

# FKAOS:一种面向 Agent 需求工程方法

刘宗田<sup>1</sup>,邵 翀<sup>1</sup>,孙志勇<sup>2</sup>,刘 炜<sup>1</sup>

(1. 上海大学计算机工程与科学学院,上海 200072;2. 合肥工业大学计算机系,安徽合肥 230039)

**摘 要:** 面向 Agent 需求工程重点研究软件系统的面向 Agent 的抽象组织形式. 本文提出一种新的面向 Agent 需求工程方法——FKAOS 方法. 该方法作为较高抽象程度的以 Agent 为核心的软件系统描述方法能辅助用户和系统设计人员更好地分析和设计系统需求. FKAOS 方法通过引入 Fuzzy Z 语言,扩展经典的 KAOS 方法,描述多 Agent 系统中的非确定需求问题. FKAOS 方法中的描述视图引入了 KAOS 方法的目标和责任视图,并添加了一些新的专门描述面向 Agent 的需求问题的视图和表. 本文重点介绍了 FKAOS 方法的基本元模型概念和相关的主要视图描述模型等内容,并通过一个实例演示了 FKAOS 方法的实际应用过程.

**关键词:** 面向 Agent 需求工程; FKAOS 方法; DESIRE 环境; Fuzzy Z 语言; KAOS 方法; Agent

**中图分类号:** TP396. 81 **文献标识码:** A **文章编号:** 0372-2112 (2003) 12A-2171-04

## FKAOS :A Method for Agent-Oriented Requirement Analysis

LIU Zong-tian<sup>1</sup>,SHAO Kun<sup>1</sup>,SUN Zhi-yong<sup>2</sup>,LIU Wei<sup>1</sup>

(1. School of Computer Engineering and Science of Shanghai University, Shanghai 200072, China;

2. Computer Application Institute of Hefei University of Technology, Hefei, Anhui 230039, China)

**Abstract:** Agent-oriented requirement analysis focuses on determining a higher-level organization of a system that meets a given problem specification. This paper presents the FKAOS agent-oriented requirement engineering method, which helps users and system designers analyze and describe the higher-level agent-centered requirement of software systems. FKAOS extends KAOS by modeling the uncertain property of MAS. It integrates a fuzzy version of Z with the KAOS language. The diagrams of FKAOS extend the goal and responsibility diagrams of KAOS and add some new diagrams and tables to describe the agent-oriented requirement. Based on the metamodels of view diagrams, the FKAOS environment includes a goal decomposing modeling and an agent optimizing modeling to integrate the knowledge about agent-oriented software design principles, prototypical software architectures, and application domain. The paper illustrates the design in FKAOS through an example, focusing on the ability of FKAOS to provide a cooperative design environment.

**Key words:** AORE; FKAOS; DESIRE; Fuzzy Z; KAOS; Agent

## 1 引言

FKAOS(Fuzzy knowledge acquisition in automated specification)是一种面向 Agent 需求工程方法. 该方法通过引入 Fuzzy Z 语言<sup>[1]</sup>,扩展经典的 KAOS 方法<sup>[2,3]</sup>,描述多 Agent 系统中的非确定需求问题. FKAOS 方法提供了一种能够描述非确定问题的描述语言<sup>[4]</sup>,一套基于目标和 Agent 实体对象的分析方法,和一个多层的可视化表示体系. FKAOS 语言严格定义了一些基本的描述面向 Agent 需求的元模型实体对象,如目标、Agent、角色、约束、责任和资源. DESIRE 是一个支持 FKAOS 方法的 CASE 工具环境. DESIRE 工具环境包括有图形编辑器、目标分析工具、Agent 优化工具和 XML 报告的产生器等<sup>[5]</sup>.

