passing)模式

这是目前软件系统(尤其是面向对象系统)中常用的传统方法，例如：(远程)过程调用、(远程)函数调用等。Agent 使用一组事先约定的格式和规则通过消息形式相互传递计算请求和处理结果，当特定状态出现或预先定义的事件发生(比如 Agent 收到其它 Agent 发来的消息，或者 Agent 感知到环境的某些变化)时，这些规则就会被激活，Agent 于是采取相应的行动。该模式通常都以某一方 Agent

为任务中心，其它 Agent 则向该中心请求服务。其优点是高效、可支持分布式计算和移动计算，但缺点却是不灵活、不便于扩展，以及不利于 Agent 合作等。图 2.10 为消息传送的原理图。

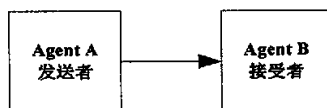


图 2.10 消息传送原理

2.2.5 通信语言 ACL

目前主要有两种 ACL，一种是 ARPA 的 KSE(Knowledge Sharing Effort)联盟提出的 KQML 语言，另一种是欧洲的 FIPA 协会提出的语言标准，即 FIPA—ACL。

2.2.5.1 KQML 语言

KQML(Knowledge Query and Manipulation Language)是一种用于信息和知识交换的语言和协议。KQML 既是一种 agent 之间的消息的表示格式，也是一种消息处理协议，它支持 agent 之间的运行时(Run-time)知识共享。KQML 可作为应用程序与智能系统之间或智能系统相互之间进行交流的一种语言，以知识共享为基础。支持协同问题处理。KQML 的核心为一组可扩展的行为原语(Perfonnative)。它们定义了可作用于 agent 彼此的知识库和目标库的各种许可的操作。基于 KQML 行为原语，可建立 Agent 之间交流的更高层模型，例如协商机制和合约网(Contract Nets)。此外，KQML 还提供了一种进行知识共享的体系结构，通过一类被称为交流设施(Facilitator)的 agent，来协调其他 agent 之间的交互。

KQML 的协议：在 KQML 中有多种进程间的信息交换协议。最简单的情况就是一个进程(客户端)向另一个进程(服务器)发出一个查询消息，然后等待一个回答，如图 2.11(a)。这种情况下，协议要求服务器端所回复的消息确实包含了必需的信息。另一种常见的情况是当服务器端的答复不能在一次传送中完成，而客

户端每次只有一个向负责回复的组件(服务器端)请求的句柄,如图 2.11(b)。例如,一个简单的客户端向关系数据库或推理机发出一个查询消息,关系数据库或推理机可能会作出一连串的回答。除了上述两种情况外,实时系统又是另外一种情况。在客户端向服务器预定输出以后,不确定数量的回复按照不规则的时间间隔到达,如图 2.11(C),在这种情形中,客户端不知每一个回复消息何时到达,当它们到达时客户端可能正忙于做其他任务。

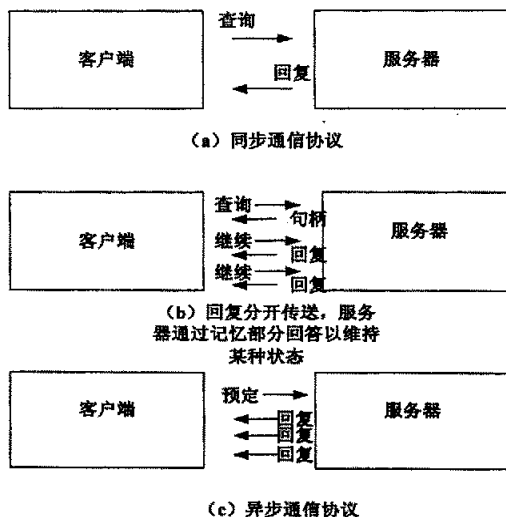


图 2.11 KQML 协议

2.2.5.2 KQML 语言结构

- KQML 是一种层次结构型语言,可分为三层:内容层、通信层和消息层。如图 2.12 所示。

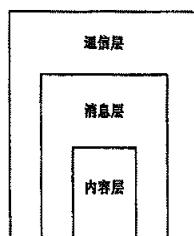


图 2.12 KQML 结构

- 内容层(Content Layer): 是消息所包含的真正内容, 用程序自己的表示语言来表示。KQML 可以采用任何形式的表示语言, 以 ASCII 字符串或二进制形式都可以。值得指出的是, 所有 KQML 语言的具体实现都不关心消息中的内容部分的具体含义。
- 通信层(Communication Layer): 通信层描述了与通信双方有关的一组属性参数, 例如发送方和接受方的身份, 与此次通信相关的惟一的标识等。
- 消息层(Message Layer): 消息层构成了 KQML 语言的核心。它确定了可以与使用 KQML 语言的 agent 进行交流的类型。消息层的基本功能是确定传送消息所使用的协议, 并由发送方提供一个与内涵相关的行为原语, 用于指明消息中的内涵为确认、询问、命令或是其他已知的原语类型。因为内涵对于 KQML 是透明的, 所以在消息层也包含对内涵进行描述的可选参数, 例如所用的语言、采用的 ontology 等。这些属性参数可以让 KQML 语言的具体实现在内涵是透明的情况下, 仍能够正确地分析和传送消息。

KQML 的语法非常简单, 是基于平衡的括号表。表的开始为行为原语的名称, 其余部分为一组以“: 关键字值”形式出现的参数表。

KQML 中定义了一组含义明确的、预留的行为原语。这些预留的行为原语并非 KQML 具体实现中必须实现的最小子集, 它可以根据需要选择实现或添加新的原语, 但是通常要求选择实现的预留原语应满足 KQML 标准的要求。

习惯上, 一条 KQML 行为原语也称为一条消息, 一条典型的 KQML 消息如下:

(ask—about

```

: sender Chongqing
: content(Type of goods?x
: receiver Shanghai
: reply—with id1
: language KIFX)
: ontology NYSE—TICKS)
```

其中, ask—about 为 KQML 的一条用于询问的预留的行为原语名字, sender、content、receiver、reply—with、language、ontology 为 KQML 预留的行为原语的参数。参数 content 的值是一个表达式, 它必须符合参数 language 所指定

的语言的语法, 其中的常量必须在参数 ontology 指定的概念关系中有定义。由于 agent 之间多采用异步方式通信, 所以利用参数 in—reply—to 和参数 reply—with 来匹配发出的询问和收到的回答。Ontology(实体抽象)是一个重要的概念, 一个 ontology 是指存在于交互的 agent 之间的或 agent 中的概念与概念之间的关系的一种描述。在 KQML 中, ontology 参数指的是 content 参数使用的实体集(如术语定义集)的名称。

常用的预留的行为原语通常可以被分为以下几类:

- 基本询问原语(如 evaluate, ask—if, ask—in, ask—one, ask—all, ask—about)
- 简单询问回答原语(如 reply, sorry)
- 多重询问回答原语(如 stream—in, stream—all)
- 通用信息原语(如 tell, achieve, cancel, untell, unachieve)
- 发生器原语(如 standby, ready, next, rest. dis. card, generator)
- 能力定义原语(如 advertise, subscribe, monitor, import, export)
- 网络原语(如 register, unregister, forward, broad-cast, route)

现在, 还没有为基本的 KQML 行为原语或与之相关的协议定义出正式的语义。一个发展中的语义模型假定一个使用 KQML 的 Agent 有一个虚拟知识库, 这知识库由两部分组成: 信息库(信念)和目标库(意图)。基本行为原语按照它们对这两个库的影响而定义。比如, TELL(S) 表示发送 agent 向接受 agent 声明句子 s 在它的虚拟信念库中, ACHIEVE(S) 表示发送 agent 向接受 agent 请求将 s 添加到它自己的目标库中。当一个 Agent 接受到一个 KQML 消息时, 控制它产生适当响应的协议也被定义出了。

2.3 移动 Agent 技术

随着 Internet 应用的逐步深入, 特别是信息搜索、分布式计算以及电子商务的蓬勃发展, 人们越来越希望在整个 Internet 范围内获得最佳的服务, 渴望将整个网络虚拟成为一个整体, 使软件 Agent 能够在整个网络中自由移动, 移动 Agent 的概念随即孕育而生。

2.3.1 移动 Agent 的概念

20 世纪 90 年代初, General Magic 公司在推出其商业系统 Telescript 时第一次提出了移动 Agent 的概念, 即一个能在异构网络环境中自主地从一台主机迁移到另一台主机, 并可与其它 Agent 或资源交互的软件实体。移动 Agent 是一类特殊的软件 Agent, 它除了具有软件 Agent 的基本特性——自治性、响应性、主动性和推理性外, 还具有移动性, 即它可以在网络上从一台主机自主地移动到另一台主机, 代表用户完成指定的任务。由于移动 Agent 可以在异构的软、硬件网络环境中自由移动, 因此这种新的计算模式能有效地降低分布式计算中的网络负载、提高通信效率、动态适应变化了的网络环境, 并具有很好的安全性和容错能力。

移动 Agent 可以看成是软件 Agent 技术与分布式计算技术相结合的产物, 它与传统网络计算模式有着本质上的区别。移动 Agent 不同于远程过程调用(RPC), 这是因为移动 Agent 能够不断地从网络中的一个节点移动到另一个节点, 而且这种移动是可以根据自身需要进行选择的。移动 Agent 也不同于一般的进程迁移, 因为一般来说进程迁移系统不允许进程自己选择什么时候迁移以及迁移到哪里, 而移动 Agent 却可以在任意时刻进行移动, 并且可以移动到它想去的任何地方。移动 Agent 更不同于 Java 语言中的 Applet, 因为 Applet 只能从服务器向客户机做单方向的移动, 而移动 Agent 却可以在客户机和服务器之间进行双向移动。

2.3.2 移动 Agent 的系统结构

虽然目前不同移动 Agent 系统的体系结构各不相同, 但几乎所有的移动 Agent 系统都包含移动 Agent (简称 MA) 和移动 Agent 服务设施 (简称 MAE) 两个部分, 如图 2.13 所示。MAE 负责为 MA 建立安全、正确的运行环境, 为 MA 提供最基本的服务 (包括创建、传输、执行), 实施针对具体 MA 的约束机制、容错策略、安全控制和通信机制等。MA 的移动性和问题求解能力很大程度上取取决于 MAE 所提供的服务, 一般来讲, MAE 至少应包括以下基本服务:

- 事务服务 实现移动 Agent 的创建、移动、持久化和执行环境分配。
- 事件服务 包含 Agent 传输协议和 Agent 通信协议, 实现移动 Agent 间的事

件传递。

- 目录服务 提供移动 Agent 的定位信息，形成路由选择。
- 安全服务 提供安全的执行环境。
- 应用服务 提供面向特定任务的服务接口。

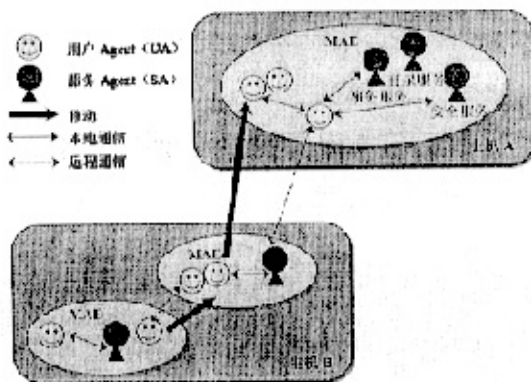


图 2.13 移动 Agent 系统

通常情况下，一个 MAE 只位于网络中的一台主机上，但如果主机间是以高速网络进行互联的话，一个 MAE 也可以跨越多台主机而不影响整个系统的运行效率。MAE 利用 Agent 传输协议 (Agent Transfer Protocol, ATP) 实现 MA 在主机间的移动，并为其分配执行环境和服务接口。MA 在 MAE 中执行，通过 Agent 通信语言 (Agent Communication Language, ACL) 相互通信并访问 MAE 提供的各种服务。在移动 Agent 系统的体系结构中，MA 可以细分为用户 Agent (User Agent, UA) 和服务 Agent (Server Agent, SA)。UA 可以从一个 MAE 移动到另一个 MAE，它在 MAE 中执行，并通过 ACL 与其它 MA 通信或访问 MAE 提供的服务。UA 的主要作用是完成用户委托的任务，它需要实现移动语义、安全控制、与外界的通信等功能。SA 不具有移动能力，其主要功能是向本地的 MA 或来访的 MA 提供服务，一个 MAE 上通常驻有多个 SA，分别提供不同的服务。由于 SA 是能不移动的，并且只能由它所在 MAE 的管理员启动和管理，这就保证了 SA 不会是“恶意的”。UA 不能直接访问系统资源，只能通过 SA 提供的接口访问受控的资源，从而避免恶意 Agent 对主机的攻击，这是移动 Agent 系统经常采用的安全策略。

