

对等计算中的基于多移动 Agent 的协作联盟机制

徐小龙^① 王汝传^{①②}

^①(南京邮电学院计算机科学与技术系 南京 210003)

^②(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘要: 对等计算技术为 Internet 这样开放式的、动态变化的网络环境里各节点间的协同工作, 进行复杂的分布式业务处理提供了运行基础设施。但是 P2P 环境中各节点随机加入和退出网络的、工作承担者事先不确定等特点也为工作的顺利完成带来了困难。该文提出一种对等计算环境中基于多移动 Agent 的柔性的、动态的协作联盟机制, 适合部署于复杂、多变的应用环境之中, 以克服对等节点协作的困难。主要思想是在某一需求目标的驱动下, 主导节点建立项目, 经过工作主体遴选, 形成暂时的、优化的协作联盟, 在复杂的处理流程的规范下, 基于合作承诺, 联合完成项目设定的任务。该文还将该机制应用于构建虚拟的敏捷企业供应链管理原型系统, 以此证明了该机制的实用性价值。

关键词: 对等计算; 协作联盟; 多移动 Agent; 供应链

中图分类号: TP393.06

文献标识码: A

文章编号: 1009-5896(2007)02-0345-05

The Collaboration Alliance Mechanism of P2P Based on Mobile Multi-agent Technology

Xu Xiao-long Wang Ru-chuan

(Dept. of Computer Science and Technology, Nanjing Univ. of Posts and Telecommunications, Nanjing 210003, China)

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

Abstract: Peer to Peer (P2P) computing technology provides the basic operation facility for collaborations among nodes of open and dynamic network (esp. Internet) and complicated distributed business processing. However, it is difficult to accomplish such works successfully because of some inherent facts of P2P environment, such as peers could randomly join in or quit the P2P network, which make it hard to choose appropriate task undertakers before the project is constructed. In this paper, a flexible and dynamic collaboration alliance mechanism of P2P based on mobile multi-agent technology is proposed. The collaboration alliance mechanism can be deployed in complex and changeable environments to overcome the difficult problem of peers' collaboration, which is working as the following ways: driven by one demand, the leader peer constructs the project; after the work undertakers are all selected, the temporary and optimized collaboration alliance is formed; according to the workflow, the leader peer and other work peers collaborate to accomplish the tasks of the project based on their commitments. The mechanism is also utilized to build the agile Supply Chain Management (SCM) system to verify its practical value.

Key words: Peer to Peer computing; Collaboration alliance; Mobile multi-agent; Supply chain

1 引言

对等计算(Peer to Peer computing, P2P)技术为加入 Internet 的各种资源的使用主体和提供主体提供了非中心化的、自组织的、且所有的或大部分联系是对称的分布式环境。P2P 环境下每个参与的结点既可以成为服务的提供者, 也可同时成为请求服务的用户^[1]。但是 P2P 技术的价值不仅仅在

于它在广域的范围实现了对数据信息、存储空间、计算能力、功能组件、通信资源的充分利用, 还在于它为 Internet 这样开放式的、动态变化的网络环境里各主体间的协同工作, 解决复杂的业务问题提供了运行基础设施。

一种典型的应用就是基于 Internet 的企业间(Business-to-Business, B2B)的开放的、分布式业务协作行为的实现: 不同的企业因某个具体的市场需求而形成暂时的联盟, 按照一定的工作流程完成任务, 获得利益并随之解体, 等待下一次商机, 再寻求新的组合。合作过程中, 企业的角色不同而地位对等。与企业内的相对较为固定的工作流程不同, 这种分布式业务协作行为是临时的、灵活的、按需定制的, P2P 环境中各节点随机加入和退出网络, 工作承担者事先不确

2005-05-30 收到, 2005-10-17 改回

国家自然科学基金(6057314, 70271050), 江苏省自然科学基金(BK2005146), 江苏省自然科学基金预研项目(BK2004218), 江苏省高技术研究计划(BG2004004, BG2005038)和江苏省计算机信息处理技术重点实验室基金(kjs050001)资助课题

定, 这些特点也为工作的顺利完成带来了困难。

本文的贡献就在于: 在 P2P 的架构下提出一种协作联盟机制, 这种群体协同机制是在复杂、多变的应用环境和 Internet 平台上, 在某一需求目标的驱动下, 建立项目, 经过工作主体遴选, 形成优化的协作联盟, 将个体的意图转变为联盟群体的共同意图, 在复杂的处理流程的规范下, 基于合作承诺, 联合完成项目设定的任务。这种协作联盟是柔性的、可根据需要动态扩展, 并可充分利用 Internet 上丰富的服务资源, 由此带来整体效益的提升。