FKAOS 方法的分析视图是一个描述 FKAOS 的基本元模型概念和关系网络<sup>[5]</sup>. 这些视图包括目标精化视图 (Goal Refinement view, GRV), Agent/角色视图 (Agent/role view, AV), Agent 责任视图 (ARV), 组织视图 (Organization view, OV), Agent 约简视图 (Agent reduce view, ADV) 和域视图 (Domain view, DV)

等. 与元模型和视图相关的标识、图和方案的主要思想在随后的各节中结合一个会议调度系统的设计实例详细介绍.

## 2 FKAOS 基本元模型概念

主要的 FKAOS 元模型概念和关系<sup>[5]</sup>描述如下:

(1) **目标** 目标概念定义了一个需要通过多个 Agent 相互协同才能够实现的合成系统的需求目的. 目标可以通过目标责任的分配与 Agent 相互关联或通过精化连接将其分解为多个子目标.

(2) **Agent** Agent 作为需求系统的基本组件单位,能够为相关系统目标的实现提供一些潜在的必要功能. 这些功能的能力是实现 Agent 所承担责任的重要手段.

(3) **角色** 角色和 Agent 之间的不同类似于接口与类概念之间的不同. 角色描述了 Agent 的外部特性. 一个 Agent 可以扮演多个角色,同时多个 Agent 也可扮演同样的一个角色.

(4) **资源** 资源被用来表示实现系统设计所需求的一些具体实体对象,如数据库表和实现特定功能的基本程序模块

等. 资源一般可以在不同的应用环境中多次重复使用. 通常的面向对象的概念如对象、组件和类等也可作为构成系统服务的可重用的基本资源使用.

(5) 责任 目标的责任和 Agent 所负担的责任采用统一的描述领域责任的责任本体表示. 通过对目标分解过程中获得的相关领域知识的分析建立责任本体.

(6) 目标关系 在目标精化过程中, 我们定义了四种推理精化关系: Relationships (表示从  $C_1$  元模型到  $C_n$  元模型的  $n$  元关系, 被形式化的定义为  $R = \text{TUPLE}(C_1, \dots, C_n)$ ), AndOr relationships (表示一个目标能够被约简为几个子目标的 And 或 Or 关系的结合), IsA relationship (IsA( $C_1, C_2$ ) 表示每一个  $C_1$  元模型的实例也是  $C_2$  元模型的一个实例) 和  $fR$  关系.

图 1 是 FKAOS 方法的主要元模型关系图, 以 Agent 元模型为核心, 显示了主要各类元模型概念之间的相互关系.

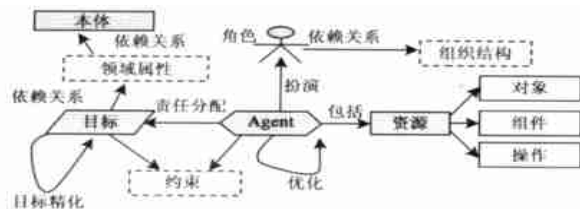


图 1 FKAOS 方法的主要元模型关系

### 3 FKAOS 方法的分析过程

本节通过一个分布式会议调度系统的应用实例, 演示使用 DESIRE/ FKAOS 需求分析环境分析待开发软件系统需求的全过程. 演示过程以反映系统不同特性需求的各类视图模型的构建为主线, 辅助以一些简要的描述说明. 作为一个概要的应用演示, 该分析过程仅仅对其中的主要环节进行了说明, 忽略了一些不太重要的枝节问题.

#### 3.1 问题描述

本论文选择一个分布式会议调度系统的应用作为演示 DESIRE/ FKAOS 需求分析环境的分析对象. 分布式会议调度应用的需求分析是一个非常典型又较为复杂的过程. 其复杂性主要体现在, 首先, 在这样的应用中同一个人可以扮演几种不同的待开发系统的角色: 系统角色、领域专家角色和某种特定的角色. 其次, 该应用需求的分析过程将会涉及到许多具有挑战性的特性, 例如障碍目标分析、实时和分布设计、多 Agent 的协作和通讯结构、模糊意念表示和优化 Agent 设计等<sup>[6]</sup>.