2.3.3 移动 Agent 传输协议

Agent 传输协议定义了移动 Agent 传输的语法和语义, 具体实现了移动 Agent 在服务设施间的移动机制。IBM 提出的 ATP 框架结构定义了一组原语性的接口和基础消息集, 可以看作是一个 Agent 传输协议的最小实现, 其基本操作如图 2.14 所示。

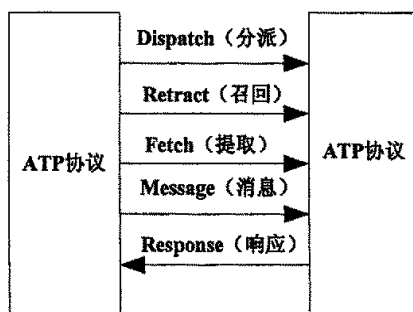


图 2.14 Agent 的传输协议

2.3.4 移动 Agent 基本特征

移动 Agent 是一个全新的概念, 虽然目前还没有统一的定义, 但它至少具有如下一些基本特征:

- 身份唯一性 移动 Agent 必须具有特定的身份, 能够代表用户的意愿。
- 移动自主性 移动 Agent 必须可以自主地从一个节点移动到另一个节点, 这是移动 Agent 最基本的特征, 也是它区别与其他 Agent 的标志。
- 运行连续性 移动 Agent 必须能够在不同的地址空间中连续运行, 即保持运行的连续性。具体说来就是当移动 Agent 转移到另一节点上运行时, 其状态必须是在上一节点挂起时那一刻的状态。

2.3.5 移动 Agent 技术与电子商务系统结合的优势

(1) 提高商务信息处理的效率。一般说来, 电子商务主要涉及商品信息的发布和获取、交易的协商和谈判、合同的签订和执行、货币的支付与确认、私有性和公平性以及安全性的保证等方面。移动 Agent 技术可以在商品信息的发布和获取、交易的协商和谈判等方面节省大量的时间和费用。

(2)快速便捷，易于使用。按照旅行计划自动进行的 Agent 网络旅行将取代费时费力的人工查找，极大地方便了客户的使用，而 Agent 系统支撑及旅行中动态地对环境的自适应(旅行计划的自我调整、商务信息处理策略的改变等)使信息系统能直接面向最终用户。

(3)适应灵活的市场开拓机制。与静态的 Web 信息相比，可移动、可协作的 Agent 可随时根据商家的信息需要携带商品及服务信息实现“送货上门，主动服务”，变被动为主动。

(4)高效、安全的商务信息处理过程。只有永不疲惫的、能充分利用本机交互特性做出第一时间快速反应的移动 Agent 才能适应“时间就是金钱、商机稍纵即逝”的市场法则，而安全的 Agent 所具有的坚定性和兼容性能提供较大程度的商务信息处理的安全性，加之数字签名和认证、信息加密和解密等。

因此，我们有理由认为，移动 Agent 技术所具有的移动性、协同性、安全性等优点将较好地弥补 www 技术的不足，为电子商务的发展提供更好、更大的发展空间。

2.4 小结

本章是背景知识介绍，可以分为三部分。第一部分:电子商务技术的知识，介绍了电子商务的分类、流程、基本交易程序以及直销和中介交易的流程。第二部分:Agent 技术的知识，介绍了 Agent 的定义、Agent 的基本结构、Agent 的通信方式以及 Agent 通信语言。第三部分:移动 Agent 技术的知识，讨论了移动 Agent 系统的结构、传输协议、以及与电子商务系统结合的优势。

第三章 基于 Agent 的协商模型

3.1 引言

多 Agent 系统(MAS)是分布式人工智能的一个重要研究领域, MAS 中每个 Agent 具有自治性, Agent 可以控制自己的内部意识态度(信念、愿望、意图等), 也可以控制行动并作用于周围环境, 达到自己的目标。协商是 MAS 实现协同、协作、冲突消解和矛盾处理的关键环节, 是建立在 Agent 通信语言之上的一种 Agent 交互机制, 通过协商对某些问题达成一致意见。在属于电子商务领域的消费者购买行为(CBB)模式中, 协商是其中的一个重要阶段, 通过采用市场机制手段, 将有限的资源进行公平分配。目前主要是一些采用简单的协商协议实现的系统, 如网上拍卖系统 AuctionBot。多 Agent 系统中的协商、协作问题受到越来越多学者的重视, 学者们从经济学、社会心理学、DAI 等不同的领域进行研究。目前已有多种协商的理论、方法, 主要是基于对策论和基于启发式的协商, 但都不完善。

本文主要研究双边多项目协商问题, 给出了相应的协商模型、协商协议和协商算法。

3.2 协商模型

一个好的协商模型应具有如下属性: 能够有效地描述协商问题; 能够描述协商的动态特性; 协商过程需要的计算资源是有限的。基于以上设计目标, 本文给出双边多项目协商模型。

本文中 Agent 的思维状态采用 BDI 模型(信念、愿望、意图)不同的 Agent 有不同的目标, 也就是基于意图的协商。由于 Agent 置于特定环境下能感知环境, 能代表其设计者或使用者完成一系列目标而且能灵活、自治地运行的计算实体或程序。本文的 Agent 协商是在如下基本假设的前提下完成的, 即:

- 假设 1 Agent 是自利的(追求自身效用最大化)。
- 假设 2 Agent 具有有限理性。
- 假设 3 Agent 具有不完备信息(比如不知道对方的偏好信息等)。
- 假设 4 时间因素对协商双方是宝贵的。

- 假设 5 协商双方有诚意通过协商达成一致(协商中没有欺骗行为)。

协商模型包括协商函数和决策函数, 通过该协商模型我们可以选择协商策略, 进而完成协商功能。

3.2.1 协商函数

下面我们就对协商函数进行描述, 协商函数定义如下:

$M = \langle Ag, Q, X, W, W', T, R, A \rangle$, 其中:

- Ag ——协商双方 Agent, $Ag_i \in Agent$, 其中 Ag_i 可以表示协商中的买方或卖方。 $(1 \leq i \leq n)$. 这里我们只做最简单的情况, 即 $i \in (a, b)$, 其中 Ag_a 表示买方 Agent, Ag_b 表示卖方 Agent。
- Q ——协商项目集, $Q = \{Q_1, Q_2 \dots Q_J\}$, Q_j 是第 j 个协商项目, 项目数 $J \geq 2$ 。
例如可以对商品的价格、商品的数量、商品服务时间、商品服务的提交地点等协商项目进行协商。
- X ——总协商策略空间, $X = (x_1^i \dots x_j^i \dots x_J^i)$, 其中 x_j^i 是 Ag_i 对项目 Q_j 的一个提议 $(1 \leq i \leq N, 1 \leq j \leq J)$ 。若协商项目 Q_j 有 n_j 种则可能的提议, 总协商策略 $N = n_1 * n_2 * \dots * n_J$
- W ——协商项目的权重集, $W = \{W_1^i, W_2^i, \dots, W_J^i\}$, W_j^i 表示 Ag_i 给项目 Q_j 目的权重 $(1 \leq j \leq J)$, 比如 $W_1^i > W_2^i$, 则表示对协商项目 Q_1, Q_2 Ag_i 更偏好 Q_1 。
- W' ——协商项目之间的权重值。则表示协商项目之间的权重偏好值。例如, 当商品的数量达到一定范围值后, 就可以对价格的提出更满意的要求, 要求更低的价格。
- T ——协商时间(期限)。即协商总要有个限制的期限, 不能无休止的等待下去。
- R ——所有可能协商回合集, $r^1, r^2 \dots r^k \in R$ 。即协商过程中的协商次数。
- A ——所有可能行动集。即协商过程可能产生的动作。 $A = \{call, request, propose, agree, reject\}$ 。其含义是, call: 发出协商请求; request:

响应协商请求: propose: 发出提议; agree: 同意提议; reject: 拒绝提议}

通过这个协商函数,我们就可以把双边多项目协商问题描述出来,那么具体如何协商则要根据决策函数来定。

3.2.2 决策函数

下面,本文就对决策函数进行说明。决策函数主要是决定Agent选择合适的协商策略,从而实现自动协商的过程。

设 Ag_i 对项目 Q_j 的权重 W_j^i , 且 $\sum_{1 \leq j \leq J} W_j^i = 1$,

函数 $V_j^i: D_j^i \rightarrow [0,1]$, V_j^i 表示 Ag_i 对 x_j 的评价函数, 其中

$D_j^i = [\min_j^i, \max_j^i]$, ($1 \leq j \leq J$) 表示 Ag_i 对项目 Q_j 的 n_j 种可能的提议。

定义函数 $R_s = \sum_{1 \leq j \leq J} W_j^i V_j^i(x_j)$ 作为选择协商策略的决策函数。

评价函数既可以是线形的也可以是非线形的, 简单起见, 定义为线形的单调减函数或单调增函数:

其中单调减函数

$$V_j^i(x_j) = \frac{\max_j^i - x_j^i}{\max_j^i - \min_j^i};$$

其中单调增函数

$$V_j^i(x_j) = \frac{x_j^i - \min_j^i}{\max_j^i - \min_j^i};$$

但是, 由于某个协商项目取值过大或过小会对别的协商项目产生影响, 故通过 W' 来表示其产生的影响。例如, 当商品的数量多达一定值时, 会对商品的价格看的很重, 要求更低的价格。

为此我们通过 $R'_s = W'_R V_R + (1 - W'_R) R_s$ 做为最终的决策函数。由 R'_s 决定选择协商结果。其中 V_R 就是看重的协商项目。

例如, 有买方 Ag_a 和卖方 Ag_b 就某商品的价格、交货时间进行协商

即 $Q = \{\text{价格}, \text{交货时间}\}$ ，设卖方 Ag_b 设定的权重 $w_{\text{价格}}^b = 0.7$,

$$w_{\text{交货时间}}^b = 0.3, D_{\text{价格}}^b = [10, 20], D_{\text{交货时间}}^b = [1, 5],$$

对卖方 Ag_b 而言，某商品的价格越高，其评价也越高，故用增函数

$$V_j^i(x_j) = \frac{x_j^i - \min_j^i}{\max_j^i - \min_j^i};$$

交货时间越快，其评价也越高，故用减函数

$$V_j^i(x_j) = \frac{\max_j^i - x_j^i}{\max_j^i - \min_j^i};$$

$$V_{\text{价格}}^b(x_{\text{价格}}^b) = \frac{x_{\text{价格}}^b - \min_{\text{价格}}^b}{\max_{\text{价格}}^b - \min_{\text{价格}}^b},$$

$$V_{\text{交货时间}}^b(x_{\text{交货时间}}^b) = \frac{\max_{\text{交货时间}}^b - x_{\text{交货时间}}^b}{\max_{\text{交货时间}}^b - \min_{\text{交货时间}}^b}$$

Ag_b 在时间 t_1, t_2 分别提出提议 $x_1^b = (11, 5)$, $x_2^b = (15, 2)$,

Ag_b 评价 x_1^b 和 x_2^b : 对 $V_{\text{价格}}^b = 0.1, V_{\text{交货时间}}^b = 0$,

$$R_s(x_1^b) = 0.7 * 0.1 + 0.3 * 0 = 0.07,$$

同理可以计算

$R_s(x_2^b) = 0.575$. 在这里由于交货时间与价格之间影响不大，故 $w_R' = 0$,

$$R_s' = w_R' V_R + (1 - w_R') R_s,$$

$R_s' = R_s$ 。即决策函数就是最终的决策函数。根据假设1，即 Ag_b 选择 x_2^b 。

也就是 Ag_b 通过选择 x_2^b 与其他的 Ag 进行协商。

那么如何决定是否进一步协商呢？这就要用到评价函数。关于评价函数这在上文已经交代了。这里只是讨论评价函数的增减性的问题，从而判断Agent是否进一步协商。

记 $\text{Res}V_j^i$ 表示 Ag_i 对项目 Q_i 的底价。

$$\text{Res} = \begin{cases} \max_j^i, & \text{若 } V_j^i \text{ 单调递减} \\ \min_j^i, & \text{若 } V_j^i \text{ 单调递增} \end{cases},$$

