

复杂分布式系统的面向智能 agent 建模

付 宇, 徐 涛, 丁建立, 王 红

(中国民航大学 计算机科学与技术学院, 天津 300300)

摘 要: 为了有效管理复杂分布式系统建造过程中的复杂性, 提出了一种面向智能多 agent 系统的工程化软件建模技术. 该方法使用扩展的 UML 用况图和顺序图来认定角色并建立角色模型, 通过对 agent 的心智状态建模, 使用扩展的 UML 状态图和 Petri 网图来发展 agent 自治行为和交互行为的动态模型. 给出了一个覆盖了从问题域分析到生成 agent 类模型和 agent 交互模型全部过程的网络资源管理系统应用实例, 并验证了建模方法抽象和管理系统复杂性的有效性.

关键词: 复杂分布式系统; 心智状态模型; 交互模型

中图分类号: TP 311

文献标志码: A

文章编号: 0254-0037(2011)06-0955-06

复杂分布式系统具有交互复杂性和个体复杂性^[1-2]: 部件之间的交互不仅收发消息, 而且参与复杂的合作协议; 既以目标驱动的方式主动完成给定的任务, 又以反应驱动的方式对环境的变化及时作出反应. 这类系统广泛存在于过程控制、医疗监控、网络管理等任务攸关领域. 建造高质量的复杂分布式系统是软件工程领域一项富有挑战性的工作. 智能 agent^[3] 技术已被证明是一种有效的手段, 但软件工程师仍需要严格的软件工程范型将其开发过程工程化. 目前已有的工作中, Gaia 方法^[4] 通过组织结构的抽象导出多 agent 系统的分析与设计, 但没有提供具体的方法来确定系统中各角色及其交互; MaSE 方法^[5] 提供了直接从需求分析半自动地产生 agent 内部结构的方法和工具, 但在刻画 agent 的智能性方面存在不足.

本文提出了一种系统化的软件开发与建模方法来管理上述智能多 agent 系统的复杂性. 工作集中于软件开发的建模阶段, 以一个通信网络带宽管理系统作为应用的问题域, 将生成由具有心智状态模型^[6-7] agent 组成的多 agent 系统作为目标域, 围绕着对 agent 自治性和 agent 交互性 2 条线索, 通过一个从抽象到具体的建模过程逐步递增构造出系统所需的细节模型. 相对于 Gaia 方法的不足, 分析阶段从对问题域复杂性的分解开始, 通过显式的图表与规则抽取出 agent 角色及其职责模型. 与 MaSE 方法不同的是, 把对 agent 智能性的刻画放在首要位置, 对 agent 心智状态(由信念、能力、目标等组成)及其转换的建模贯穿了分析到设计的整个过程.

1 系统建模过程

如图 1 所示, 整个建模过程分为分析和设计 2 个层次. 分析阶段从问题域分析开始, 通过扩充的 UML 用况图和顺序图形成 agent 的 Role Schema 和 Protocol definition. 在设计阶段则将角色模型进一步具体化为 agent 类模型, 形成目标系统更为充实的 agent 信念部件(belief set)、动机部件(agent 目标分解图)和能力部件(plan 过程体图), 并通过 agent 交互过程体图获得 agent 交互行为的动态模型.

建模过程引入迭代式软件开发思想^[8], 划分成问题域分析、agent 角色认定、建立 agent 角色模型(与

收稿日期: 2009-08-30.

基金项目: 国家“八六三”计划资助项目(2006AA12A106); 中国民航大学科研启动基金资助项目(07QD11X); 天津市应用基础及前沿技术研究计划基金资助项目(09JCYBJC02300); 中央高校基本科研业务费中国民航大学专项资助项目(ZXH2009A006).

作者简介: 付 宇(1973—), 男, 浙江义乌人, 讲师.