应用需求描述: “会议调度系统的设计是自动会议组织服务系统实现的重要支持目标. 其主要责任是为每一个申请举办的会议分配一个最合适的会议时间和会议地点, 以满足必须出席会议的关键人物和最大数量的相关人员参与.”

初始目标: 根据上文的简单需求描述, 我们提取出两个初始目标: 满足会议请求和保证最大数量的会议参与人员. 利用 FKAOS 语言, 该实例分析的初始目标被定义如下:

**SystemGoal Achieve** [ 请求会议举行 ] **InstanceOf** SatisfactionGoal ; ...

**SystemGoal Maximize** [ 参与会议人数 ] **InstanceOf** Satisfac-

tionGoal ; ...

在本文中, 为了描述简便, 我们仅讨论第一个初始目标的分析和设计, **SystemGoal Achieve** [ 请求会议举行 ].

#### 3.2 需求建模第一步

需求建模的第一步行为以获取的初始目标为分析对象, 针对初始目标进行分析. 基于目标对象的分析行为, 同时建立一些重要的有关领域知识的描述模型, 如域模型, 责任本体模型等.

(1) 目标分析视图 (GAV) 目标分析视图是整个 DESIRE/ FKAOS 需求分析环境的核心分析视图之一. 在这个视图中, 用户的初始目标被逐步精化为一个通过 AND、OR 或  $fR$  关系连接的子目标的目标结构图. 这一视图的分析过程涉及目标定义、目标分类、域属性描述、目标精化和目标冲突等行为, 以及这些行为之间的相互依赖关系的描述等. 当该分析步骤分解出的子目标的责任能够被分配到一个独立的 Agent 实现时, 就表明该子目标已经是最小的基本目标单元, 并停止精化该子目标. 一个目标的实现最终将依赖这些与其分解出的各个基本单元子目标相关的多个潜在 Agent 的相互协作.

在目标分析视图中, 设计者使用 AND、OR 和  $fR$  模糊推理等关系分解主目标, **system Achieve** [ 请求会议举行 ]. 有关该目标的分解视图如图 2 所示.

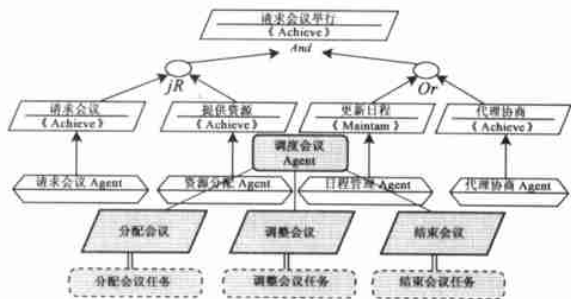


图 2 目标分析视图

由图 2 可以看出, 当举行会议相关的必要物质资源, 如日期和地点等, 和必须出席会议的关键人员等都能满足举行会议的条件时, 申请召开会议的主要目标才被最终满足, 否则将需要重新调整会议安排或修改会议请求. 子目标 Achieve [ 请求会议 ] 提出举行会议的请求和一些重要的举行该会议的必要条件. 同时, 子目标 Achieve [ 资源分配 ] 描述了当前或预期的会议资源的使用状况, 如日期和地点等已经被计划安排的信息和相关关键人员的日程安排状况等. 这两个子目标 Achieve [ 请求会议 ] 和 Achieve [ 资源分配 ] 之间, 被看作是一种冲突的关系, 这种冲突关系体现在动态的会议调整与已经确定的其他会议计划之间的矛盾. 我们使用  $fR$  模糊推理关系表示这两个目标之间的冲突问题. 使用 FKAOS 方法提供的  $fR$  模糊推理关系分析冲突问题不仅能够表示出这两个目标之间的冲突关系, 而且还可以分析产生这种冲突的原因, 并且依据领域经验选择出最佳的解决方案. 可选择的解决方案由子目标 Maintain [ 更新日期 ] 和子目标 Achieve [ 代理协商 ] 描述.