即 V_j^i 是单调递减的函数时, Res 有最大值。例如当买方对商品的价格的评价函数是递减的。价格越低, 评价函数值就越大, 但有最大值。也就是说对商品的最高接受价格, 超过此值不能在接受, 因而有最大值。

同理, 即当 V_j^i 是单调递增的函数时, Res 有最小值。例如当卖方对商品的价格的评价函数是递增的。价格越高, 评价函数值就越大, 但有最小值。也就是说对商品的最低接受价格, 超过此值不能在接受, 因而有最小值。

Ag_i 对协商的项目的预测底价 $rp^i = \sum_{1 \leq j \leq J} W_j^i Res V_j^i$ ($1 \leq i \leq n$), 即是 Ag_i 能接受的底价。对买方而言是能接受最高价位, 对卖方而言是能接受最低价位。

新的评价函数:

$RP = \sum_{1 \leq j \leq J} W_j^i x_j$, 这是在协商过程中给出的评价函数值。

若 V_j^i 单调递减, 当 $RP < rp^i$ 时, Ag_i 可以进一步协商;

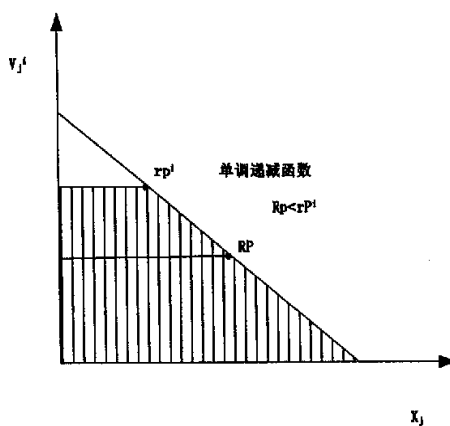


图3.1 V_j^i 单调递减函数图

图3.1的阴影为可以协商的区域。

若 V_j^i 单调递增, 当 $RP > rp^i$ 时, Ag_i 可以进一步协商。

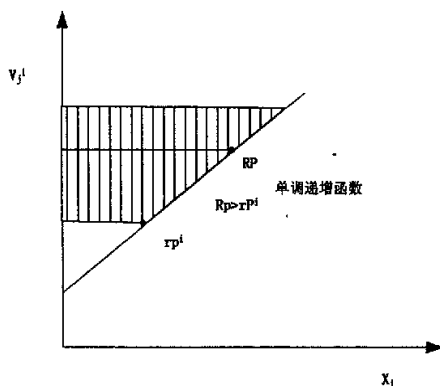

 图3.2 V_j^i 单调递增函数图

图3.2的阴影为可以协商的区域。

那么如果若 V_j^i 单调递减, 当 $RP > rp^i$ 时, 若 V_j^i 单调递增, 或 当 $RP < rp^i$ 时, 即超过买卖双方的底价值, 协商就会停止。

只有通过改变底价, 来重新进行协商。

评价函数增量的算法:

协商过程, Agent要经过多次协商才能达到满意的协商结果, 那么每次协商的增量就需要描述出来, 即

$V_j^i \leftarrow (V_j^i \pm \alpha)$ (α 是单调递增或单调递减的步长) α 可以是常量也可以是非线形的(可以是平方, 立方, 开平方, 指数, 幂函数关系等)当然也可是离散的量。这里我们只做最简单的假设认为是常量, 即设

$$\alpha = \frac{\min D_j^i}{\max D_j^i - \min D_j^i} \text{ 这样Agent每次都按照固定的增量增加进行协商, 直到}$$

协商成功, 或协商失败。

以上我们讨论了协商的全部过程, 通过讨论可以看出此协商模型研究的是双边多项目协商模型, 由于该协商模型具有一般性, 因此在存在广泛的适用空间。

3.3 协商协议

协商协议是双方Agent在协商开始前, 必须首先达成一致的规则。协议对参

与协商的Agent给出约束条件、规定协商状态(开始、结束等)及引起协商状态改变的有效事件。

下面给出一个5阶段协商协议，作为指导Agent进行协商交互的一般框架。协商过程限定在协商时间 T 范围内，若超时则协商失败。

- (1) 请求阶段，买方 $Agent_a$ 发出协商请求，卖方 $Agent_b$ 响应请求。
- (2) 回送阶段，卖方 $Agent_b$ 对买方 $Agent_a$ 发出协商请求作出决定。
- (3) 协商阶段，双方交互发出提议，直到达成一致或协商失败。
- (4) 反馈阶段，双方Agent将协商结果反馈给委托方(用户)。
- (5) 结束阶段，用户最终决定同意或拒绝协商结果，结束协商。

用户在结束阶段，可以改变项目 Q_i 的权重 W_j ，委托Agent开始一个新的协商会话。协商协议简单的可以以下的图3.3表示。

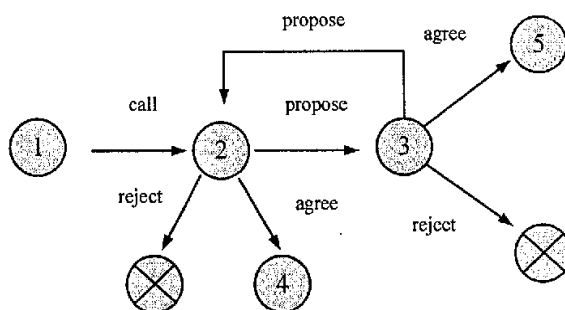


图3.3 协商协议

如图，当买方 Ag_a 向卖方 Ag_b 发出一个协商请求时 $call$ （状态1→状态2），

Ag_b 有三种情况：

- (1) 同意提议，协商成功 $agree$ 。（状态2→状态4）
- (2) 拒绝提议，协商失败 $reject$ 。（状态2→状态失败）
- (3) 发出提议，协商阶段 $propose$ 。（状态2→状态3）

同样卖方 Ag_b 向买方 Ag_a 发出协商提议时，

Ag_a 有三种情况：

- (1) 同意提议，协商成功 $agree$ 。（状态3→状态5）

(2) 拒绝提议, 协商失败 reject。(状态3→状态失败)

(3) 发出提议, 协商阶段 propose。(状态3→状态2)

也就是说在状态2, 状态3之间可能有多次交互, 在交互过程中, 达到协商成功或协商失败。

3.4 协商算法及协商流程

3.4.1 协商算法

针对上文提出的协商模型和协商协议, 下面给出一个Agent执行的协商算法。

假定协商中有两类Agent: 消费者 Ag_a 和生产者 Ag_b , rp 是 Ag_a 协商的底价, 即 Ag_a 能接受的对自己最不利的协商结果。 T 是协商期限。

算法: (Ag_a 方的算法流程)

(1) Ag_a 向 Ag_b 发出协商请求, 置 $time=0$, Ag_b 响应协商请求, 发送消息request。

(2) 若request是拒绝信息或 $time > T$, 转(11)。

(3) Ag_a 接收request, 计算 RP 。若 $RP > rp$, Ag_a 向 Ag_b 发出拒绝信息, 转(11);

否则, 准备与 Ag_b 进行协商。

(4) Ag_a 行协商初始化工作: $r=1$, $a=1$ 。

(5) 在第 r 轮协商中, Ag_a 向 Ag_b 以出提议 X_a^r 。

(6) Ag_b 收对方提议 X_a^r 。

(7) Ag_a 评价对方提议 X_b^r 。

(8) 若 Ag_a 同意对方提议 X_b^r , 则生成一致的合同, $X=X_a^r=X_b^r$, 转(10); 否

则, 若 $(X_a^r = X_a^{r-1}) \cap (X_b^r = X_b^{r-1})$, 则转(11); 否则转(9)

(9) 对下一轮协商是否让步, 做出决定: 若 $RP < rp$, 则让步, $a=a+1$, $r=r+1$ 。
否则, 坚持原提议, $r=r+1$ 。若 $time < T$, 转(5), 否则转(11)。

(10) X为达成的合同，协商终止，算法结束。

(11) 协商失败，算法结束。

同样，我们给出生产者 Ag_b 方的算法流程。 rp 是 Ag 协商的底价，即 Ag_b 能接受的对自己最不利的协商结果。 T 是协商期限。

(Ag_b 方的算法流程)

(1) Ag_b 等待协商请求，若接到 Ag_a 请求， Ag_b 响应协商请求，置 $time=0$ ，发送消息request。

(2) 若request是拒绝信息或 $time > T$ ，转(11)

(3) Ag_b 接收request，计算 RP 。若 $RP < rp$ ， Ag_b 向 Ag_a 发出拒绝信息，转(11)；否则，准备与 Ag_a 进行协商。

(4) Ag_b 行协商初始化工作： $r=1$ ， $b=1$ 。

(5) 在第 r 轮协商中， Ag_a 向 Ag_b 以出提议 X_a^r 。

(6) Ag_b 接收 Ag_a 提出提议 X_a^r 。

(7) Ag_b 评价对方提议 X_a^r 。

(8) 若 Ag_b 同意对方提议 X_a^r ，则生成一致的合同， $X=X_a^r=X_b^r$ ，转(10)； 否

则，若， $(X_a^r=X_a^{r-1}) \cap (X_b^r=X_b^{r-1})$ 则转(11)；否则转(9)

(9) 对下一轮协商是否让步，做出决定： 若 $RP > rp$ ，则让步， $b=b+1$ ， $r=r+1$ 。否则，坚持原提议， $r=r+1$ 。若 $time < T$ ，转(5)，否则转(11)。

