

面向角色的多 agent workflow 模型研究

赵卫东¹, 黄丽华²

(1. 复旦大学软件学院, 上海 200433; 2. 复旦大学管理学院, 上海 200433)

摘要: 目前的工作流建模工具大多是面向活动、面向产品、面向目标或面向决策的, 没有强调工作流是多个角色主体协作的过程, 工作流的绩效最终依赖员工及其角色的作用. 从角色及其合作的角度, 分析了面向角色的工作流模型及其表示方法, 为工作流的管理提供新的思路. 在此基础上, 研究了角色、agent 和工作流的关系, 给出了一种面向角色的工作流多 agent 管理系统模型和原型.

关键词: 角色; 智能体; 工作流模型; 工作流管理系统

中图分类号: F273 **文献标识码:** A **文章编号:** 1007 - 9807(2004)02 - 0055 - 08

0 引言

流程建模是流程管理的基础, 目前已出现了许多建模分析工具, 但大多数的流程模型局限于用活动及其逻辑关系描述, 对组织人员及其角色缺乏重视. 实际上, 流程的绩效最终依赖于员工和角色的作用, 而不是产品和流程^[1].

传统的流程模型多是基于活动、面向产品、面向目标或面向决策的, 即分别从流程的主体行为、行为结果、目的等角度描述. 信息时代的流程是面向客户的, 流程的参与者之间存在着任务分担、结果共享的相对客户关系. 企业是通过流程提供对顾客有价值的产品和服务, 顾客对产品和服务的满意度是衡量一个流程绩效较重要的指标. 从建模的对象看, 企业流程模型应集中在员工或组织单元的交互行为上^[2]. 面向角色的方法, 详细说明并分析参与流程各角色的作用, 强调执行流程的参与者角色及其关系, 表达了多个角色协同合作的过程.

本文从角色的观点研究工作流, 分析了面向角色的工作流模型及其表示方法, 为工作流的管理提供新的思路. 其次, 在分析角色、agent 和流程关系的基础上, 给出了一种面向角色的多 agent 工作流管理系统模型和原型.

1 角色基础

角色被理解为接受信息、加工信息和发送信息的抽象对象. 角色概念曾用于管理信息系统的自动生成工具的研究, 其中角色指管理信息系统的基本单元^[3]. 角色理论认为, 角色是责任和权利的统一体. 其中责任规定了角色的行为规范和约束^[4], 也就是说, 角色是某一类对象结构、性质、行为、职能等方面所共有的特征集合. 它具有目标、能力、责任 (obligation)、许可、约束和协议等对象多方面本质特征的综合反映, 能够作为事物分类的合理标准. 流程的活动是由各种参与者完成的. 就参与者和角色的关系而言, 角色的概念是根据参与者的技能、能力等要素, 把参与者分组抽象而来的. 一个参与者可能担任多种角色.

在流程中, 角色是一个主动的、相对独立的抽象单元, 具有一定目标, 能按顺序完成一系列操作. 角色有活动、资源和状态等元素, 其中活动是担任此角色的参与者执行的任务, 资源是角色活动需要的设备、原材料和信息等. 角色之间通过信息联系, 信息构成了事件的前提条件, 活动由事件触发, 并改变角色的状态. 在不同的时刻, 角色可处于不同的状态. 角色的性质可用状态关系、角色扮演关系和活动关系等概念表示.

收稿日期: 2002 - 03 - 26; 修订日期: 2003 - 08 - 15.

基金项目: 国家自然科学基金资助项目 (70301004); 国家 863 高技术研究发展资助项目 (2001AA414120).

作者简介: 赵卫东 (1971 -), 男, 江苏徐州人, 博士, 副教授.

定义 1 角色扮演关系

ACTOR = Performer, Role, f

式中: f: Performer → 2^Role; Role 表示角色集合; Performer 表示参与者集合. 某个参与者的角色扮演集 role ⊆ Role.

角色扮演关系描述了参与者在组成流程的活动中担任角色的情况.