(2) 域视图 (DV) 域视图描述有关待开发系统的主要领

域知识概念及其相互关系(例如:对于一个会议调度系统的需求设计来说,域视图涉及的主要领域知识概念包括有“会议调度者”、“会议参与人员”、“会议地点”以及“会议时间”..等)

域视图中有领域知识概念及其相互关系的表示符号借用了典型的 UML 标准的类及其关联的标识. 其中的类标识表示与领域相关的特定知识概念,而类与类之间的关联表示这些领域知识概念之间的特定关系. 该视图随着其他视图的建立而不断完善,也就是说根据建立其他视图的需要不断地在该视图添加相关的概念和关系. 图 3 给出了一个简单的会议调度系统的域视图实例.

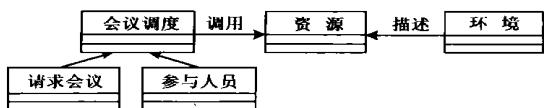


图 3 域视图

为了满足对 Agent 元模型进行责任优化的需要,我们在构建目标分析视图和域视图的过程中同时地构建一个描述相关领域责任的责任本体. 详细的有关责任本体的构建过程请参阅相关文献[7].

3.3 其他分析步骤

我们将目标分析过程之外的所有重点针对 Agent 元模型进行分析的过程统称为其他分析步骤. 这些分析行为以 Agent 元模型的优化为核心,并建立描述 Agent 的不同特性的各类视图模型:Agent 责任视图(ARV)、Agent 交互视图(AIV)和 Agent/角色视图(AV)等.

(1) Agent 责任视图(ARV) 相对于 GAV 视图,ARV 视图以 Agent 元模型为核心分析对象. ARV 视图模型描述与 Agent 元模型相关的各类子目标信息及其所承担的责任信息. 依据第一步目标分析过程所建立的责任本体的责任分类关系,Agent 责任优化分析使用责任优化原则对原来分属于不同子目标的 Agent 元模型进行优化. 具体的优化策略和优化方法请参看 Agent 优化分析有关文献的内容[7]. 通过分析会议调度系统应用实例的相关视图,分配会议 Agent、调整会议 Agent 和结束会议 Agent 等三个 Agent 实体在责任本体中都属于调度会议本体的范畴,它们分别描述了该责任本体的不同行为属性. 依据 Agent 责任优化原则,我们可以将这三种 Agent 元模型合并为一个单一的 Agent 元模型:调度会议 Agent. 这里我们给出优化后的调度会议 Agent 的 ARV 视图(图 4):



图 4 调度会议 Agent 的 ARV 视图

ARV 视图模型的设计还涉及到两个贯穿整个分析过程的全局表对象:Agent 资源配置表(RCT)和 Agent 优化表(ADT). Agent 资源配

置表作为面向 Agent 需求分析模型的重要配置信息表,将所有有关 Agent 元模型涉及到的资源信息通过一个共享的表格保存起来.

Agent 优化表是另一个与 ARV 视图模型需要交互的全局表对象. DESIRE/ FKAOS 需求工程环境在创建一个需求工程项目的同时就创建了一个空的 Agent 优化表. 当第一步的目标分析过程完成之后,作为目标分析结果之一的所有 Agent 元模型都被添加到 Agent 优化表中. 在建模 ARV 视图的过程中,依据责任优化原则对 Agent 优化表中的 Agent 元模型进行分析,优化 Agent 元模型的设计,并依据优化后的 Agent 优化表建模 ARV 模型. 调度会议 Agent 相关的 Agent 优化表如表 1.

(2) Agent/角色视图(AV) ARV 视图分析过程所获取的 Agent 资源信息是非常粗糙且不全面的,需要通过进一步的对 Agent 元模型设计结构的分析获取更多且详细的资源信息. AV 视图从建模 Agent 元模型的设计结构入手,重点对 ARV 视图所描述的 Agent 元模型进行进一步的资源分析工作. 对于每个 Agent 及其在系统中所承担的角色,该视图使用图形支持的方案描述该 Agent 的结构特性,例如需要感知的事件、包含的资源对象、相关的行为规则、重要的约束条件和必要的协作行为等.