(10) X为达成的合同，协商终止，算法结束

(11) 协商失败，算法结束。

此算法是建立在协商模型基础上的，具有一般性，通用性较强。

3.4.2 协商流程

下面给出算法的流程图如图3.4所示：

Ag_a 方的算法流程

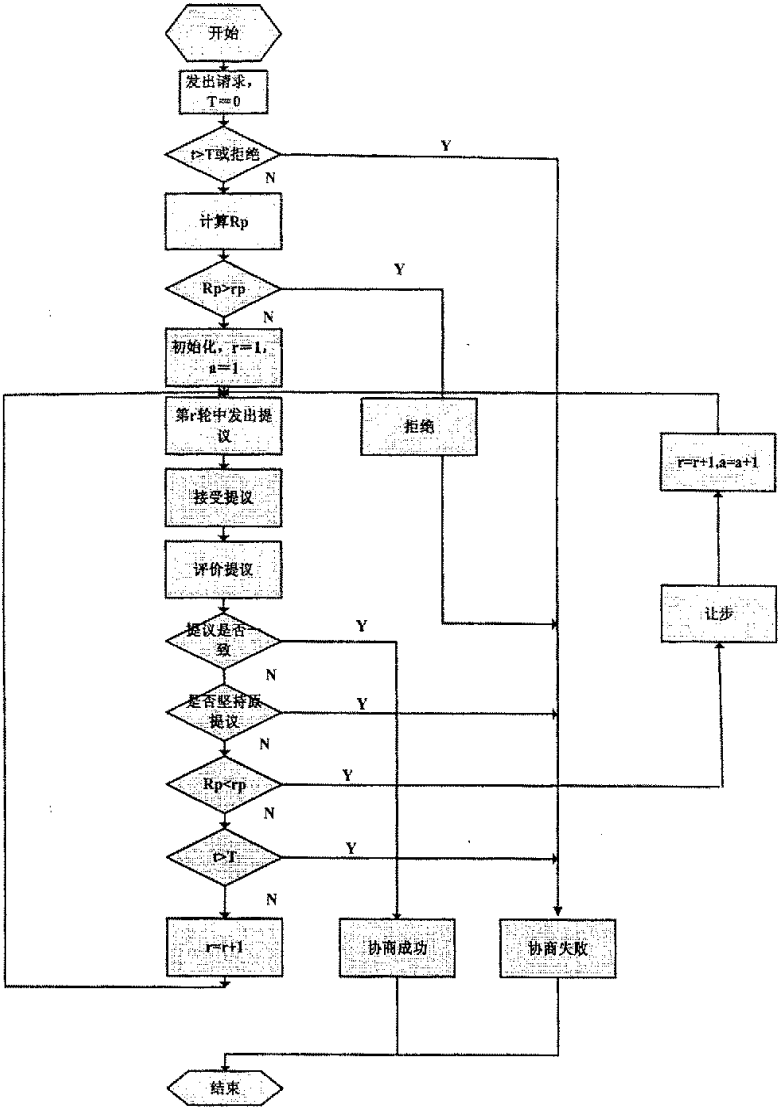


图3.4 Ag_a 方的算法流程

下面给出算法的流程图如图3.5所示：

Ag_b 方的算法流程：

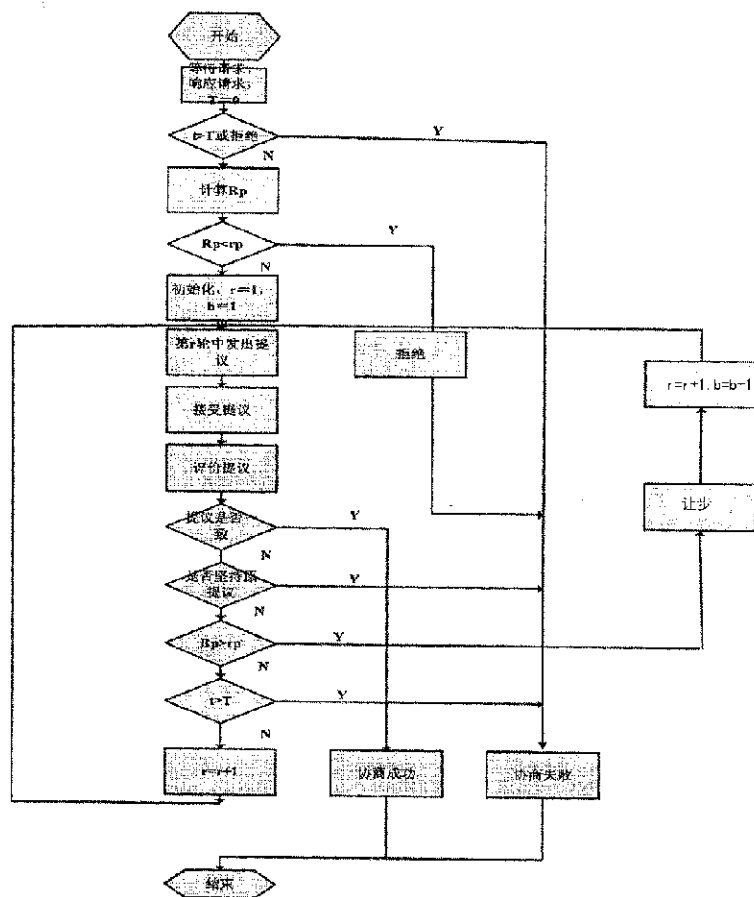


图3.5 Ag_b 方的算法流程

3.4.3 协商算法的通信描述

下面介绍 Agent 之间主要的协商交互信息和信息格式。即 Agent 之间的通信描述。

1. $Agent_a$ 发出协商请求。

(request

: sender $Agent_a$

: content 协商请求

: receiver $Agent_b$

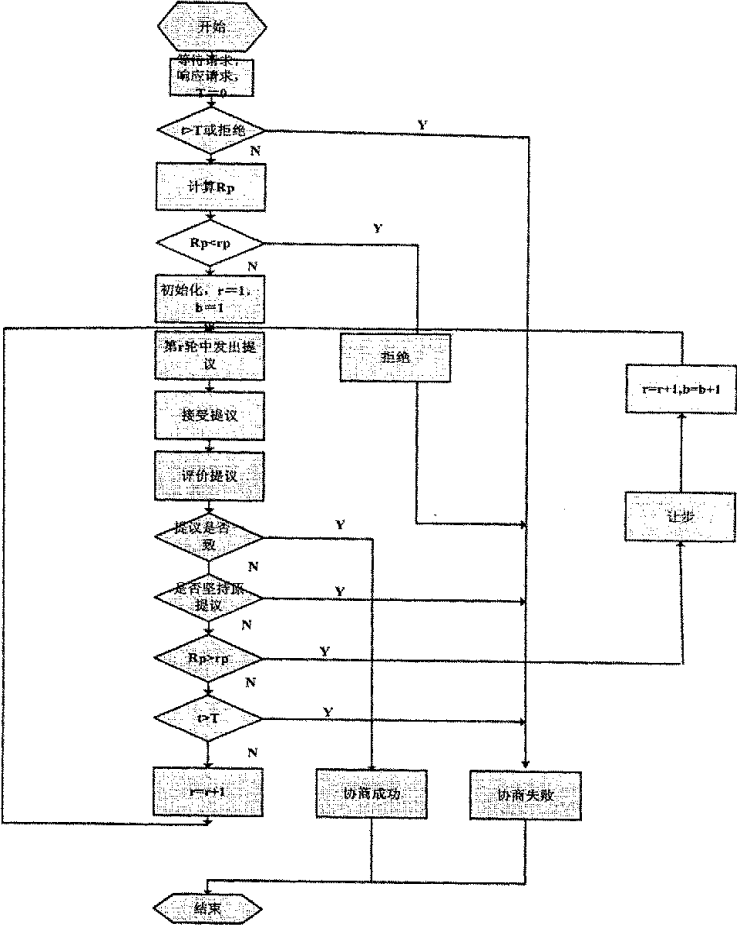


图3.5 Ag_b 方的算法流程

3.4.3 协商算法的通信描述

下面介绍 Agent 之间主要的协商交互信息和信息格式。即 Agent 之间的通信描述。

1. $Agent_a$ 发出协商请求。

(request
: sender $Agent_a$
: content 协商请求
: receiver $Agent_b$

: language KQML)

2. Agent_b 收到请求后, 决定接受或拒绝。

(accept

: sender Agent_b

: content 同意协商

: receiver Agent_a

: language KQML)

(reject

: sender Agent_b

: content 拒绝协商

: receiver Agent_a

: language KQML)

3. 在第 n 轮 Agent_a 发出协商提议。

(negotiate

: sender Agent_a

: content 发出的第 n 轮协商提议

: receiver Agent_b

: language KQML)

4. Agent_b 评价提议

(evaluate

: sender Agent_b

: content 评价提议

: receiver Agent_a

: language KQML)

5. Agent_b 给出新的提议

(newnegotiate

```

: sender Agentb
: content 给出新的提议
: receiver Agenta
: language KQML)

```

6. Agent_a 同意提议，协商成功。

(succeed

```

: sender Agenta
: content 协商成功
: receiver Agentb
: language KQML)

```

7. Agent_a 不同意提议，协商失败。

(lost

```

: sender Agenta
: content 协商失败
: receiver Agentb
: language KQML)

```

3.5 小结

本章是本文的重点也是难点，讨论了基于 Agent 协商的算法的模型，可以分为三部分。第一部分：讨论了协商模型，其中协商模型包括协商协商函数和决策函数，详细介绍了该协商模型在协商过程中全部的步骤。第二部分：介绍了协商协议。第三部分：讨论了协商算法、协商流程及 Agent 之间协商的通信描述。

第四章 基于 Agent 协商系统的设计

4.1 系统简介

本章主要是针对电子协商提出了一个多 Agent 的电子交易框架,来解决电子商务中由于协商所花费的大量工作和时间,提高交易效率和有效降低交易成本的缺陷,而采用基于 Agent 技术进行自动协商的框架。本章讨论了 B2C 电子商务活动的部分过程,包括用户的注册、用户搜索商品信息、自行比较相关商品、双方的协商、交易、以及在商务活动中的对商品信息的管理问题。框架中以多 Agent 代替交易双方的参与人员,形成由卖方多 Agent 系统和买方多 Agent 系统组成的新营销模式。由于在电子商务中交易本身是一种很复杂的行为,涉及到第三方系统,加上实验室的条件限制,只设计了一个虚拟电子市场系统。市场中存在三类成员:买方多 Agent 系统,卖方多 Agent 系统和服务器管理员。

4.2 系统的体系结构

结构整个系统采用 C(客户机)/S(服务器)体系结构,如图 4.1 所示。在不同的客户机上有卖方 Agent 买方 Agent;服务器上则有服务器管理员。其中买方 Agent 和卖方 Agent 都是多 Agent 系统,采用分工明确的多种 Agent 进行搜索、协商、交易等工作。

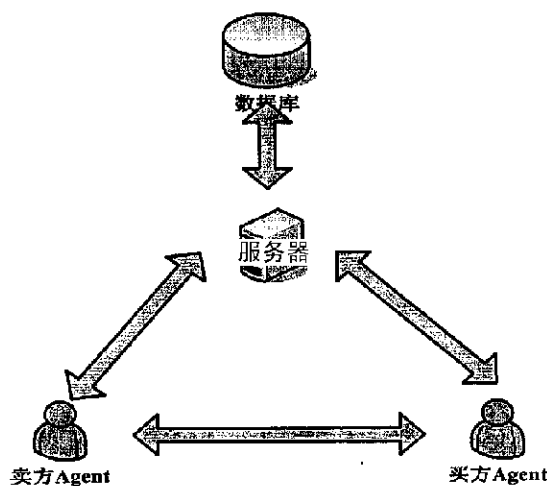


图 4.1 系统的体系结构图

4.2.1 服务器管理员

服务器管理员是这个电子市场中唯一的管理者。它主要具有下述三种功能。如图 4.2 所示。

- (1)注册：当系统中生成一个管理 Agent 时，服务器管理员注册它的基本信息(包括创建者、创建时间)，赋予这个管理 Agent 唯一的身份识别(ID)号。使用这个 ID 号，管理员可以方便的管理和协调系统中其他的 Agent，同时不同的移动 Agent 之间可以使用各自的 ID 号通讯和交流。
- (2)登记：服务器管理员负责管理登记系统中由买卖方管理 Agent 系统向系统添加那些市场中新出现的买卖方 Agent 系统信息，删除那些退出市场的买卖方 Agent 系统，并赋予唯一的身份识别(ID)号。同时管理员还负责管理登记新添加的商品信息，包括买卖方的站点地址，卖方商品的电子目录，商品的种类、最初定价及买方的需求信息，把这些信息存放在卖方或买方的数据库里。
- (3)查询：服务器管理员负责接待来自买卖方的搜索 Agent 的查询工作。当有搜索 Agent 来寻找所需商品信息时，管理员会提供相应的买卖方商品信息和站点地址。并记录查询信息和搜索 Agent 的唯一 ID 号。

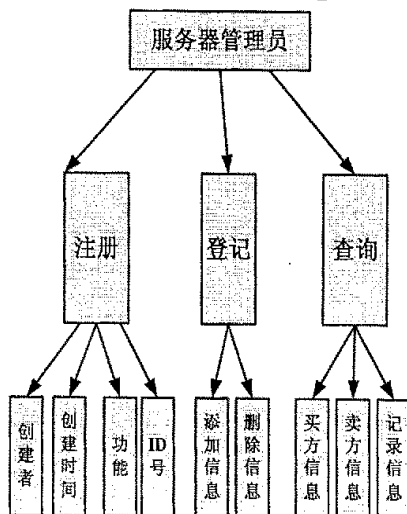


图 4.2 服务器管理员的功能

4.2.2 买方多 Agent 系统

4.2.2.1 买方多 Agent 系统简介

买方多 Agent 系统这个系统是由需要购买商品的用户在任意时间、任意站点生成的，并向服务器管理员注册登记信息，退出市场也要向服务器登记。在用户生成这个系统时，用户需要提供自定义的信息，其中包括：商品名称、商品种类、理想价格、最高可以接受的价格范围、协商中的策略、购买商品的最后期限、交货的地点。系统会向买方提供几种固定的协商策略，供买方用户选择。用户所提供的各种要求都会被存贮在用户需求库中。买方系统是一个多 Agent 系统，包括搜索 Agent、决策 Agent、协商 Agent、管理 Agent，交易 Agent，还有一个用户需求库。搜索 Agent、协商 Agent、交易 Agent 是移动 Agent，可以在电子市场中移动。如图 4.3 所示。