定义 2 状态关系

STATE = Role, Information, Resource, Time, State, g

式中: Information 表示输入信息集合; Resource 表示资源集合; State 表示状态集合; Time 表示时刻集合; 映射 g: Role × Information × Resource × Time → 2^State.

状态关系反映了角色使用信息和资源而在不同时刻处于不同状态.

定义 3 活动关系

ACTIVITY = Role, Event, Activity, h

式中: Event 表示事件集合; Activity 表示活动集合; 映射 h: Role × Event → 2^Activity.

活动关系表示角色的活动由一定的事件激发.

定义 4 面向角色的流程

P = { Role, Relation }

其中: Role 表示角色空间集; Relation 表示角色间的协作关系.

从活动的角度讲, 流程由活动及其逻辑关系组成. 这种观点重在对活动关系的限定而不是参与者之间的承诺 (commitment). 而从满足顾客的要求分析, 流程的基本组成元素是角色 (单员工角色和复合角色) 及其协调关系, 每一角色有输入与输出参数集. 复杂的流程是由各种各样的角色智能体组成, 它们具有不同的职能、能力和任务, 相互协作完成流程的目标. 活动及其关系只是角色及其合作的外在表现. 流程本质上是角色的集合, 即参与者集合及其交互. 图 1 是角色的示意图.

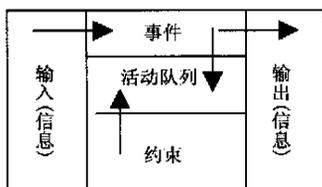


图 1 角色 Fig. 1 Role

角色也是个层次概念, 其粒度是和其目标分解的粒度对应的. 根据流程跨越组织的范围, 小到— 个员工、工作小组, 大到供应链中的企业, 都担负— 定的角色. 如图 2 (图中箭头表示可分解), 从角色的观点来说, 角色的确定依赖企业目标的分解, 高层角色负责高层目标. 目标的分解可采用情景分析 (scenarios analysis)、自下而上、自上而下或者两者的结合等. 宏观上看, 企业作为一个系统, 为其所在供应链上的客户提供产品和服务, 或作为客户接受其他服务者的服务. 它可以视为高抽象层次的角色. 另一方面, 企业的目标由多个核心流程完成. 从企业内部看, 每个流程可以由跨职能的工作小组完成. 流程之间往往存在前驱后继关系, 如某公司的采购 workflow, 包括从制定物料采购计划开始到物料入库及付款为止的所有活动, 涉及三个子流程: 制定采购计划流程 签定采购合同流程 采购合同控制流程. 它们由不同的角色完成, 存在客户和服务关系. 同样, 每个流程也包含不同的角色. 在实施 BPR 的企业中, 流程的任务分配给工作小组完成. 此时的角色以团队及其成员的形式出现, 它们之间存在协作关系. 从上可见角色的客户— 服务者关系是相对的, 通过流程客户— 服务者链的各个环节, 实现企业目标和价值.

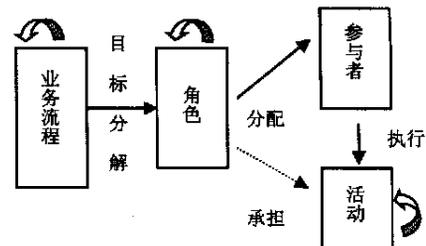


图 2 面向角色的 workflow Fig. 2 Role-oriented workflow model

2 面向角色的 workflow 模型

workflow 是流程的自动化, 流程的建模方式对 workflow 的管理有很大的影响.

2.1 基于活动的 workflow 模型的不足

以往大多数流程模型都是按活动建模, 即将流程视为完成某一目标而进行的一系列逻辑相关的活动的有序集合. 这种基于活动的描述方法在实际应用中还难以应付流程的动态性, 主要存在

以下缺点:

(1) 把各参与者执行的活动交错分布在一个模型中, 强调活动的关系却忽略了其主体角色的交互. 这样, 模型不但复杂, 而且缺乏对某个角色所执行的活动整体描述.