P2P环境的不确定性与动态性给该协作联盟机制的实现带来了困难。本文的另外一个贡献就在于引入了多移动 Agent(Mobile Multi-Agent, MMA)技术, 利用其自主性、社会性、移动性等特点^[3], 在P2P环境中的每个对等节点上部署协作模块, 利用Agent通信语言(Agent Communication Language, ACL)实现Agent之间的协调工作, 从而在P2P环境下实现高效的协作联盟系统。

2 对等计算中的协作联盟

2.1 协作联盟建立过程

构成协作联盟的目的, 是由于单一的主体无法或很难完成某一需求支配下的所有任务, 因此需要由多个主体组成暂时的联盟, 按照一定的工作流程分别完成其中某一项或几项任务。这里需要解决的关键问题是, 联盟如何构成以及群体如何协作。

在基于 Internet 的 P2P 的环境中, 存在着提供不同的服务多个对等节点(Peer), 为实现对服务的合理利用, 需要设立注册/认证节点(Register/Authentication Peer, RAP), 并形成服务列表, 从而有效管理各对等节点身份和服务信息(服务名, 服务类型, 输入要求, 输出信息); 受用户(Customer)需求支配的任务发起者就成为了主导节点(Leader Peer, LP); 承担具体任务的节点就是工作节点(Worker Peer, WP)。协作联盟建立过程如下:

(1) User $\xrightarrow{\text{business}}$ LP, LP 接到来自用户的业务需求信息。

(2) LP 按需、按统一规定服务类型分解工作, 将工作分解为任务集合:

$$\text{Tasks} = \{t_0, t_1, t_2, \dots, t_i, \dots, t_m\}, \quad t_i \text{ 表示其中某一任务 } i。$$

(3) LP 根据任务的划分确定工作流程结构, 任务的划分有按步骤分解、按功能分解和按数据分解这 3 种策略。典型的工作流程(Procedure)如图 1 所示。

如图 1 所示, 任务之间的关系有 3 种: 严格的前向(forward)执行关系, 如一个任务的执行结果是下一个任务的

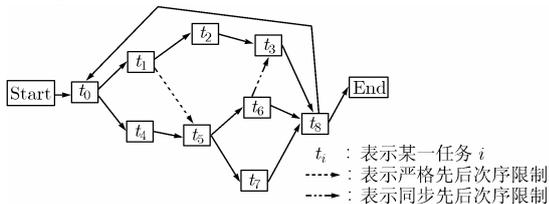


图 1 典型的工作流程

输入前提(如图中的 t_2 和 t_3); 严格先后次序限制关系, 即一个任务的执行必须在另一个任务执行完才允许开始(如图中的 t_1 和 t_5); 同步先后次序限制关系, 即一个任务的执行必须在另一个任务开始后允许开始(如图中的 t_6 和 t_3); 循环(loop)执行关系, 如图中的 t_8 和 t_0 。

(4) LP 依据待解决的问题的需求和目标, 确立解决该问题的项目。

定义 1 项目(Project)被定义为如下的六元式:

Project = (Demand, Goal, Tasks, Procedure, Rules, Status) 其中Demand为来自用户的需求描述; Goal为根据需求所要最终达到的目标, 或是拟实现的结果; Tasks为实现目标, 该项目必须完成的任务集合; Procedure为该项目在进行过程中, 各任务组成的工作流程; Rules为在该项目的生命周期里, 任务承担者所必须遵循的规则集合, 具体规则与问题的需求和目标以及任务的划分策略相关: $\text{Rules} = \{r_0, r_1, r_2, \dots, r_i, \dots, r_n\}$, r_i 表示执行任务*i*所必须遵循的规则; Status为项目当前所处的运行状态:

Status \rightarrow ready | running | suspended | restored | accomplished | failed

(5) LP从RAP处获得的对等节点身份和所提供的服务信息, 直接向所有对应的节点($P_1|P_2|\dots|P_m$)进行招标。利用了合同网(Contract-net)^[4]的思想, 对项目每个任务 t_i , LP都制定明确的标书从而实现P2P系统中各对应节点进行招标, 标书的差别数目和“任务集”中的任务数相等。下面给出标书的定义:

定义 2 标书(Task Announcement, TA)被定义为如下的五元式:

$$\text{TA} = (\text{Initiator, Task, Requirement, Period, Budget})$$

其中 Initiator 为任务的发起者, 即 LP; Task 为任务的发起者希望 WP 执行的任务或提供的服务; Requirement 为包括与任务相关的 WP 需要遵守的规则、完成任务需要达到的最低要求、申请任务的失效期限, 完成任务最后期限; Period 为限定了任务开始的最早和最晚时间, 以及任务结束最早和最晚的时间; Budget 为完成该任务, LP 支付给完成该任务的 WP 的报酬预算。

(6) LP $\xrightarrow{\text{TA}_1|\text{TA}_2|\dots|\text{TA}_n}$ $P_1|P_2|\dots|P_m$, LP 利用标书向对应的 Peer 进行招标。

(7) 收到标书的 Peer, 对标书的来源进行验证, 确信其来自合法的、可信任的发起者, 就对标书的内容、自身的能力和利益进行评估, 以决定是否参加对执行该任务的竞标。如决定参与则首先要制定申请书。下面给出申请书的定义:

定义 3 申请书(Bid Application, BA)被定义为如下的五元式:

$$\text{BA} = (\text{Candidate, Task, Description, Need-time, Need-cost})$$

其中 Candidate 为标识了愿意执行该任务的候选节点的经过认证的身份; Task 为发起者标书中指定的任务或服务考过;

Description 为描述任务的发起者、标书收到时间、可以执行该任务的时间段; Need-time 为该节点执行该任务或提供该服务的所需花费时间; Need-cost 为预计完成任务所需的费用。

(8) $P_1|P_2|\dots|P_m \xrightarrow{BA_1|BA_2|\dots|BA_n} LP$, 愿意执行该任务的候选节点将申请书发给 LP, 参与竞标。

(9) LP 收到任务申请书, 利用既定的策略、算法对任务进行评估, 以便从中选择最适合执行该任务的 WP。为了避免复杂计算, 采用简洁的算法, 以便快速建立合作关系。可采用的一种选择公式如下:

$$SELECT = \{ i \mid i \in BidList \wedge \min(W_{cost} * (COST_i / COST) + W_{time} * (TIME_i / TIME)) \}$$

BidList 是LP收到的竞标标书的列表; i 是标书列表中的一项; W_{cost} 是Cost的权值; W_{time} 是Time的权值; $COST_i$ 是候选节点预计完成任务所需的费用, 它等于申请书BA中Need-cost的值; COST是LP在分配任务声明标书中对任务费用的预算, 它等于标书TA中Budget的值; $TIME_i$ 是候选节点预计完成任务所需的时间, 它等于BA中Need-time的值; TIME是LP在分配任务标书中要求完成任务的最后期限。该公式可通过调整 W_{cost} 和 W_{time} 的大小以决定任务的时间优先或费用优先。

(10) LP 根据选择的结果, 制定相应的合约书, 下面给出合约书的定义:

定义 3 合约书(Agreement, Ag)被定义为如下的六元式:

$Ag = (Initiator, Worker, Task, Procedure, Time, Payment)$ 其中 Initiator 为任务的发起者, 即 LP; Worker 为任务的承担者, 即 WP; Task 为指定了具体的执行任务, 与标书和申请书一致; Procedure 为该项目在进行过程中, 各任务组成的工作流程, 由此确定本任务的输入来源、输出的对象以及其它制约关系(如先后关系、最后期限等); Time 为该节点执行该任务或提供该服务的所需花费时间; Payment 为完成该任务任务发起者支付给任务承担者的报酬。

(11) $LP \xrightarrow{Ag_1|Ag_2|\dots|Ag_r} WP_1|WP_2|\dots|WP_s$, LP 将合约书发给各对应的 WP。

(12) WP 向 LP 发回确认信息, 缔定合约, 全部合约缔定后, 由此建立了完成该项目的协作联盟。

2.2 任务转包与任务重起

当协作联盟建立好之后, 由 LP 启动项目的进行, 项目进入运行状态(running)。各 WP 按工作流程依次执行自己的任务。该协作联盟是在运行过程中也允许根据当前情况动态变化。典型的两种情况是任务转包与任务重起。

当某 WP 因为外界和自身情况的动态变化, 感觉到所安排的任务超过了它的当前能力和资源, 那么可以进一步划分任务, 并向其它节点招标。这时它就扮演了前面 LP 所扮演