类似于 UML 方法<sup>[8]</sup>,顺序结构图被用来描述该视图. 一个顺序结构图根据 Agent 元模型所承担的责任描述怎样实现该责任的执行过程. 通过以实现责任为目标的执行过程的分析,自然地获取该 Agent 元模型应包含的资源对象信息、不变的行为规则和对外的交互关系等. 在顺序图中,角色表示与该 Agent 有交互行为的外部组件对象,这些外部对象既可以是另外一个 Agent 元模型,也可以是一个简单的外部监控参数. 图 5 给出了一个调度会议 Agent 的顺序结构图.

同样,AV 视图也需要和 Agent 资源配置表和 Agent 优化表进行交互. 由于 AV 视图的构建就是以获取 Agent 元模型包含的资源信息为主要目标,因此与 Agent 资源配置表的交互过程贯穿了 AV 模型分析的整个过程. 在构建 AV 视图模型的资源对象时,首先 AV 模型会从现有的资源信息中选取,如果

表 1 依据责任优化原则优化后的 Agent 优化表

编号	名称	相关目标	约束关系	责任描述	资源分配
1	分配会议 Agent	分配会议	初始化 Agent 参与管理 Agent 资源管理 Agent	调度会议	分析资源 分析参与人员 通知 确认
2	调整会议 Agent	调整会议	初始化 Agent 参与管理 Agent 资源管理 Agent	调度会议	分析资源 分析参与人员 通知
3	结束会议 Agent	结束会议	参与管理 Agent 资源管理 Agent	调度会议	分析资源 分析参与人员 通知
4	调度会议 Agent	调度会议	初始化 Agent 参与管理 Agent 资源管理 Agent	调度会议	分析资源 分析参与人员 通知 确认

(1, 2, 3)  
(分配会议, 调整会议, 结束会议)

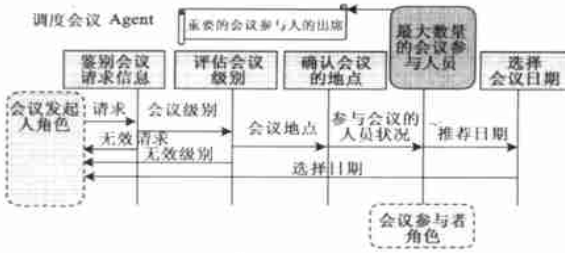


图 5 Agent/角色视图

Agent 资源配置表中包含有该资源则直接引用,如果没有该资源,则需要表中创建一个新的资源记录,并添加到 AV 视图模型中.基于 AV 视图模型的分析,Agent 资源配置表将更加完善,并成为下一阶段面向 Agent 程序设计过程的重要设计依据.

AV 视图与 Agent 优化表的交互,在 AV 视图分析完成之后进行.首先依据 AV 视图分析的结果更新 Agent 优化表中的资源数据信息,依据更新的资源数据信息,按照 Agent 资源优化策略优化 Agent 设计.具体的优化策略请参看文献[7].

(3) Agent 交互视图(AIV) 虽然 GDV 视图、ARV 视图和 AV 视图的分析过程都是以搞清楚 Agent 元模型内在特性为主要目的,但在建模其内在特性的同时我们也有意地添加了一些外部特性的描述,如 AV 视图中的角色的分析和 Agent 优化表中的约束关系描述等.依据 AV 视图中的角色关系和 Agent 优化表中的约束关系描述,我们可以建立起一个反映不同 Agent 元模型之间交互行为的 Agent 交互视图(AIV).

参考  $i^*$  框架的战略依赖模型,我们定义出会议调度系统应用的 AIV 视图<sup>[9]</sup>,如图 6.