(2) 在用多 agent 研究 workflow 模型时, 往往把活动抽象为 agent, 活动之间的关系抽象为 agent 之间的交互^[5]. 这种做法不免牵强. 由于活动是对角色动态特性的描述, 流程的主体应该是具有能力、目标、责任等特性的角色, 其实质是面向角色的, 把流程看作角色的交互. 另一方面, agent 是一类在特定环境下能感知环境, 并能自治地运行以代表其使用者实现目标的计算实体, 具有交互性、自主性和主动性. 角色和 agent 互相对应, 用多 agent 系统仿真流程的运行显得更自然.

(3) 角色是流程最活跃的因素. 研究表明, workflow 的变化经常可以找到角色方面的原因. 面对变化的环境, workflow 的动态性日显重要. 角色柔性也是提高 workflow 灵活性的主要方法. 现有的 workflow 系统仍然存在着适用范围窄, 灵活性差, 扩展能力不足以及对意外的情况不能及时处理等问题, 这与没有重视角色及其交互有关. 作为一个计算机支持协同工作 (CSCW) 系统, workflow 的自组织、自学习和协同工作能力等智能行为的研究最终依赖于角色理论的完善.

2.2 用角色描述 workflow

workflow 是为提高流程的工作效率而提出的. workflow 管理联盟 (WFMC) 将 workflow 定义为: 按照预定义的规则传递文档、信息或任务, 实现组织成员间的协调工作, 以达到预期的业务目标. 这里的组织成员担负相应的角色, 从上述定义可以看出, workflow 用角色的交互来刻画要比用活动链更容易理解, workflow 的核心是角色的交互. 目前的工作流模型从形式上来看, 可分为基于活动的工作流、基于通讯的工作流和结合两者优点的混合工作流^[6]. 基于通讯的工作流强调角色之间的交互, 把参与 workflow 的角色之间看作客户和服务者的关系, 这种客户—服务链构成的 workflow, 与基于活动的工作流相比, 更突出了顾客的观点和满足.

复杂业务流程的实施过程都可以通过目标的分解, 角色任务的分配及活动的执行完成. 角色之间通过目标分解和活动执行建立相互关系. 业务

目标是个层次型概念, 根据一定规则可以逐级分解, 直到原子目标 (参与者可承担的任务). 目标的实现依赖角色的活动, 活动和角色也是可分解的. 角色的分解和业务目标的分解对应, 并由参与者负责相应层次的活动. 角色和活动的层次性是和目标层次性相对应的. 高层目标可能涉及不同的角色, 角色负责成员的活动分配. 面向角色的 workflow 元模型如图 3 所示.

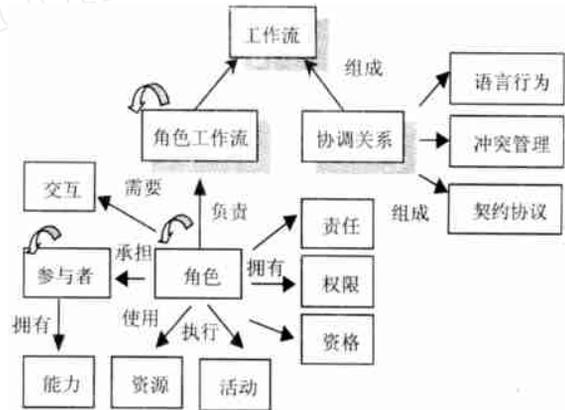


图 3 基于角色的 workflow 元模型

Fig. 3 Role-based workflow meta-model

企业的目标是由相互合作的角色参与的流程完成. 角色、参与者、流程和业务目标的关系如图 2 所示. 用角色描述 workflow, 其实质是由相对独立的角色根据流程规则交互的过程. 更确切地讲, workflow 是把业务目标分解, 然后由各级角色分配给相应的参与者通过活动完成的过程.

面向角色的 workflow 模型具有以下特点:

(1) 每个 workflow 具有特定的最终客户. 通过 workflow 客户—服务链的传递, 实现客户 (组织外部的或内部的) 的满意度.