的任务发起者的角色, 这样就形成了一个分层任务结构: 每个节点可以同时扮演任务发起者、任务申请者和工作节点的角色。

利用这种转包任务的特点我们可以在P2P环境中项目的执行过程中动态调整活动的执行者。以往的工作流系统通常在工作流执行前就为活动分配好执行者, 当发生执行者无法按时完成任务等异常的时候, 还需要由工作流机或者管理人员处理异常。这不仅影响流程执行的进度而且加重了工作流机或者管理人员的负担^[5]。同时, 这种方式也显然很难适应动态变化的P2P执行环境。而采用任务转包的方式, 签订了合约的WP负责管理并协调活动的执行。它会根据实际执行的情况, 预先采取一些行动以确保活动的顺利完成, 例如当WP预感到将不能按时完成任务的时候, 它就把任务分解成子任务, 为子任务寻找执行任务的节点, 这样就在协作联盟中又形成了一个动态的协作联盟, 而承包初始任务的WP就成为这个协作联盟的LP, 如图2所示。协作联盟的建立方法与上面所述的过程相同。

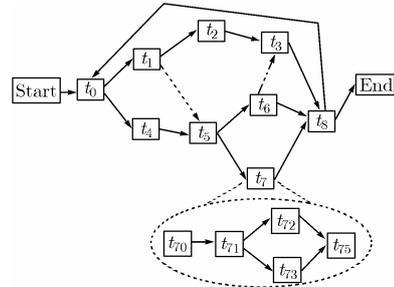


图2 任务分解并转包的模型

协作联盟建立以后, LP 负责启动项目和协调各 WP, 实施群体协同机制, 监视/维护联盟的运行状况。当 WP 在执行过程中, 发现无法完成任务, 也不能实现任务转包, 其就向 LP 发送任务失败信息, LP 重新进行招标。招标范围发来任务申请书的候选节点中选择, 从中选择下一个最适合执行该任务的 WP。调整项目运行流程, 并通知与原任务承担者直接相关的工作节点。LP 并将各个任务的结果进行合成, 项目完成或是失败时, 协作联盟解体。

3 基于多移动 Agent 技术构建协作联盟

在P2P环境中建立协作联盟可以有效利用Internet上丰富的服务资源, 实现了多个对等节点之间的协作, 提高了整体的效益。然而, P2P环境的动态性和不确定特征都为实现协作联盟机制带来了困难。

在系统实现的过程中, 我们引入了多移动 Agent 技术。多移动 Agent 技术是将多 Agent(Multi-agent)和移动 Agent(Mobile agent)技术集成在一起。移动 Agent 是一个能携带任务或待求解的问题, 在异构网络中自主地从一台主机迁移到另一台主机, 并可与其他 Agent 或资源交互的主动程序。在开放的分布式网络环境的每个节点上都部署有相对独立的智能 Agent, 它们具有自主性, 并能根据具有的知识信念以及周围发生的事件进行推理、规划与通信。多 Agent 之间

彼此在逻辑上相互独立，通过共享知识、任务和中间结果，协同在工作中形成问题的解决方案。因此多移动 Agent 技术适用于在地理上或逻辑上分布、自主或异构的结点间提供服务；在高度动态的环境下，能对多变的环境做出响应或自适应；能自主地处理失效或冲突，以进行再调度、再计划或资源再分配。除此之外，Agent 还能进行长期计划驱动的行为和短期实时响应行为，在复杂的或安全性很重要的应用程序中，还保证适宜的反映和应答时间。由此可以看出，将多移动 Agent 技术用于实现 P2P 环境下的协作联盟系统是适宜的。

因此，我们将协作联盟内部各对等节点的关系转化为 Multi-agent 系统中 Agent 之间的协作关系，利用 Agent 的自主性和社会性等智能特性降低了协作联盟实现的难度。同时可通过移动 Agent 分发任务和传递合约书等交互文件。其中需要重点解决的就是在动态、复杂的应用环境中，存在于异地对等节点上的 Agent 如何解决信息交互和协调的社会性计算实现问题，这主要依赖于 Agent 间的通信。

FIPA(FOUNDATION for Intelligent Physical Agent)ACL 是近几年发展起来的 Agent 通信语言，目前 FIPA ACL 主要包括 3 类协议：基本动作请求协议、合同网(Contract-net)协议、拍卖(Auction)协议，分别反映了 3 种不同的基本社会交互模式。基本动作请求协议描述了任务委托者与受托者之间的交互；合同网协议用于协调 Agent 间的任务分配；拍卖协议则刻画了 Agent 间的价格等参数的协商过程^[6,7]。本文在任务发起者与其它参与节点的 Agent 间进行任务分配活动时采用合同网协议，利用 Agent 统一建模语言(AUML)^[8]，如图 3 所示。