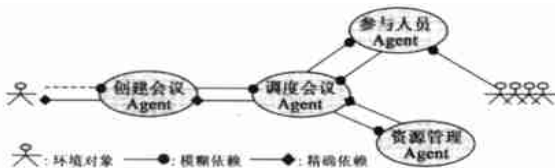


图 6 AIV 视图模型

图 6 中,环境对象:表示需求系统外部的环境因素参与交互的行为;模糊依赖:表示模糊需求依赖关系,体现的是一种协商模式的交互行为;精确依赖:表示明确要求的依赖关系,体现的是命令式的交互行为.

依据交互关系优化结果,Agent 交互视图需要与 Agent 优化表进行交互.交互行为主要表现在,根据 Agent 交互视图分析优化的结果,相应地调整 Agent 优化表中的 Agent 元模型描述.有关 Agent 交互关系优化原则请参看相关文献的论述<sup>[7]</sup>.

### 4 结论

本文通过一个会议调度系统应用的实例分析,详细介绍了 FKAOS 方法各类元模型对象和分析视图的意义和作用.为了让一般用户也能够容易地理解,在选择元模型元素的视图符号时,FKAOS 方法尽量采用已被广泛接受且意义相近的 UML 语

言中的符号.作为对 UML 语言的重要补充方法,FKAOS 方法从一开始就力图将两种方法的相互融合.FKAOS 方法遵从 UML 语言一贯的简便实用的设计原则,尽力降低对用户设计素质的要求,并提供灵活的设计行为.FKAOS 方法的所有分析策略和分析过程都能够既相互关联又相互独立的执行.

### 参考文献:

- [ 1 ] Mattheys C, Swatman P A. Fuzzy Z[A]? The Second Australian Workshop on Requirements Engineering[C]. Sydney:Macquaire University,1997. 99 - 114.
- [ 2 ] Anne Dardenne ,Axel van Lamsweerde ,Stephen Fickas. Goal-directed requirements acquisition[J]. Science of Computer Programming,1993, 20:3 - 50.
- [ 3 ] Lamsweerde. Goal-oriented requirements engineering: A culdled tour [A]. 5<sup>th</sup> IEEE International Symposium on Requirements Engineering [C]. Toronto :IEEE,2001. 249 - 263.
- [ 4 ] Shao Kun ,Liu Zongtian ,Sun Zhiyong. Modeling the imprecise relationship of goals for agent-oriented requirement engineering [J]. Chinese Journal Electronics,2004 ,13(1) :127 - 132.
- [ 5 ] Shao Kun ,Liu Zongtian. DESIRE FKAOS- An environment for agent-oriented requirement analysis [A]. Computer Science and Technology 2003 [C]. Cancun ,Mexico :ACTA Press,2003. 21 - 26.
- [ 6 ] Emmanuel Letier. Reasoning about Agent in Goal-Oriented Requirement Engineering[D]. 2001.
- [ 7 ] 邵 翀. 面向 Agent 需求工程的分析与建模[D]. 上海:上海大学计算机工程与科学学院,2003.
- [ 8 ] Bauer B ,et al. Response to the OMG Analysis and Design Task Force UML 2.0 Request for Information: Extending UML for the specification of Agent Interaction Protocols [DB/OL]. ftp://ftp. omg. org/pub/docs/ad/99-12-03. pdf. OMG,1999 - 12.
- [ 9 ] Eric Yu. Agent-oriented modelling: Software versus the world. Agent-oriented software engineering [A]. AOSE-2001 Workshop Proceedings [C]. LNCS 2222. Springer Verlag,2001. 206 - 225.

### 作者简介:



刘宗田 男,1946 年生于山东省,教授,博士生导师,主要研究领域为软件工程与人工智能理论.已在期刊和会议文集中发表 180 多篇文章,并有几十篇被 EI 和 SCI 检索.



邵 翀 男,1967 年 12 月生于北京,博士,主要研究领域为需求工程技术和 Agent 组件理论,已经发表十多篇论文并有多篇论文被 ISTP、EI 和 SCI 检索.