(2) 每个 workflow 包括多个角色, 这些角色为了同一个 workflow 最终客户的满意进行合作, 提供客户所需要的产品或服务. 可以看出, 角色交互是 workflow 的纽带. 基于角色的 workflow 使传统基于活动的工作流转化为参与者之间意图、任务和目标、资源间的依赖关系.

(3) 基于角色的 workflow 模型把 workflow 的活动聚类成角色. 角色承担特定的活动或活动集. 这样避免了传统基于活动的工作流描述中单个角色承担的活动分布在 workflow 的多个环节中.

(4) 在基于活动的工作流模型中, 许多建模方法是从别的领域借用的, 由于缺乏统一的理论

指导,导致目前流程建模方法的多样性.协调理论^[7]、多智能体系统(multi-agent systems, MAS)理论的逐渐完善以及与 workflow 研究内容方面的重叠为基于角色的 workflow 建模提供了可靠的理论基础.

workflow 是特定时间产生特定输出的一系列客户—服务者关系.目标、资源、角色和 activity 是 workflow 的基本要素.把不同的要素分别作为 workflow 描述的中心,会得到不同的模型.在传统的工作流定义中,基于 activity 的 workflow 描述方法的确可以突出 workflow 的动态特性,而且 activity 的价值、时间等属性为企业价值链的分析提供了基础,但这种描述忽略了参与者的交互和 customer satisfaction.产品和服务最终是 workflow 的主体——担任不同角色的参与者创建的, activity 只是实现价值的途径.在 customer-oriented 的信息时代,企业必须放弃只注重产品生产规模、功能质量的思想,而将更好地满足 customer demand 和提高 customer satisfaction 放在首位.

workflow 中任何目标的具体实现过程是通过目标、任务分解、角色、资源分配及 activity 执行完成的.从 CSCW 的角度来说,workflow 更强调角色之间异步、异地的协作,可以将角色、目标和 activity 之间看成是一种层次映射关系来刻画现实中的 workflow 结构.而 workflow 可视为多个角色的交互集合,角色则对应成员 activity 集合,角色间通过消息传递的事件激发机制建立客户—服务链.从这个意义上讲,workflow 通常虽被描述成一种协同 activity 集合,但协同工作的主体基本单位是角色,角色能力和智能化才是达到 workflow 目标的保证,而角色间的依赖关系通过一组相关联的 activity 集体现.

workflow 中角色协作通常有两种类型:集体参与和交替执行.这两种类型适合不同粒度、不同层次的角色.集体参与是指某任务由一个角色不能独立实现,需要多个角色的共同参与.交替执行是指任务需要多个角色的轮流执行,每个角色的行为都和前驱角色的若干行为相关,包括数据关系、控制关系、资源关系和任务关系.

2.3 面向角色的 workflow 表示

面向角色的 workflow 目前出现了几种表示方法,它们多是从基于 activity 的 workflow 描述方法扩充而来,突出了角色之间的交互.总体上这些方法又可分为非形式化的图形表示和形式化的语言脚本

表示,前者直观,可作为后者的基础.如在基于通信的 workflow 建模方法中,workflow 分析的目标是提高 customer satisfaction.顾客和服务者之间的交互简化成多个相连的子 workflow 环(包括准备、协商、执行和接受几个步骤).多个 workflow 环组成完整的 workflow,这种协作图描述了相关角色之间的联系以及这些角色产生的操作或结果,但对各角色执行的 activity 表达不完整,不便于 workflow 的仿真分析.为此,文献[2]又将角色 activity 图(role activity diagram)加入角色交互模型,角色交互图能转化为角色 activity 图,并且后者又可转化为角色仿真模型(role simulation model).在统一建模语言 UML 表示法中,协作图也表示特定环境下一组相关对象之间的协作,以及为产生所要求的目标对象协作交换的消息.协作图的图形表示以对象为结点,结点之间的连线或箭头表示消息或关联.在角色协作图中,事件是 workflow 中角色 activity 的触发者,而前一个角色 activity 结束时产生的输出信息可能激发下一个角色,所以信息是客户—服务链的连接者.信息、事件和 activity 的上述关系可表示成企业的业务规则.此外,也可拓展面向角色的可执行的流程模型 RolEnact^[8]. RolEnact 是一种能生成可执行流程实例的建模语言,这种语言采用角色、角色的状态、activity 或事件等要素描述流程.角色通过执行 activity (除独立的功能 activity 外,还包括选择和交互 activity)改变自身状态,也可能改变其它角色的状态. RolEnact 的上述特点,很适合用来描述面向角色的 workflow.但 RolEnact 毕竟只是一个流程建模工具,它的基础在于对象技术常用的条件——活动模式,对角色的产生(由虚拟的角色产生)、高层角色内部的开发支持不足.