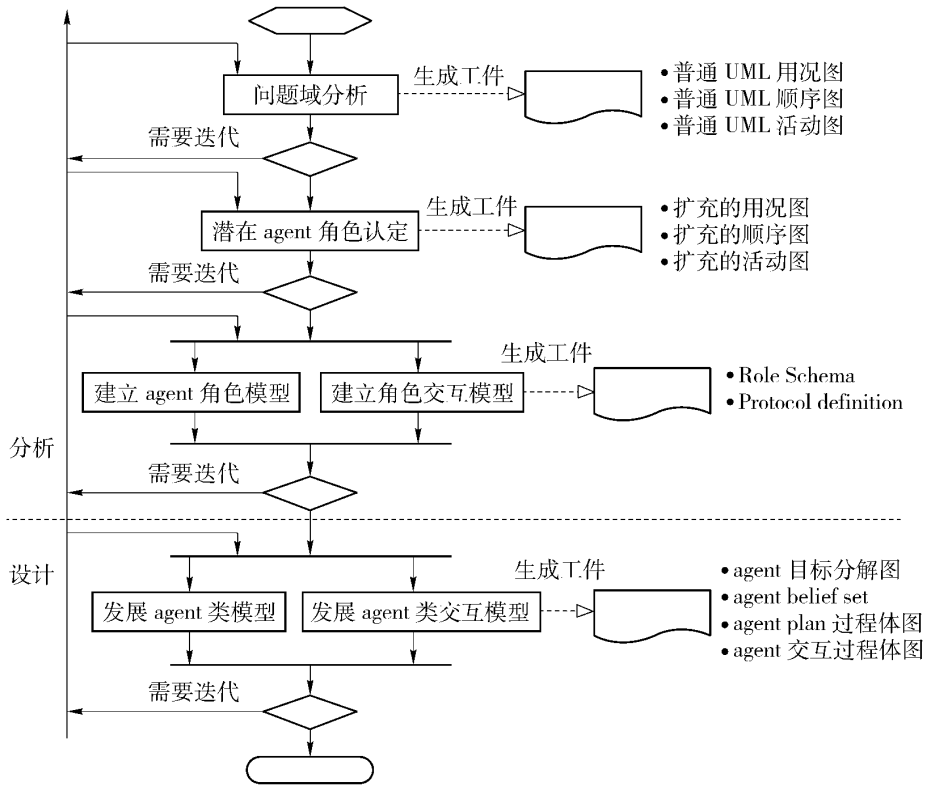


图1 面向智能 agent 的系统建模过程

Fig. 1 Modeling process of developing agent-oriented system

建立角色交互模型并行)、发展 agent 类模型(与发展 agent 类交互模型并行)等子阶段,以生成工件作为迭代目标.在每个子阶段的结束处判断是否需要返回上一阶段进行迭代.迭代式开发有助于实现整个建模过程的持续质量保证.

2 面向智能 agent 的系统分析

2.1 问题域分析

分析阶段的目标是建立对系统及其结构的理解而不涉及实现的细节.整个过程从建立普通的 UML 用况图、顺序图和活动图开始,并使用以下规则来抽取系统中的智能 agent.

1) 自治性: ①它是否使用内在的信息部件来保存自身和环境的实时状态? ②它是否自主决定何时、做什么动作?

2) 主动性和反应性: ①是否需要目标驱动的行为? ②是否需要对环境的变化做出实时的反应?

3) 社会性: ①是否需要与其他 agent 交换知识层上的信息? ②是否需要与其他 agent 合作完成目标?

对于顺序图作者将对象之间的消息扩充为承载 agent 通信语言的消息,并增加了表示 agent 内部动作的图符以区别于 agent 间的交互动作.为了支持 agent 概念,对 UML 用况图作如下扩充:

1) agent 模型为带有矩形头的 actors.

2) 环境中的元素模型为 clouds.

3) 在 1 个 actor 与 1 个反应用况间的弧线表示该 actor 是触发这个用况的源.

2.2 建立角色模型

该过程根据问题域分析步骤中初步获取的 agent 静态、动态方面的特性,生成该角色的 Role Schema.