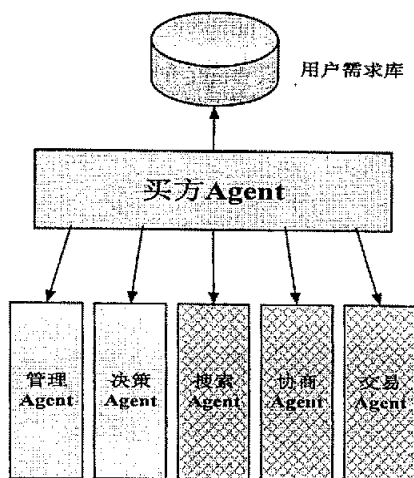


图 4.3 买方 Agent 系统的结构

4.2.2.2 买方多 Agent 系统功能

买方 Agent 系统中各种 Agent 的详细描述如下：如图 4.4 所示。

管理 Agent:

- (1)负责管理系统中各种 Agent 的协调和协作问题，以及系统的安全

问题。

(2)在买方多 Agent 系统生成时,添加 Agent 信息和添加用户需求信息存贮到用户需求库中。

(3)在协商结束,决策 Agent 决定交易伙伴后,管理 Agent 负责交易后续的管理。

(4)提供人机接口,用户可以直接干预,使用户可以方便的控制整个交易的各个阶段。

决策 Agent:

主要负责做出各种决策。

(1)负责比较搜索 Agent 返回的各种商品信息,决定哪些商品需要进行协商。然后把决策的结果通知给协商 Agent。

(2)限定的购买商品的最后期限,决定合适的协商时间,即一轮协商中买方 Agent 能够等待、能够忍受的协商时间。

(3)在和商家的协商完成以后,因为可能会有多个商品的符合要求,负责决策交易的伙伴,返回交易 Agent,由交易 Agent 进行交易。

搜索 Agent:

根据用户的需求,它首先移动到服务器管理员站点,搜索商家的电子目录,查找那些符合用户自定义条件的商品,得到相应卖方的站点信息,然后把找到的信息返回给买方的决策 Agent。

协商 Agent:

根据决策 Agent 的决定的结果,即需要协商的商家的个数,它生成多个子协商 Agent,由这些子协商 Agent 分别移动到相应的卖方站点,和卖方进行协商。这样协商的过程是并行的,节省了系统的协商时间。最后它综合多个子协商 Agent 的协商结果,协商成功后将最后的信息返回给决策 Agent。

交易 Agent:

主要负责交易。当接到决策 Agent 的交易命令后,交易 Agent 移动到指定卖方站点通过认证中心,电子银行等第三方机构与卖方交易 Agent 进行交易。

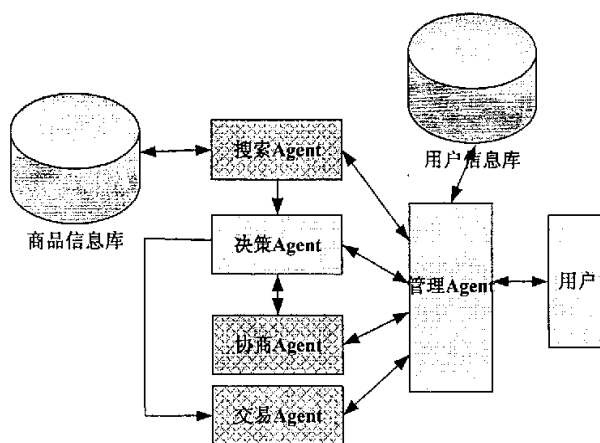


图 4.4 买方 Agent 系统的功能图

4.2.3 卖方多 Agent 系统

4.2.3.1 卖方多 Agent 系统简介

这个系统是由供应商品的卖方在任意时间、任意站点生成的，存在的时间长短卖方自主决定的，但是它的生成和消亡必须向服务器管理员注册，卖方系统也是一个多 Agent 系统，包括管理 Agent、协商 Agent、决策 Agent、交易 Agent、搜索 Agent，及商品信息库数据库。其中搜索 Agent 协商 Agent、交易 Agent 是移动 Agent。如图 4.5 所示。

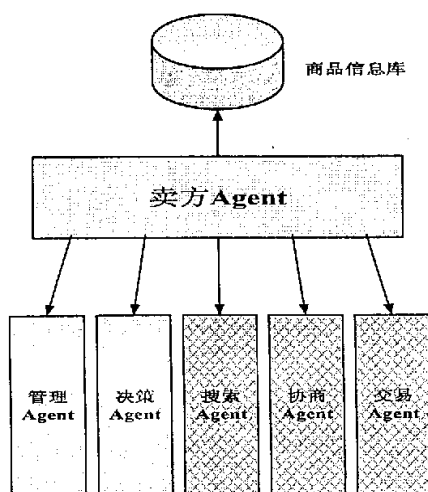


图 4.5 卖方 Agent 系统的结构

4.2.3.2 卖方多 Agent 系统功能

系统中各种 Agent 的详细描述如下：如图 4.6 所示。

管理 Agent:

它主要负责以下 4 个方面的工作。

- (1)负责系统中多个 Agent 之间的协调和协作, 以及系统的安全问题。
- (2)在卖方多 Agent 系统生成时, 添加 Agent 信息和负责更新商品信息库, 向商品信息库中记录商品信息, 每条商品信息包括商品名称、商品种类、商品的初始定价、最低可以接受的价格范围、商品必须出售的最后期限。
- (3)在卖方系统被生成和消亡时, 它负责向服务器管理员登记自己的开张和结束信息, 开张时需要向服务器管理员登记自己的商品信息, 商品的电子目录和站点信息。然后, 在卖方的商品目录发生变化时, 负责和服务器管理员联系更新。
- (4)负责协商成功后具体的交易后续管理的问题。
- (5)提供人机接口, 用户可以直接干预, 使用户可以方便的控制整个交易的各个阶段。

搜索 Agent:

根据用户的需求, 它首先移动到服务器管理员站点, 搜索买家的电子目录, 查找那些符合自己条件的买家, 得到相应买方的站点信息。向其发布信息。然后把找到的信息返回给卖方的决策 Agent。

协商 Agent:

根据等待协商的买方个数, 它可以生成多个子协商 Agent, 分别和多个买方进行协商。每个子协商 Agent 根据协商算法, 选择合适的协商策略进行协商, 然后把协商的结果返回给决策 Agent。

决策 Agent:

主要负责做出各种决策。

- (1)Agent 的协商结果做决策, 决定是否与其协商。返回交易 Agent, 由交易 Agent 进行交易
- (2)选择买方。

交易 Agent:

主要负责交易，与买方 Agent 进行安全的交易。

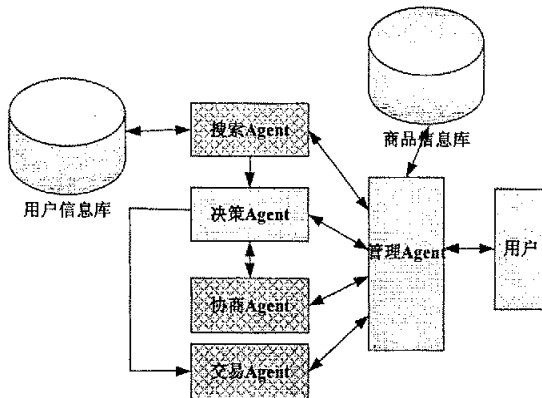


图 4.6 卖方 Agent 系统功能图

4.3 系统的工作流程

以下将从一个买方用户的角度，描述他在整个电子市场的工作过程。

一次完整的购买过程：如图4.7所示。

- (1) 户生成一个买方多Agent系统。向它提出自己的购买要求。
- (2) 根据用户的需求，搜索Agent移动到服务器管理员站点，查找其上的电子目录，得到相应的卖方站点信息。
- (3) 搜索Agent把查找结果返回给决策Agent。
- (4) 决策Agent决定：有必要进行协商的商家个数及站点地址，把决策结果发送给协商Agent。
- (5) 根据决策Agent做出的结果，协商Agent生成多个子协商Agent，把他们派送到相应的卖方站点进行协商。然后某一个子协商Agent移动到某一个卖方站点，开始和对方的子协商Agent进行协商。
- (6) 协商Agent总结所有子协商Agent的协商结果，把成功的协商信息发送给决策Agent。
- (7) 决策Agent分析协商Agent发送的信息，决定交易的伙伴。把交易信息发送给交易Agent。
- (8) 交易Agent负责具体的交易事项。

(9)管理Agent在整个流程中负责协调与管理，同时用户可以通过控制管理Agent进行用户的直接控制。

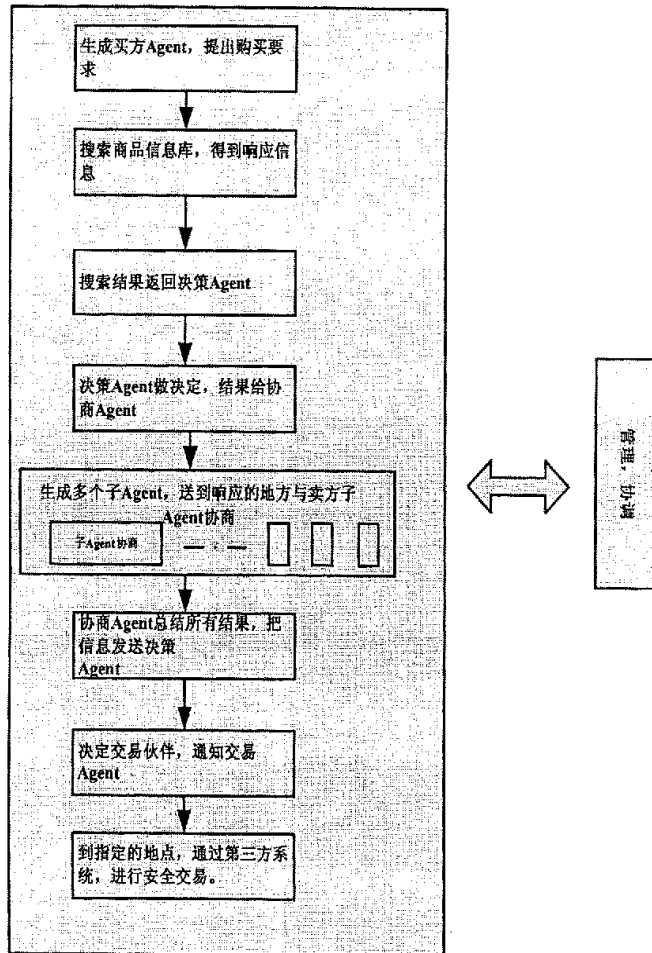


图4.7 系统的工作流程图

以上过程只描述了一切步骤处理顺利的情况。但是在实际过程中由于各种不可测因素的出现，比如市场的主观原因，很可能整个过程只进行上述过程其中的几个步骤就终止。

比如：

- (1) 在市场管理员站点上，搜索Agent没有找到用户所需的商品。
- (2) 所有的子协商Agent在给定的协商时间内都失败。
- (3) 在决策Agent和用户交互的过程中，用户不满意搜索或者协商的结果。
- (4) 在用户自定义的购买商品的最后期限，买方系统依然没有找到适合条件

的商品。

对于以上问题，我们提供了可行的处理方案。如果出现问题，把失败的结果返回给用户，由用户决定是否开始新一轮新的商品购买过程，当然系统会根据出现的问题为用户提出一些有用的建议，比如制定更合理的商品的搜索范围，修改购买商品的最后期限，制定合理的协商忍受时间，供用户选择。对问题(2)，如果购买的最后期限还没有达到，由决策Agent决定：或者修改协商可忍受时间，或者修改可接受的协商范围，重新开始新的价格协商过程。图4.8为系统的整个功能图。

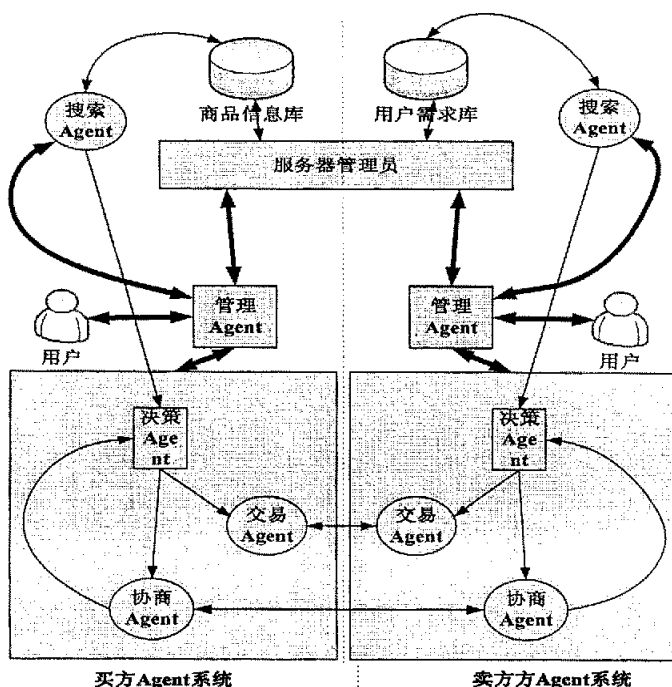


图4.8 系统的功能图

4.4 小结

本章是本文的重点章节，提出了基于 Agent 协商系统的体系结构，可分四部分。第一部分：介绍了服务器管理员的结构和功能。第二部分：介绍了买方多 Agent 系统的结构和功能。第三部分：介绍卖方多 Agent 系统的结构和功能。第四部分：通过具体的实例给出系统的工作流程。