有关角色驱动的 workflow 形式化描述,最直接的方法是对比 workflow 联盟的基于 activity 的 workflow 描述语言,使之由基于 activity 转变为面向角色.根据前面的分析,从角色观点看,workflow 模型是由角色、角色 workflow 及其交互组成.下面用 EBNF 简要表示上述元素:

```

< Workflow Process Definition > ::=
WORKFLOW < process id >
< Workflow Process Definition Header > //参考
WfMC 的工作流过程定义语言
< Role Workflow List >

```



```

< Role List >
< Communication Information List >
< Start Condition >
END. WORKFLOW
< Role List > ::= ROLE < role id >
[DESCRIPTION < description > ]
[NAME < Name > ]
< Role Kind Information >
< Actors >
END. ROLE
< Role Kind Information > ::= PERSONAL |
TEAM
< Role Workflow List > ::=
[DESCRIPTION < description > ]
< Role >
< Workflow Process >
...
END. ROLEWORKFLOW
< Communication Information List > ::=
[DESCRIPTION < description > ]
< Sender >
< Receiver >
< Event >
< Speech Act >
END. COMMUNICATION INFORMATION

```

Workflow Process 是用活动描述的,可参考 WMC 的工作流过程定义语言标准,而语言行动 (speech act) 可参考行为会话 (状态变迁图) 来描述角色 agent 之间的协调。

3 角色驱动的多 agent workflow 管理模型

3.1 模型

根据上述角色的定义,可以发现角色和 agent 的特征表现了许多相似性。agent 可看作组织中担负某角色的智能实体,比较适合 workflow 角色的描述。workflow 的各级子目标是由不同层次的角色通过活动完成,作为角色的智能代理,agent 能够模拟或表现出被代理者具有的意识态度,如信念、愿望、意图、目标、承诺、责任等。

多个 agent 之间可以相互协作、相互服务,组成多 agent 系统,其求解问题的能力超过单个 agent。另一方面,作为多角色协作的群件系统,workflow 也属于分布式的协同工作研究范围。从信息处理的角度看,一个 workflow 是由多个角色通过协作完成某种任务或目标的。每个角色具有一定的自主性和目的性,通过角色从事的活动——输入和输出关系联系起来,这种以协作方式组成 workflow 的角色集合实质上是一种多智能体系统,所以 workflow 与多 agent 系统有着某种相似的特点,体现在多 agent 系统也能够模拟角色之间的协作。多 agent 系统应用于 workflow 的建模已引起了人们的关注^[5,9],但面向角色的 agent workflow 模型还是一个较新的课题,有望结合 MAS 理论克服传统基于活动的工作流不足。

Agent 的独立性、自主性和交互性等特性足以表现角色的行为和特征。Agent 之间发送消息通讯,消息会激发一定事件。这些事件会由 agent 的事件监测器捕捉,以便根据内部状态和外部事件决定和控制下一步的活动,激活相应的 ECA 规则。推理机和 ECA 规则库是 agent 自主性、主动性的核心。规则的执行会改变 agent 的状态,并且会向别的 agent 发送消息,产生新的事件。各 agent 如此往复,通过完成各自的子目标而促进 workflow 的任务完成。agent 的上述通讯方式,实质上对应了角色的协议 (protocols) 属性。