图 3 中，首先由任务发起者 Initiator 向参与者 Participant 进行招标(Call For Propose, CFP)消息，当收到 CFP 后，Participant 根据标书内容和自身能力与利益进行评估得出正确意愿，从而作出适宜响应，这在申请任务的最后期限(Deadline)以前有效。Participant 可能拒绝(Refuse)联合请求，也可能同意，并递交申请(Propose)。前一种情况使协议终止。当 Initiator 收到 Participant 的 Propose 消息后，对各个 Participant 进行横向比较评估，并选出适合完成此项任务 Participant，对其发出接收申请(Accept-propose)的

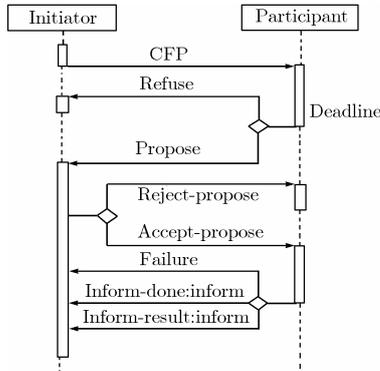


图 3 合同网协议的 AUML 表示

信息，内容为合约书；对其它 Participant 发出拒绝申请

(Reject-propose)的信息。此时在 Initiator 和被接收申请的 Participant 之间形成合作关系。若 Participant 意识到任务已完成(done)，或者任务已失败(Failure)，都有责任将结果通知(inform) Initiator。

4 原型应用系统

为了验证本文提出的基于多移动 Agent 的对等计算协作联盟机制，我们在校园网的 Intranet 的环境中模拟出 Internet 的平台上，构建虚拟的敏捷企业供应链管理(Supply Chain Management, SCM)原型系统。所谓供应链是围绕核心企业的，通过对物流、信息流和资金流的控制，从采购原材料开始，制成中间产品以及最终产品，最后由销售网络把产品送到，将供应商、制造商、分销商、零售商直到最终用户连成一个整体的功能网络^[9]；而敏捷供应链则将企业内部和外部形成具有快速响应能力的动态联盟，并具有可重构、可重用和可扩充的能力，各不同企业间因某个商机而临时组成联盟，任务完成又立刻解体。

在这种供应链应用中，各企业角色不同，而地位对等，而企业间的联系又是动态的、暂时的和不固定的。如果要采用传统的工作流机制实现这种供应链管理系统，则需要事先确定供应链中各任务承担者并部署各应用程序，因此很难适应上述情况，达到“敏捷”的目标，而将本文提出的基于多移动 Agent 的对等计算协作联盟机制应用于敏捷供应链中，恰好可以满足需求：由于在协作联盟的建立过程中，各协作主体是经过遴选后自愿加入联盟的，该协作联盟是优化的；协作联盟中应用了任务转包与任务重起机制，从而使应用系统更为灵活和可靠，因此通常任务发起者和其它各协作主体都可以获得自身的预期效益。

在我们构建的原型系统中，我们将各节点限定为供应链中不同的角色类型，并安装相应的应用系统。

该系统运行于采用 JXTA 技术实现的 P2P 环境中。JXTA 是 SUN 为了构建 P2P 网络而制订的一组协议，是处理构建 P2P 网络所碰到的问题的解决方法，JXTA 由对等点发现协议(Peer Discovery Protocol, PDP)和对等信息协议(Peer Information Protocol, PIP)等六个协议组成，这些协议是专为特定的、分布式的、对等的网络计算而设计的。使用这些协议，对等节点可以互相合作来建立自我组织、自我管理的对等组。协作联盟系统是由多移动 Agent 来实现。各对等节点上的系统架构如图 4 所示。其中，JXTA 核心层(JXTA core)封装了服务的核心功能，包括 Peer 对等组(Peer groups), Peer 管道(Peer pipes), Peer 监视(Peer monitoring)和相关的的原语等；JXTA 服务层(JXTA services)提供了访问 JXTA 协议的接口；协作联盟(Collaboration alliance)系统由基于 Java 的 Mobile multi-agent 技术来实现，节点间协作工作由 Multi-agent 完成，Mobile agent 负责分发任务