第五章 基于 Agent 协商原型系统的实现

5.1 系统运行环境

我们的系统分为客户端和服务端两部分，为了保证系统运行效率和可靠性，服务器应具有较高的硬件配置，客户端的要求不是很高。运行环境如图 5.1 所示

其运行要求如下：

软件环境：客户端：Windows98、Windows2000、WindowsXP、IE5.0 等。服务器：WindowsNT/ Windows2000。编程语言：JAVA，编辑工具是 JBuilder 9，数据库：采用 SQL2000。

硬件环境：服务器 CPU：P41.8G，内存：512M 以上，硬盘：80G 以上。客户机：CPU：PIII500 以上，内存：256M 以上，硬盘：40G 以上。

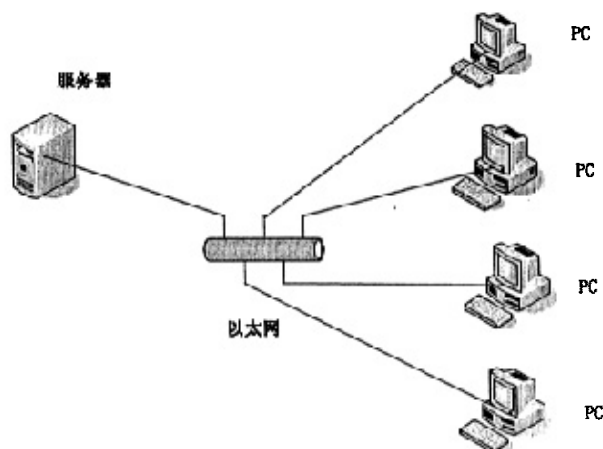


图 5.1 运行环境

5.2 系统 UML 图

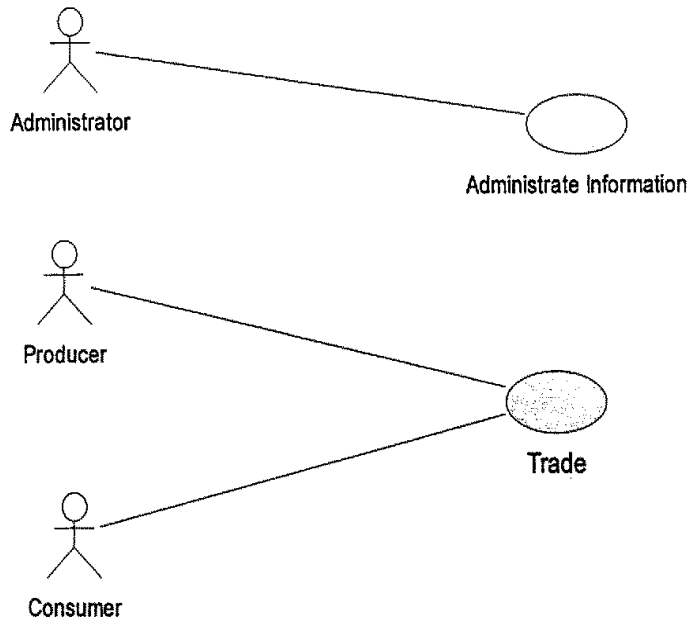


图 5.1 系统的用例图

图 5.1 为系统的用例图。

整个系统的分为两部分用例。一部分是服务器管理员的用例如图 5.1 中 Administrate information，它包括服务器管理员的功能。另一部分是卖方与买方的交易用例，如图中的 Trade，它是买卖双方协商交易的部分，买卖双方的所有功能都是在此实现的。

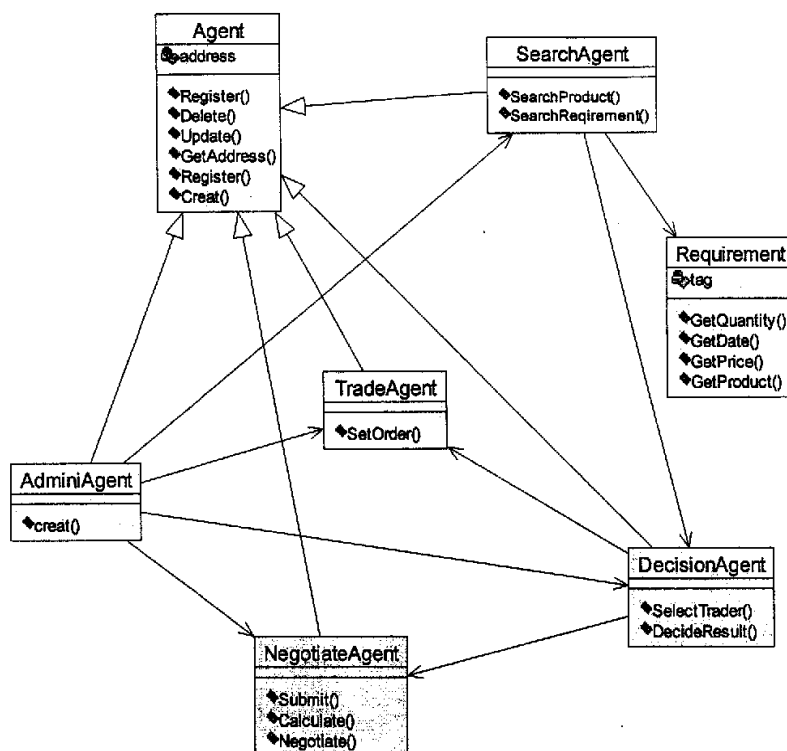


图 5.2 Agent 系统关系的类图

图 5.2 为 Agent 系统关系的类图。

其中 Agent 类是基类。SearchAgent、AdminiAgent、TradeAgent、NegotiateAgent、DecisionAgent 与 Agent 是泛化的关系。AdminiAgent 与其他的 SearchAgent、TradeAgent、NegotiateAgent、DecisionAgent 是依赖关系。DecisionAgent 与 TradeAgent、NegotiateAgent 也是依赖关系。SearchAgent 与 Requirement 是依赖关系。SearchAgent、AdminiAgent、TradeAgent、NegotiateAgent、DecisionAgent 分别是搜索 Agent 类、管理 Agent 类、交易 Agent 类、协商 Agent 类、决策 Agent 类。Requirement 是需求库，通过 Tag 的数值来决定是商品库还是用户需求库。Agent 类有很多方法，其中有 Register() 是添加那些市场中新出现的买卖方 Agent 系统信息。Delete() 是删除 Agent 信息。Update() 是修改 Agent 信息。GetAddress() 是获得站点地址的信息。Creat() 是创建 Agent 的信息等。NegotiateAgent 是协商 Agent 类，其中 Submit() 是发出协议，Calculate() 是计算权值的，Negotiate() 是协商的方法。

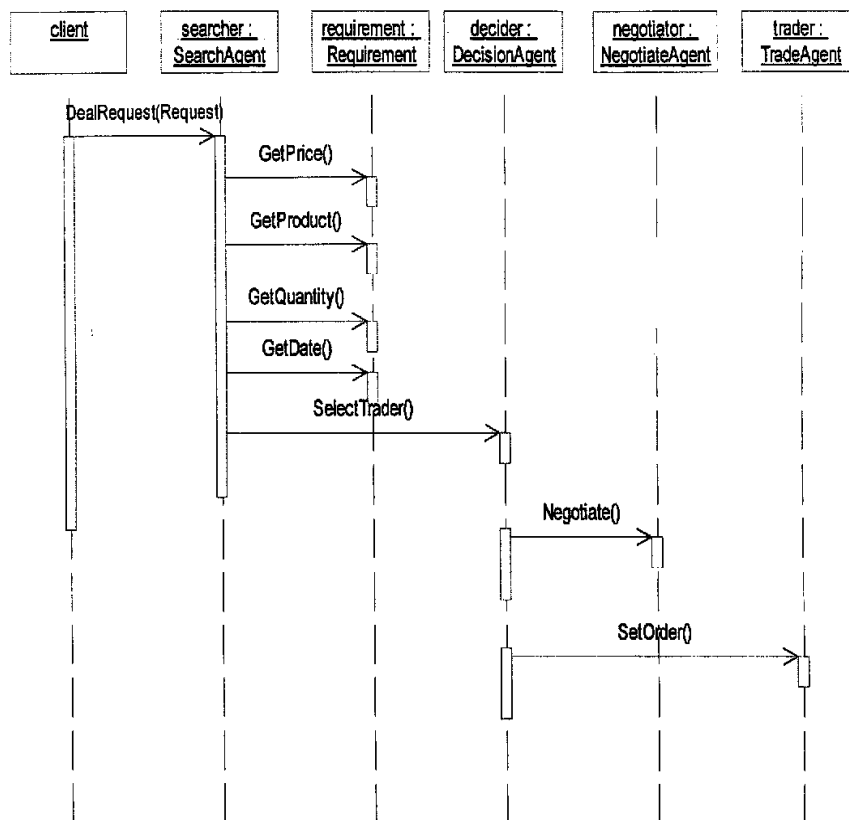


图 5.3 系统交易的时序图

图 5.3 为系统交易的时序图。

以买方为例，首先买方提出需求的请求给搜索 Agent Reques ()，搜索 Agent 到商品的信息库搜索商品的价格 GetPrice ()、商品的种类 GetProduct ()、商品的数量 GetQuantity ()、商品的日期 Get ()，把搜索的结果 SelectTrader () 给决策 Agent，决策 Agent 调用协商 Agent 进行协商 Negotiate ()。决策 Agent 在选择订单 SetOrder () 调用交易 Agent 进行交易。

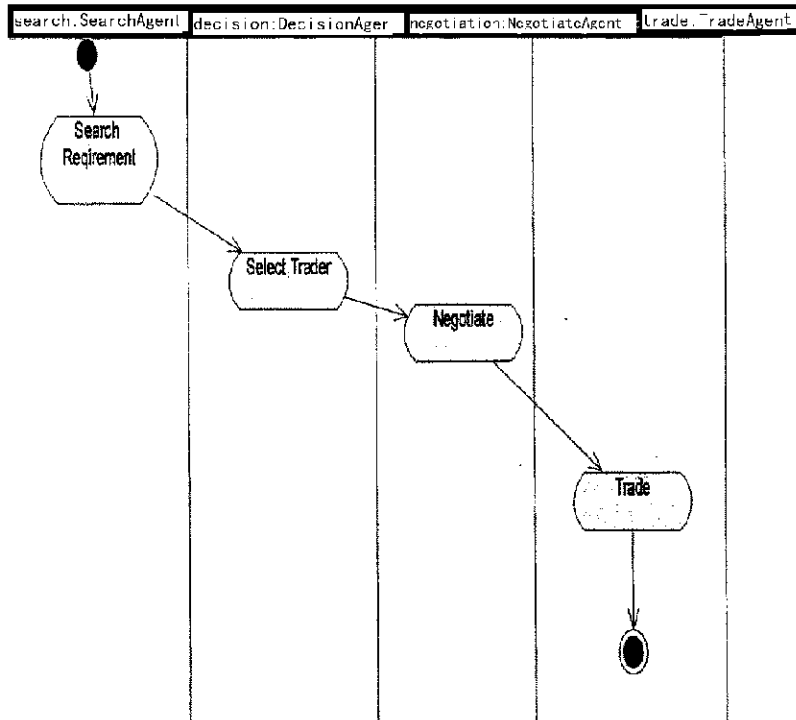


图 5.4 系统交易的活动图

图 5.4 系统交易的活动图。

搜索 Agent 到信息库中搜索商品 Search Requirement, 之后决策 Agent 选定商品 Select Trader, Agent 选定商品给协商 Agent 进行协商 Negotiate, 最后由交易 Agent 进行交易 Trade。