角色模型的主要属性有:

- 1) 职责(responsibilities) ,表征角色的关键属性.
- 2) 知识库(knowledge base) ,反映了实现角色的职责所应该掌握的关于环境和自身状态的信息.
- 3) 目标驱动行为(goal-directed actions) ,体现了 agent 的主动性.
- 4) 反应驱动行为(reactive actions) ,体现了 agent 的反应性.
- 5) 交互(interactions) ,描述了该角色在实现哪些职责时需要与哪些角色进行交互.
- 6) 权限(permissions) ,描述了角色访问资源的许可方式.
- 7) 安全性(safety) ,反映了一个角色生命周期中所应保持的那些可接受的、不变的状态.

2.3 建立角色交互模型

交互模型由一系列交互协议的定义组成 ,主要包括以下几个属性:

- 1) 目的 ,关于交互本质的简要文字性描述.
- 2) 发起者 ,开启该会话的角色.
- 3) 回应者 ,参与到发起者所发动的会话中的那些角色.
- 4) 输入 ,发起者定义一个协议和开始一个会话时所需要的信息.
- 5) 输出 ,交互过程中提供给回应者的信息.
- 6) 处理过程 ,交互过程中与会话相关过程的文字性描述.

3 面向智能 agent 的系统设计

3.1 发展 agent 类模型

1) 对信念部件建模. 涉及了 agent 关于自身及环境的状态 ,包含 1 个 belief set 模型及其对应的多个 belief 状态.

2) 对动机部件建模. 分为目标和事件 2 类 ,分别用以激活主动型和反应型行为. 目标分为 ACHIEVE、PERFORM 和 MAINTAIN 3 种类型. 1 个目标通常要分解为多个子目标 ,从而形成了目标的层次结构. 对事件驱动建模主要来自环境或 agent 信念集中状态数据的变化.

3) 对能力部件建模. 能力部件由 plan 集合组成 ,包括 plan 名称、应用条件、激活条件和 plan 过程体. 应用条件描述了使用该 plan 时应满足的内部心智状态或外部环境. 激活条件则描述了激活该 plan 的条件是什么目标或事件. 使用扩充的 UML 状态图——plan 图对 plan 过程体进行建模 ,其基本元素包括: 结点(node) 、跃迁(transitions) 以及事件(events) 、条件(conditions) 和动作(actions) . 与传统的面向对象模型不同 ,plan 图包含了失败结点. 中间状态可以包含嵌套的子状态 ,对应于子目标. 事件、条件和动作都附加在连接两状态结点间跃迁的有向边上. 当 1 条有向边上的激活条件满足时 ,其对应的动作将得到执行.

3.2 发展 agent 交互 plan 模型

通过交互 plan 的形式对分析阶段的角色交互建模 ,其重点在于交互消息的序列与同步信息. 已有的方法多使用有限状态自动机模型. 由于只能将这些会话以相互之间无联系孤立进程的形式表现出来 ,该模型在处理多 agent 系统的并发性方面存在不足 ,因此使用了扩充的 Petri 网图对 agent 交互过程建模:

1) 每个参与交互的 agent 被描述为 1 个 Petri 子网 ,位置对应于 agent 的内部状态 ,跃迁对应了并发动作的起始或同步点.

2) 将 agent 部件进行接受和发送消息的接口表示为输入和输出端子 ,一个部件的输入(输出) 端子总是会与某个子部件的相应端子相连(当然也有可能是子部件的相应端子的组合) . 连接 2 个部件的输入和输出端子的通讯链使部件之间能交换信息.