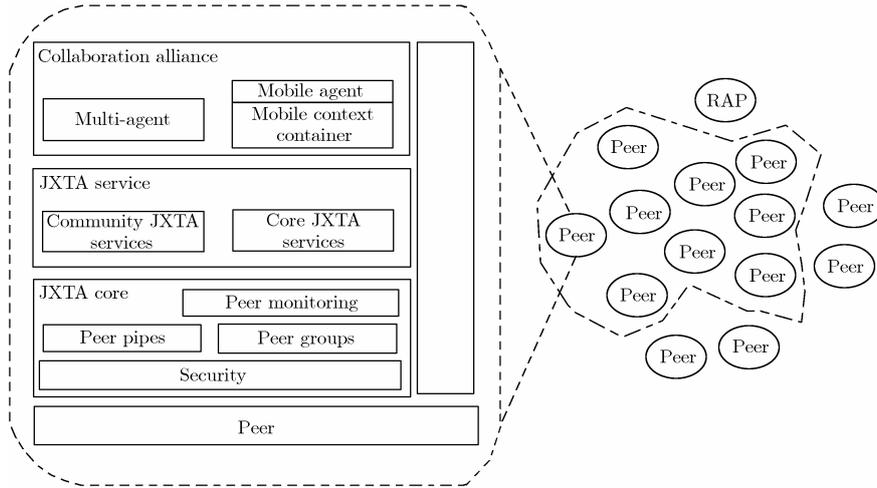


图4 对等节点上的系统架构

和传递 XML 格式的合约书等交互文件, 其运行于当地的移动上下文容器(Mobile context container)中; 供应链管理系统具体任务由相应的本地的应用程序(Local applications)完成。

5 结束语

针对 P2P 技术目前还主要集中应用于简单的文件传输、共享, 或是进行相对固定的科学计算的现状, 为了利用 P2P 潜在的分布式复杂业务处理能力, 本文提出一种 P2P 环境中各节点协作联盟机制, 并利用多移动 Agent 技术予以实施。为了验证其可用性, 我们构建了虚拟的敏捷企业供应链管理原型系统, 将本文提出的协作联盟机制应用到该系统中, 从而实现了供应链中各环节之间的协调工作。

进一步的研究主要是如何增强系统的鲁棒性, 如何在协作联盟建立以后, 其中某工作节点任务失败或系统瘫痪的情况下, 如何及时地发现并实现任务重起, 从而避免因某一节点失效而导致整个项目的失败或延误。目前采用的方法是依靠工作节点的主动报告和主导节点的轮询, 这两种方法不够可靠和及时, 需要进一步的探讨。

参考文献

[1] Moore D and Hebler J. Peer-to-Peer: Building Secure, Scalable, and Manageable Networks. Osborne: McGraw Hill, 2002: 5-16.

[2] Zeng L Z, Ngu A, Benatallah B, and Dell M. An agent-based approach for supporting cross-enterprise workflows. In: Orłowska M, Roddick J, eds. Proc. of the Australasian Database Conf. Gold Coast: IEEE Press, 2001: 123-130.

[3] Wooldridge M. An Introduction to Multi-agent Systems, Chichester, England: John Wiley & Sons, 2002: 1-8, 10-31.

[4] Sandholm T. An implementation of the contract net protocol based on marginal cost calculations [A]. In: Communication in Multi-agent Systems: Agent Communication Languages and Conversation Policies [C], Melbourne, Australia, July 2003: 51-97.

[5] Fischer L. The Workflow Handbook 2002, The Workflow Management Coalition, Florida: Future Strategies Inc., Lighthouse Point, 2002: 19-38.

[6] FIFA. FIPA Agent communication language specifications. 2002. <http://www.fipa.org/repository/aclspecs.html>

[7] Sheng Q J, Zhao Z K, and Liu S H. A teamwork protocol for multi-agent. *Journal of Software*, 2004, 15(5): 689-696.

[8] Bauer B, Müller J P, and Odell J R E. Agent UML: A formalism for specifying multiagent interaction. In: Ciancarini P, Wooldridge M, eds. Agent-Oriented Software Engineering. Heidelberg: Springer-Verlag, 2001: 91-103.

[9] Chai Y T, Li F Y, and Ren S J, et al. Agile supply chain management system architecture. *Journal of Tsinghua University: Science and Technology*, 2000, 40(3): 79-82.

徐小龙: 男, 1977 年生, 讲师, 博士生, 研究方向为计算机软件、计算机网络、信息安全、移动代理和移动数据库等。

王汝传: 男, 1943 年生, 教授, 博士生导师, 主要研究方向为计算机软件、计算机网络和网络、信息安全、移动代理和虚拟现实技术等。