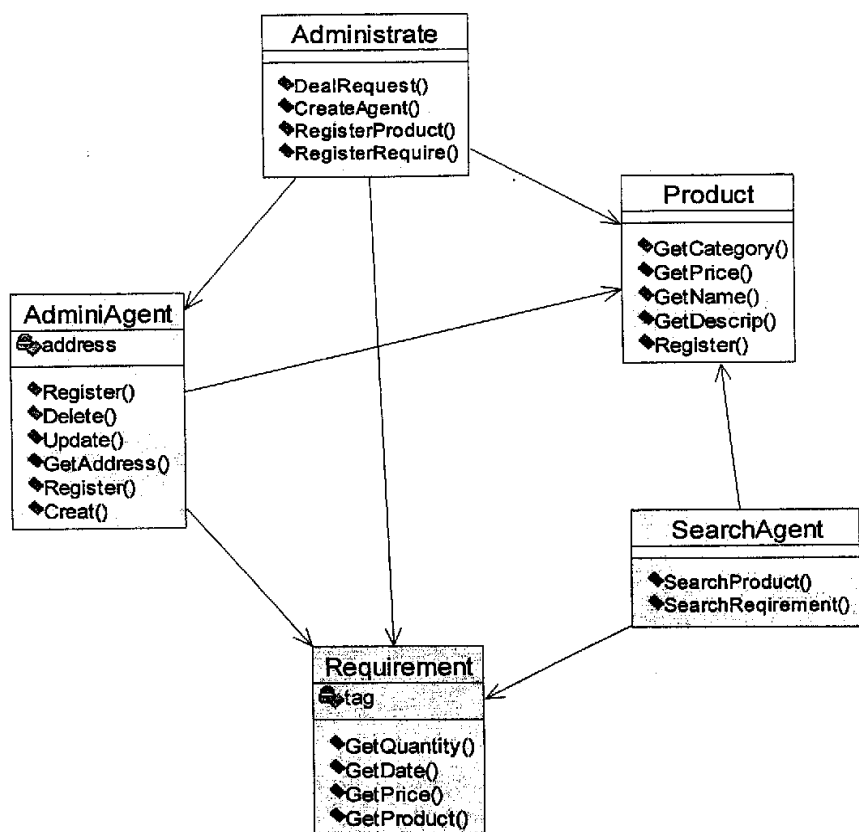


图 5.5 管理员管理信息的类图

图 5.5 为管理员管理信息的类图。

Administrative 是服务器管理员类，AdminiAgent 是管理 Agent 类，Product 是商品信息库类，Requirement 是用户需求库类，SearchAgent 是搜索 Agent 类。服务器管理员与 AdminiAgent、product、Requirement 是依赖关系。AdminiAgent 与 product、Requirement 也是依赖关系。SearchAgent 与 product、Requirement 是依赖关系。

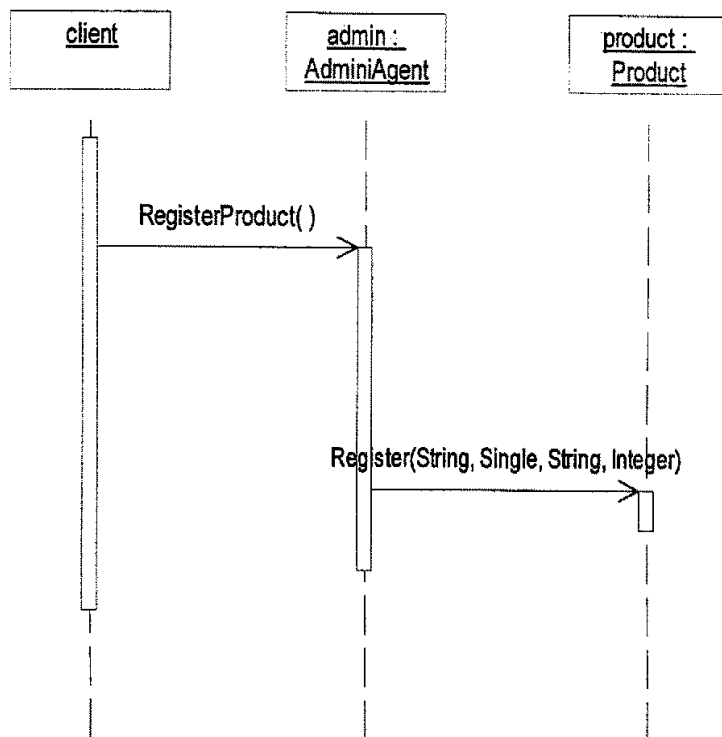


图 5.6 注册的时序图

图 5.6 注册的时序图。

当买卖方有注册请求时 `RegisterProduct()`，把请求发送给管理 Agent，管理 Agent 进行注册 `Register (String, Single, String, Integer)`。

5.3 系统的运行实例

下面我们通过一个具体的实例来看看系统的运行情况。

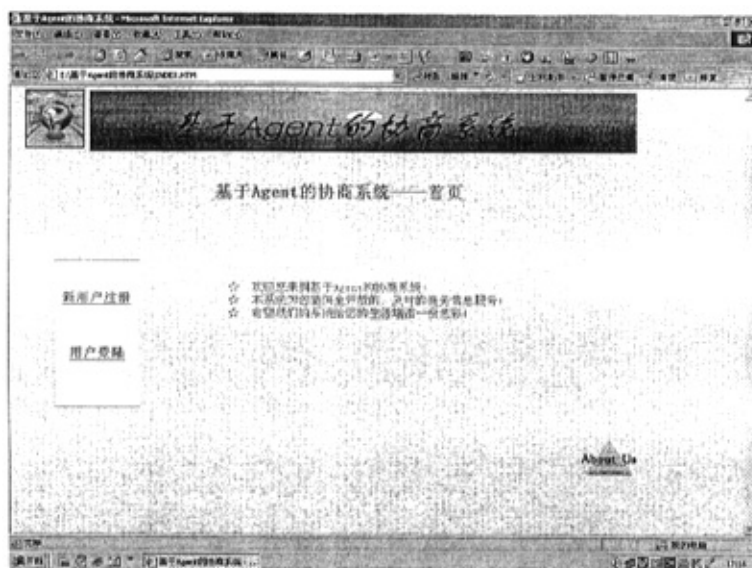


图 5.7 基于 Agent 的协商系统的首页

我们假设是一个买方的用户为例，他首先要登陆我们的系统，如图 5.7 是基于 Agent 的协商系统的首页，在首页里共有两个选项，分别是新用户注册和用户登陆选项。如果他是第一次登陆我们的系统，要先注册，他就要点击注册功能选项，进入注册的页面，按照页面提示的要求一步一步完成。如果注册成功，他将成为我们系统的合法用户，就可以登陆我们的系统完成他想要的操作。

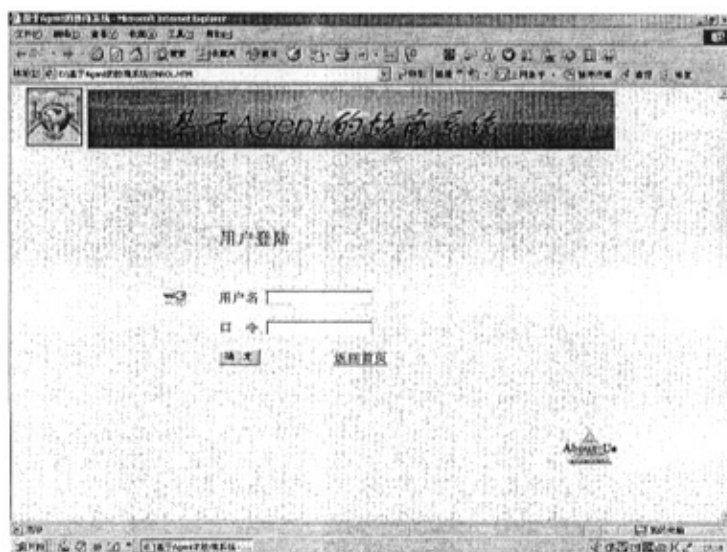


图 5.8 系统登陆运行界面图

当他再次登陆我们的系统，他已是我们系统的用户，就不要注册了，只需点击用户登陆按钮进入登陆页面，如图 5.8 是系统的登陆界面图，输入正确的用户名及口令，就可以进入我们的系统，如图 5.9 是系统的功能选项界面图。



图 5.9 系统的功能选项界面图

在功能选项页面中共有五个功能选项分别为：登记、管理、查询、协商、交易功能选项，它们分别执行不同的功能。

如果他要买 CD 这个商品，他就要先登陆查询功能页面，如图 5.10 查询功能选项界面图，其中查询包括商品的信息库与需求信息库查询。他要买商品 CD，所以他点击商品信息库，如图 5.11 所示。

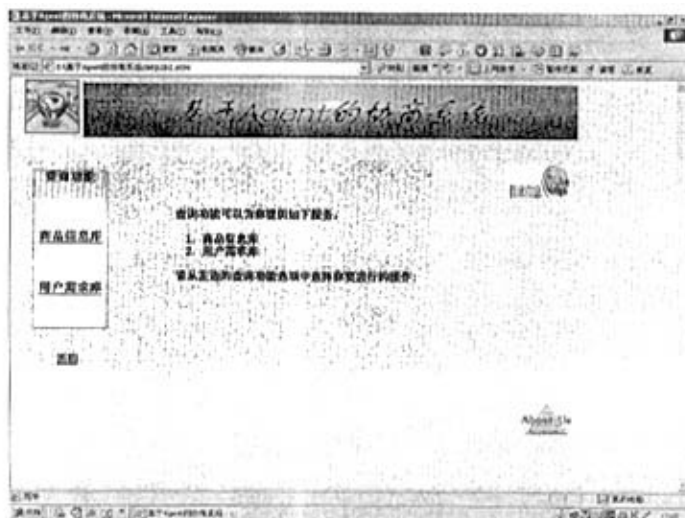


图 5.10 查询功能运行界面图



图 5.11 商品信息库的查询运行界面图

图 5.11 这是查询功能中的商品信息库的查询页面，通过此功能可以查到比较接近满意度的结果。例如现在他想要买的 CD 价格在 10 元 RMB 范围之内，交易时间、交易地点分别是 3 天、杭州等，那么输入上面的条件后点击确定就会出现很多已经添加好的商家的信息，他可以选择一家进行交易。



图 5.12 协商功能运行界面图

例如，他选择了金金 CD 专卖店进行协商，他就把要协商的项目，及其协商

项目对应的权值按照自己的偏好输入，这里商品的价格选的是 1—10RMB 对应的权值是 0.6，商品的数量是 20 张，所以他选了 1—50，对应的权值是 0.3，商品交易时间是 1—3 天，交易的地点是杭州，由于他认为交易的时间不重要所以权值选为 0 数。剩下的交易地点的权值为 $1 - 0.6 - 0.3 = 0.1$ ，由于他要买 20 张 CD，数量很多，他认为价格可以更低一些，所以他又选中了最下面商品的价格，并选中权值 0.7。点击确定就与金金 CD 专卖店开始进行了协商，图 5.12 为商品进行协商的界面图。

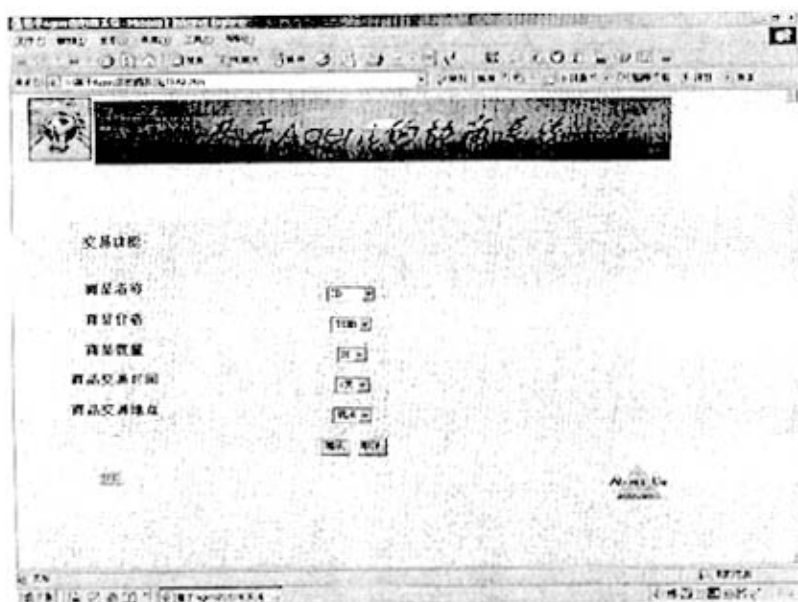


图 5.13 交易功能运行界面图

通过协商最后得到协商结果是价格 7RMB，数量 20 张，交易的时间 1 天，交易的地点是杭州，如图 5.13 是交易功能运行界面图。如果决定交易，就点击确定进行交易。