4 应用实例

异步传输模式网络的 VPC (virtual path connection) 管理面临着动态、复杂的环境: 链接是动态生成并不断变化的, VPC 上瞬间传输的业务量也存在着波动. 如果系统不能对变化作出及时调整则会出现其他 VPC 仍有空闲带宽的情况下, 由于未能及时利用这些空闲带宽而导致其他部分 VPC 处于拥塞状态. 图2展示了1个网络局部, 其中2条物理链路(link-AB 和 link-CB) 共承载了4条 VPC(VPC 1 ~ VPC 4) . 主要管理实体包括:

- 1) 对应物理链路的 link manager ,负责对物理链路路上的 VPC 和空闲带宽资源进行分配.
- 2) 对应 ATM switch 结点的 node manager .
- 3) 对应 VPC 的 VPC manager ,负责该 VPC 带宽、路由和虚电路 VCC(virtual channel connection) 管理.
- 4) 连接准入控制模块(connection admission controller) 对每个 VCC 传输请求选择一个 VPC manager 来完成传输.

系统建模过程如下: 首先根据 agent 选择规则初步选定 link manager、VPC manager、CAC 和 Node manager 这4类 agent 角色, 记为 link agent、VPC agent、CAC agent 和 Node agent ,然后使用如图3所示的扩展 UML 用况图来获取 link agent、VPC agent 的目标、反应事件与环境之间的关联. 分析阶段的结果是建立详细的 agent 角色模型和交互模型.

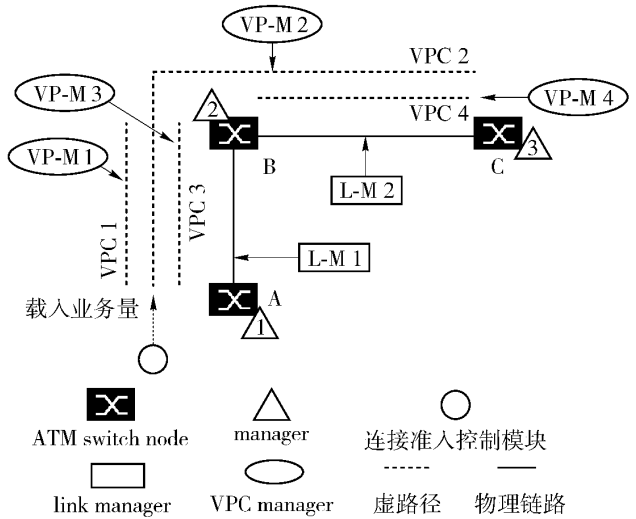


图2 1个 ATM 网络管理局部
Fig.2 A local view of an ATM network

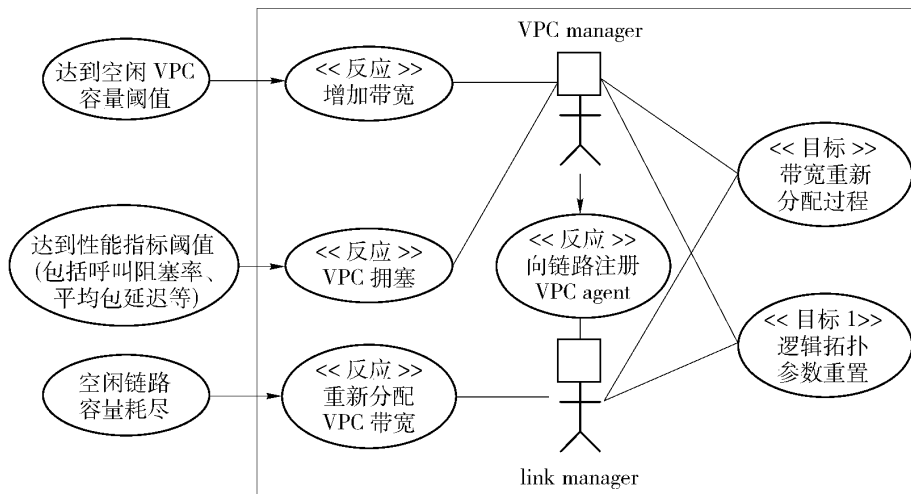


图3 扩展的 agent 用况图
Fig.3 Extended agent usecase diagram

设计阶段主要在角色模型基础上发展出系统的 agent 类模型. 首先,对 agent 的信念部件建模,建立 VPC agent 信念部件的部分 belief set 模型示例如下(\$ 标记的项为变量):