从以上分析可见,利用 agent 的分析和设计方法适合面向角色 workflow 的建模。有关角色的划分粒度,可根据角色所从事活动的复杂性、独立性等来确定。

参考 workflow 管理联盟的 workflow 管理系统体系结构,下面给出角色驱动的多 agent workflow 管理系统模型。这个模型主要由以下几个部分组成:流程建模模块,workflow 服务器,管理 agent,角色 agent,接口 agent 和参与者等,它们的功能如下:

(1) 流程建模和 workflow 服务器的主要任务包括确定 workflow 的目标,目标的分解,角色的定义,确定角色对应的 agent 类型并分配任务等。模型可能需要多次修改。目标的分配采用基于角色的路由方法 (role-based routing),即把完成子目标 (活动) 的责任直接授予角色,以增强 workflow 的柔性。上述过程实质上是面向角色的 workflow 分析和基于 agent 的工

作流设计和实现的统一。可见这种做法实现了 workflow 建模和 workflow 运行的连接, 比较容易利用 agent 技术完成 workflow 的设计。workflow 的建模和 workflow 服务器的运行需要专业人员完成。

(2) 管理 agent. 它和 workflow 服务器起到 workflow 引擎的作用。workflow 服务器需要解释流程定义, 协助管理 agent 控制 workflow 实例的执行和 workflow 中各种角色的协调等。它也负责角色 agent、接口 agent 和资源 agent 的创建、维护、调度和执行, 拥有角色 agent 和资源 agent 的信息和信念等。管理 agent 与角色 agent 之间具有良好的交互性, 能准确地接受和理解角色 agent 所传递的信息。从功能上看, 管理 agent 通过把分解的相关业务子目标分配给各角色 agent 起到协调角色 agent 的作用。角色 agent 拥有共同的目标, 一般来讲, 当角色之间有客户和服务关系时, 它们可以直接交互, 而在其它情况下如果需要(如异常、基于合同网的动态合作^[9]等), 则由管理 agent 负责角色 agent 之间的协调。角色的每次交互过程都由管理 agent 记录在 workflow 数据库中。管理 agent 根据角色 agent 提供的信息通过接口 agent 与 workflow 的实际参与者(包括 workflow 管理员)交互, 推进 workflow 实例的执行。当然专业的参与者可以直接和角色 agent 交互。管理 agent 主要负责角色 agent 的协作。它根据 workflow 服务器的控制信息, 完成相应的角色 agent 的调度和运行。

(3) 角色 agent. 角色是组织中具有完成特定活动能力的参与者的抽象表示。角色 agent 和具体的参与者共同担负相应的角色。角色 agent 具有所需的数据和知识, 由管理 agent 根据 workflow 过程的定义产生。它的作用主要是进行必要的信息处理工作, 提醒合适的参与者进行某些业务活动, 并把结果传递给管理 agent, 以便通过其它角色 agent 与相关的参与者合作。每个角色 agent 拥有自己的工作任务表。这些信息都存储在 workflow 数据库中(workflow 相关信息、控制信息和流程实例运行的历史信息等)。任务表包括可能胜任的参与者信息, 相关的角色, 交互的触发事件, 活动集的描述以及输入和输出信息。每个角色可以向资源 agent 申请所需的资源, 操作相应的应用程序, 分别在不同的岗位上履行其职能。当执行过程出现异常时, 可以访问角色本地的 ECA 规则库, 以便处理异常。

根据角色 agent 能力的互补性, 可以划分为若干组, 构成能力更强的虚拟智能体, 以辅助某一职能或组建工作小组的合作, 这与企业每一个角色 agent 根据当时的实际情况辅助适当的参与者的组织模型有关。