5.4 小结

本章是系统的原型系统的实现部分，可以分为三部分。第一部分：介绍了系统的运行环境，包括系统平台，编程工具，数据库管理系统。第二部分：介绍了系统的 UML 图，包括系统的用例图，Agent 的类图，工作流程的时序图等。第三部分：通过具体的实例来说明系统的运行情况，并给出系统的部分运行界面图。

第六章 总结与展望

6.1 本文的总结

本文主要围绕着电子商务中的关键部分协商展开论述的。论述了基于 Agent 的电子协商系统的设计与实现的研究,设计并实现了基于 Agent 的电子协商系统的原型系统。其中包括系统的体系结构的设计、协商模型设计、以及原型系统的实现等。

第一章:概述了本文研究的背景,分析了研究的意义和必要性,介绍目前本课题的研究现状,讨论分析了目前研究现状存在的问题,以及本文研究的主要内容和本文的组织结构。

第二章:详细的介绍了与文本有关的背景知识。这些知识包括电子商务技术知识、Agent 技术知识、移动 Agent 技术知识。这些背景知识是本文的论述基础知识。

第三章:提出了基于 Agent 协商模型。这一章内容既是本文的重点也是本文的难点章节。目前已有多种协商的理论、方法,主要是基于对策论和基于启发式的协商,但都不完善。本文主要研究双边多项目协商问题,是基于意图的协商。本章给出了相应的协商模型、协商协议、协商算法、协商流程等。

第四章:提出了基于 Agent 协商系统的体系结构。这一章内容为本文的重点章节。本章给出了电子商务活动的部分过程,其中包括搜索、协商、交易等活动。并以多 Agent 代替交易双方的参与人员,形成由卖方 Agent 和买方 Agent 组成的新交易模式。

第五章:介绍了本文提出的基于 Agent 协商原型系统的实现。主要介绍了系统运行的环境、系统的 UML 图、以及实例运行部分界面图。

第六章:总结了本文的主要研究的成果,以及创新点,并指出进一步需要研究的工作。

6.2 本文的创新点

本文主要设计并实现了基于 Agent 的协商的原型系统。

1. 本文提出了一个基于 Agent 的电子交易框架, 框架中以 Agent 代替交易双方的参与人员, 围绕着买卖双方对商品的协商问题。讨论 B2C 电子商务活动的部分过程, 包括用户搜索商品信息、双方的协商、交易, 以及商务活动当中买卖双方对商品信息的管理问题。

2. 本文主要研究双边多项目协商问题, 给出了相应的协商模型、协议和协商算法。讨论了协商的全部过程, 由于该协商模型具有一般性, 因此在存在广泛的适用空间。

6.3 未来工作的展望

协商问题是一个多学科交叉的研究领域, 覆盖经济学、社会心理学、人工智能等不同的领域。随着 Agent 技术在电子商务应用领域的发展, 客户和商家可以委托 Agent 进行自动协商, 实现网上交易。由于 Agent 技术本身还不完善, 本文提出的框架在具体实施中, 还存在如下问题: Agent 在网络中自主移动带来的安全问题; 以及如何更好地发挥 Agent 的合作、通信、协调能力, 提高系统解决问题的能力 and 效率, 这些问题还需进一步研究。本文虽然主要讨论双边多项目协商问题, 给出了相应的协商模型、协议和协商算法, 具有一定的通用性, 但在协商中如何考虑双方的信誉问题、如何增强 Agent 的学习能力来提高协商效率和收益等方面, 也还需进一步的研究。以上这些都是今后要研究的方向。

参考文献

- [1]. 史忠植著. 智能主体及其应用[M]. 科学出版社, 2000
- [2]. 黄敏学主编. 电子商务(第二版)[M]. 高等教育出版社, 2004
- [3]. 黄敏学主编. 企业电子商务[M]. 武汉大学出版社, 2002
- [4]. 周蓓等. 软件 Agent 技术[J]. 信息工程大学学报. 2002(3)
- [5]. 陈松. 移动 Agent 综述[J]. 计算机科学. 2002(7)
- [6]. 万蜀柏, 黄秋红. 移动 Agent 技术在电子商务信息系统中的应用. 情报技术. 2003(9)
- [7]. 罗晓等. 关于协商理论的一个介绍[J]. 运筹学杂志. 1993(12)
- [8]. 黄关山, 徐冬梅. Agent 的理论与结构模型分析[J]. 微机发展. 2003(11)
- [9]. 罗健, 王成良. 基于 KQML 的 Agent 通信体系结构设计[J]. 重庆大学学报. 2004(3)
- [10]. 赵京胜等. 移动 Agent 技术的研究与应用[J]. 青岛建筑工程学院学报. 2002(1)
- [11]. 肖文鹏. 移动 Agent : 智能化分布式计算新模型
[http://www.recursionsw.com/Mobile_Agent_Papers/Intelligent%20Distributed%20Computing%20Using%20Mobile%20Agent%20\(Chinese\).pdf](http://www.recursionsw.com/Mobile_Agent_Papers/Intelligent%20Distributed%20Computing%20Using%20Mobile%20Agent%20(Chinese).pdf)
- [12]. 谢婷. 基于 Agent 的电子商务系统中的价格协商模型研究[D]. 武汉大学计算机学院.
- [13]. 陈宁江等. 移动 Agent 在电子商务中的应用研究[J]. 计算机应用. 2000(12)
- [14]. 刘欣, 高济. 自治 Agents 的面向服务的协商模型[J]. 计算机工程 2003(10)
- [15]. 周三元, 杜运普. 电子市场 Agent 协商行为机制分析[J]. 辽宁工学院学报. 2003(5)
- [16]. 李勇, 李石君. 多 Agent 的自动协商[J]. 计算机工程. 2003(4)
- [17]. 谢婷, 孟波. 一种基于协商和 Agent 的电子交易框架[J]. 计算机工程与应用. 2003(24)
- [18]. 尹红丽等. 多 Agent 系统协商问题的技术和方法[J]. 云南师范大学学报. 2003(12)
- [19]. 张云勇, 刘锦德编著. 移动 Agent 技术[M]. 清华大学出版社. 2003
- [20]. 张云勇编著. 移动 Agent 及其应用[M]. 清华大学出版社. 2002
- [21]. NeC. 电子商务理论与实践: <http://www.fjtu.com.cn/fjnu/courseware>
- [22]. Ladislav Boloni and Marinescu. A Component Agent Model—from Theory to Implementation[J]. Purdue University, 1999. <http://bond.cs.ucf.edu/docs>
- [23]. Karl — Fredrik Blixt and Rickard Oberg[J]. Software Agent Framework

- Technology, 2000. <http://bond.es.ucf.edu/does>
- [24]. Joshi N, Ramesh V C. On Mobile Agent Architectures[J]. <http://vcr.ece.iit.edu>
- [25]. L. Thompson and J. Nadler. Negotiating via Information Technology: Theory and Application[J]. Journal of Social Studies, 2002
- [26]. R. Lewicki, D. Saunders, and J. Minton. Essentials of Negotiation. [J] Irwin, 1997
- [27]. A. G. Cass, H. Lee, B. Staudt Lerner, and L. J. Osterweil. Formally Defining Coordination Processes to Support Contract Negotiation[J]. University of Massachusetts, Amherst, MA, June 1999
- [28]. Wooldridge, M., N. R. Jennings. Agents Theories, Architectures and languages: a Survey. In Wooldridge and Jennings, Intelligent Agent, Berlin: Springer-Verlag, 1995
- [29]. Genesereth, M and Steven P. Ketchpel. Software Agent[J]. Communications of the ACM, 1994
- [30]. Maes, P. Agents that reduce work and Information overload[J]. Communication of the ACM, 1994
- [31]. Minsky, Marvin. The Society of Mind. New York: Simon and Schuster, 1985
- [32]. Bate, J. The Role of Emotion in believable Agents[J]. Communications of ACM, 1994
- [33]. Bate, J., Loyal, A. B. and Reilly, W. S. An architecture for action, emotion, and social behavior. Technical Report CMU-CS-92-144[R], School of Computer Science, CMU, Pittsburgh, 1992
- [34]. Shoham, Y. Agent-Oriented Programming[J]. Artificial Intelligence, 1993
- [35]. Guttman. R. H, Moukas. A. G, Pattie. M. Agent-mediated Electronic Commerce: A Survey. Knowledge Engineering Review. 998(13)
- [36]. Rosenschein. J. S, Zlotkin. Rules of Encounter: Design Conventions for Automated Negotiation Among Computers. Cambridge MA: MIT Press. 1994
- [37]. Kraus. S, Sycara. K, Evenchik. A. Reaching Agreements through Argumentation: A Logical Model and Implementation. Artificial intelligence. 1998
- [38]. Electronic commerce homepage. <http://ec.fed.gov>

- [39]. B2B Shopping Agent. Exchang Agent. <http://www.b2bexchangeAgent.com>
- [40]. Liu Jin-de, Zhang Yun-yong. Overview of an available mobile Agent system(Aglet)[J]. Computer Application. 2001(21)
- [41]. Zhang Yun-yong, liu Jin-de. Study of interoperability of mobile Agent[J]. Computer Science. 2001(28)
- [42]. Zhang Yun-yong, Agent oriented software engineering[R]. Phd. Research Report of Computer Science & Engineering College of UESTC. 2000(8)
- [43]. S. arit Kraus. Automated Negotiation and decision making in Multiagent Environments[J] ACAI 2001
- [44]. Martin beer. Negotiation in multi-Agent Systems[C]. ULMAS 98
- [45]. N. R. Jennings. Automated Negotiation: Prospects, Methods and Challenges[J]. journal of group decision and negotiation, 2000

致 谢

在论文完成之际，衷心地 toward 所有支持过我和帮助过我的师长、家人和朋友表示感谢。

首先要感谢我的导师**陈德人**教授。感谢他在我读研期间对我的关心和教诲。陈老师严谨、认真、务实的治学态度，丰富渊博的知识，为人师表的风范，勤奋踏实的工作作风，宽宏谦虚的人品和甘为人梯的高尚品德都给我留下了深刻的印象，使我读研期间来获益匪浅，并将在今后的工作学习中不断激励我前进。

其次感谢 IBM 技术中心**施敏华**老师，他对我的毕业设计做了耐心的指导。由于他的帮助，使我很快的选定了自己的研究方向。施老师的热心、诚恳和富于爱心，使我永远铭记在心。

还要感谢**郑小林**老师，郑老师孜孜不倦的工作作风、谦逊善良的人品，是我学习的榜样。他不仅在论文上给了我许多研究性的指导，而且给我解决了许多实质性的难题。祝他工作顺利，生活幸福！

感谢**鲍力成**同学，他扎实的基础，谦虚的态度，乐于助人的品格，使永远不能忘怀。他在平时和论文完成期间给了我许多实质性的建议。祝他在新的工作岗位再创辉煌！

感谢曾经和正在一起学习工作的博士生**胡华梁、黄昌勤、肖颖、周涛**，硕士生**田超、彭锦、赵俊、周迎风、鲁琳**等同学，和他们在一起的经历和感受，是我巨大的财富，同时在我的心中留下了美好的回忆。

感谢我的父母，姐姐、姐夫和外甥。他们不仅给了我生命，而且给了我勇往直前的信心、动力和精神支柱。虽然他们不在我身边，但是他们的关心、爱护、鼓励和支持永远陪伴我。在此，我祝福他们身体健康，万事如意！

感谢所有关心和帮助过我的人们。祝好人一生平安！

最后，谨向百忙之中抽出宝贵时间评审本论文的专家致以最诚挚的谢意！

孙金志

2005.6 浙大求是园