- VPC-identify (ID , source-node , destination-node)
- travelPath (list of (physical link))
- bandwidthAllocated (\$ VPC-ID , \$ VP-bandwidth)

• AcceptedACC (\$ VCC-ID)

然后,对 agent 的动机部件建模. 图 4 给出了 link agent 能力库中名为“LM-startUp”的 plan 过程体图,该 plan 使得 1 个 link agent 初始化后开始响应其他 agent 的服务请求. 最后,对 agent 类的交互模型建模. 图 5 给出了使用扩充的 Petri 网图描述的 VPC agent 某一 plan 的交互模型,该 plan 使用 request 通讯原语向 link agent 提出增加虚路径带宽的请求.

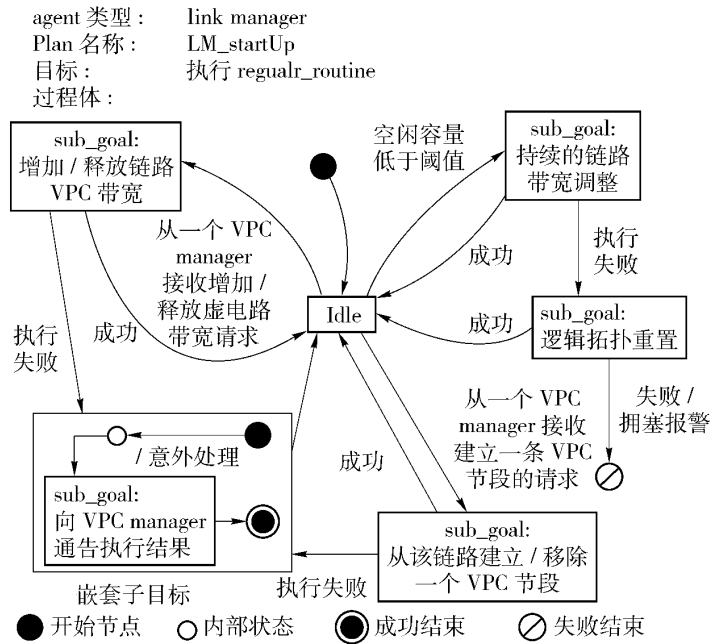


图 4 Plan 过程体图

Fig. 4 Plan body diagram

agent 类型: VPC manager
 Plan 名称: Inter_LM_addBandwidth
 目标: ACHIEVE IncreaseBandwidth(\$vpc_m,\$link_m,\$volume)
 成功: ACHIEVE commit(\$vpc_m,addBandwidth(\$volume))
 失败: refuse_do(\$link_m,\$vpc_m,addBandwidth(\$volume))
 过程体:

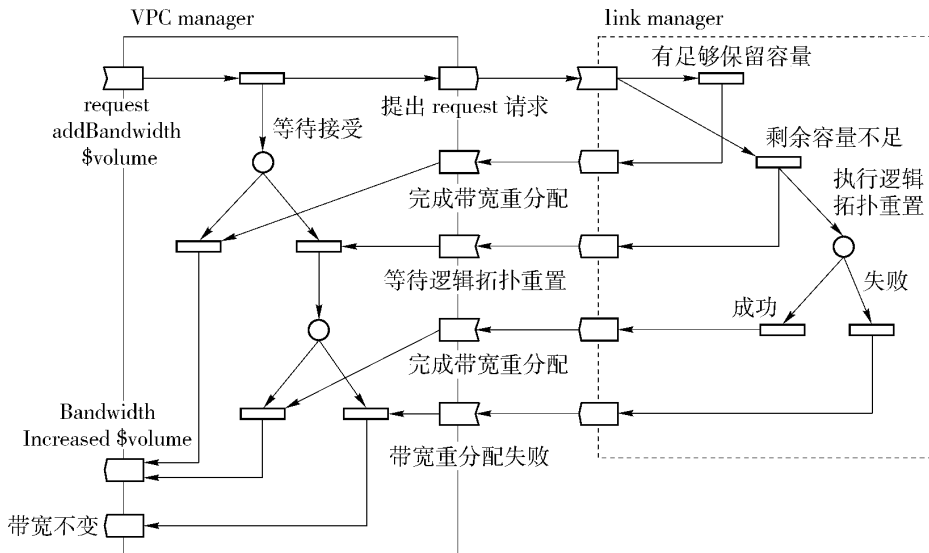


图 5 对 agent 类的交互 plan 建模

Fig. 5 Interaction plan model of agent class

5 结束语

作者提出了一种系统化的软件开发方法与建模技术来管理智能多 agent 系统复杂性. 方法的重点是角色、职责和目标的认定,同时目标驱动的分析使得系统更加稳定、健壮和模块化,系统可以动态更换为更为合适的的能力来达到同样的目标.

参考文献:

- [1] TATARA E, CINAR A, TEYMOUR F. Control of complex distributed systems with distributed intelligent agents [J]. *Journal of Process Control*, 2007, 17(5): 415-427.
- [2] TIANFIELD H, JIANG T. Agent-based modeling and simulation of Complex Distributed Systems [C]//*Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA 2008)*. Piscataway: Institute of Electrical and Electronics Engineers, Inc., 2008: 416-421.
- [3] 毛新军,常志明,王戟,等. 面向 Agent 的软件工程: 现状与挑战 [J]. *计算机研究与发展*, 2006, 43(10): 1782-1789. MAO Xin-jun, CHANG Zhi-ming, WANG Ji, et al. Agent-oriented software engineering: status and challenges [J]. *Journal of Computer Research and Development*, 2006, 43(10): 1782-1789. (in Chinese)
- [4] GARCÍA-OJEDA J, ARENAS A, PÉREZ-ALCÁZAR J. Paving the way for implementing multiagent systems: integrating Gaia with agent-UML [C]//*Agent-Oriented Software Engineering VI, Lecture Notes in Computer Science, Volume 3950*. Berlin: Springer, 2006: 179-189.
- [5] 陈青华,谢晓方,吕红,等. 以 MaSE 方法开发的无人机指挥控制系统设计 [J]. *海军航空工程学院学报*, 2008, 23(2): 177-180. CHEN Qing-hua, XIE Xiao-fang, LÜ Hong, et al. Research and design of UCAV command and control system based on MaSE [J]. *Journal of Naval Aeronautical and Astronautical University*, 2008, 23(2): 177-180. (in Chinese)
- [6] VERNON D, METTA G, SANDINI G. A survey of artificial cognitive systems: implications for the autonomous development of mental capabilities in computational agents [J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(2): 151-180.
- [7] SHARPANSKYKH A, TREUR J. Relating cognitive process models to behavioural models of agents [C]//*Proceedings of 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'08)*, Volume 2. Washington, DC: IEEE Computer Society, 2008: 330-335.
- [8] PAES C, HIRATA C, YANO E, et al. Extending RUP to develop fault tolerant software [C]//*Proceedings of the 2008 ACM Symposium on Applied computing*. New York: Association for Computing Machinery, Inc., 2008: 783-790.

Intelligent-agent Oriented Modeling for Complex Distributed Systems

FU Yu, XU Tao, DING Jian-li, WANG Hong

(College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China)

Abstract: To deal with the engineering complexity during the process of developing complex distributed systems, a software modeling approach for the Intelligent-agent oriented system has been proposed. The extended UML diagram and the sequence diagram are used to identify potential intelligent agents, and an agent class model is generated by specifying its mental state characters. The agent dynamic action model is developed with extended UML state diagram and Petri net graph. An example of developing a telecom network resource management system has been illustrated, which indicates that the modeling approach provides efficient solutions to manage the domain complexity.

Key words: complex distributed system; mental-state model; interaction model

(责任编辑 梁洁)