(4) 参与者(用户). 通过角色 agent 执行活动以及与其他参与者进行必要的合作。

(5) 接口 agent. 负责参与者与角色 agent 的交互。用户也可直接和自己的角色 agent 对话。

角色驱动的工作流模型的运行大致过程如下: workflow 服务器根据 workflow 模型, 和管理 agent 交互, 生成相应的 agent, 并初始化有关参数, 通过资源 agent 申请所需的资源, 并和相关的角色 agent 在若干活动点交互, 也可能通过管理 agent 与其他角色 agent、参与者协作。对于比较复杂的工作流, 相关角色之间的交互可能是多次、贯序状的, 而不是单纯顺序的。在 workflow 的运行过程中, 可能遇到例外, 如果这种例外不能激活 workflow 的 ECA 规则时, 可以由 workflow 管理员通过管理 agent 修改 workflow 模型。

3.2 讨论

基于活动的工作流模型强调完成流程的逻辑步骤, 相应地, workflow 管理系统按照预定义的企业规则, 通过对活动的调度和监控, 推进 workflow 实例的自动执行。其中, 虽然 workflow 引擎也需要调用应用程序以及与参与者交互, 但 workflow 引擎是通过活动之间顺序控制来集成活动主体的功能。参与者完全是根据 workflow 任务管理器提供的任务项完成具体的操作。如此, 参与者表现为一种被动的活动执行体, 它们之间的关系被弱化地散布到活动之间的关系中, 所承担的角色不容易识别。此外, 基于活动的工作流体系结构与上述 workflow 实例的运行机理不同。这是因为面向角色的 workflow 把描述流程的焦点集中在角色及其合作上, 反映了 workflow 管理系统的群件本质。角色的 workflow 任务表体现了其责任, 其中的活动在后台运行, 支持角色参与合作、交流信息和完成赋予的子目标的手段。workflow 实例的运行则表现为角色的交互链, 是一种客户服务的关系。在面向角色的 workflow 模型中, 角色是主动的实体, 它的属性、行为及与其它角色的交互关系很自然地可用多 agent 系统来模拟。随着 agent 技术的发展, 这种方法也会给目前不成熟

的 workflow 建模和仿真提供新的方法。

3.3 仿真

在上述分析基础上设计了基于角色的 workflow 模型原型。演示系统选用的仿真平台是 Java 语言的编程环境 JDK, 其中, JADE (Java agent develop framework) 提供了一种基于 Java 语言的构建 agent 系统的框架, 它完全支持 agent 间通讯标准 FIPA。还有 JESS (Java expert system shell) 是一种由 Java 语言编写的专家系统外壳, 它提供了一种脚本语言来编辑角色的活动规则, 这种脚本语言和一种常见的专家系统语言 CLIPS 很相似。同时它也提供了 Java 语言的接口, 使得用 Java 语言编写角色的活动规则变得相对简单。在设计 agent 的过程中, 尽量将 agent 的一些通用属性设计在角色类中, 而将更多的客户化的东西放在角色活动规则中实现。在实现 agent 的时候, 一方面基于 JADE 构造 agent, 并且能够在 JADE 的容器中顺利运行; 另一方面要为每一个 agent 加入一个推理机, 这是通过嵌入 JESS 的方式实现的。管理 agent 用驻留在容器中的精灵 agent 实现, 由它负责管理其它 agent, 如其它 agent 向管理 agent 注册, 管理 agent

作为中介协调其它 agent 之间的交流等。

面向角色的多 agent 仿真模型能模拟参与者的角色分配, 多角色的信息传递和交流, 资源的分配, 特别是角色之间的客户服务关系。也可通过计算 workflow 的运行时间等定量性能指标比较不同方案的性能以便优化流程, 发现 workflow 存在的瓶颈等问题。这是下一步要重点研究的问题。

4 结束语

本质上讲, workflow 系统是一种支持多人协作的群件系统, 其核心是参与者的角色分配和协调问题。以往的工作流模型多是基于活动的, 忽略了参与者的角色定位和合理配置。这种 workflow 模型在实际应用中缺乏柔性, 对意外难以处理, 可重构性不足等问题, 直接影响了 workflow 的实用范围。提高 workflow 的适应性, 是新一代 workflow 管理系统关注的问题。角色模型和目前分布式计算的多 agent 系统在概念上是一致的, 可以用多 agent 技术来仿真 workflow。本文给出了面向角色的多 agent workflow 管理系统模型的原型。

参考文献:

- [1] Whitling John T. Reengineering the corporation: A historical perspective and critique[J]. IM, 1994, (Nov/Dec): 14—16.
- [2] Cho Yoon Ho, Kim Jae Kyeong, Kim Soung He. Role-based approach to business process simulation modeling and analysis[J]. Computers Ind. Engng., 1998, 35(1-2): 343—346.
- [3] 范植华, 白光野, 蒋东溟, 等. 角色编辑器 Rb-MDE 的设计与实现[J]. 计算机学报, 1996, 19(5): 398—401.
FAN Zhi-hua, BAI Guang-ye, JIANG Dong-min, et al. Design and implementation of the role editor Rb-MDE[J]. J. Computers, 1996, 19(5): 398—401. (in Chinese)
- [4] Biddle B J, Thomas E J. Role Theory: Concepts and Research[M]. New York: John Wiley & Son Inc., 1996.
- [5] Huang G Q, Huang J, Mak K L. Agent based workflow management in collaborative product development on the internet[J]. Computer Aided Design, 2000, 32: 133—144.
- [6] Gregory Mentzas, Christos, Stylianos Kavadias. Modeling business process with workflow systems: An evaluation of alternative approaches[J]. International Journal of Information Management, 2001, 21: 123—135.
- [7] Malone T W, Crowston K. What is Coordination Theory and How Can It Help Design Cooperative Systems?[R]. Proc CSCW'90, 1990. 357—370.
- [8] Phalp Keith Thomas, Henderson Peter, Walters Robert John, et al. RolEnact: Role-based enact able models of business processes[J]. Information and Software Technology, 1998, 40: 123—133.
- [9] Davis Randall, Smith Reid G. Negotiation as a metaphor for distributed problem solving[J]. Artificial Intelligence, 1983, 20: 63—109.

Role-based multi-agent workflow systems

ZHAO Wei-dong¹, HUANG Li-hua²

1. Software College, Fudan University, Shanghai 200433, China;

2. School of Management, Fudan University, Shanghai 200433, China

Abstract : Many workflow modeling tools have been developed, but most of them describe business process in terms of activity, product, goal or decision-making. Little attention is paid to roles and interactions among them. However, the success of an enterprise depends on its people and their accountabilities in the final analysis. In this paper, role-oriented workflow modeling is analyzed from the perspective of roles and their interactions. Then how to represent the role-oriented workflow models is discussed, which provide a new method for workflow management. Furthermore, with reference to the architecture of WFMC, role-driven approach to workflow management systems described by multi-agent systems (MAS) is proposed based on the analysis of the relationship between agents and roles, between MAS and workflow systems. A prototype is realized based on the approach.

Key words : role; agent; workflow model; workflow management system

(上接第 46 页)

Senior debt and incentive effects of junior debt

XUE Ming-gao^{1,2}, LI Chu-lin², GONG Pu¹

1. Department of Finance, College of Management, Huazhong University of Science and Technology, Wuhan 430074, China;

2. Department of Mathematics, Huazhong University of Science and Technology, Wuhan 430074, China

Abstract : A model of incentive effects of senior debt and junior debt is developed in the paper, in which both the rate of inflation and the rate of dividend yield are incorporated. Using the method of the game theory analysis of options, the analytic valuation formulas of the senior debt, junior debt, equity and firm, respectively, is presented in the paper. The paper discuss that the effect of the rate of inflation and the rate of dividend yield on the firm's bankruptcy decision, the firm's decision to issue junior claims and this wealth transfer between the debt holders and the equity holder, respectively. In order to demonstrate that the rate of inflation and the rate of dividend yield cannot be neglected in borrow-lend contracts.

Key words : senior debt; junior debt; game theory; option; endogenous bankruptcy-trigger value