

知识工程的 KADS 建模方法

程 志

(北京师范大学教育技术学院知识科学与工程研究所, 北京 100875)

摘要: KADS 被广泛认为是最优秀的知识工程建模方法之一。本文论述了 KADS 建模方法的主要特点、KADS 建模的形式描述语言和 KADS 建模的实现环境。

关键词: KADS; 专家模型; 形式描述语言

中图分类号: TP18

文献标识码: A

KADS Modeling Method in Knowledge Engineering

CHENG Zhi

(Knowledge Science and Engineering Institute, School of Education Technology, Beijing Normal University, Beijing 100875, China)

Abstract: The KADS method is known as one of the most excellent methods in knowledge engineering modeling. This paper discusses in detail the three aspects of KADS modeling method, which are the main characters, the formal descriptive language, and the realizing environment of KADS modeling method.

Key words: KADS; expert model; formal descriptive language

0 引 言

KADS (Knowledge Analysis and Design System) 是用于开发基于知识系统的结构化方法, 它为专家知识的抽取以及将这些知识编成计算机程序代码提供很有用的建模方法。在 1983 至 1994 年间, 由于得到了欧共体 ESPRIT 项目的资助, KADS 方法逐渐被开发出来并在许多公司和大学里得到应用。该方法包括以下几个方面: (1) KBS (Knowledge-based System) 的开发过程; (2) 项目管理的生命周期模型; (3) 对与知识工程有关的各种活动的支持; (4) 为 KBS 的开发过程提供技术和工具。

KADS 方法在建模过程中会产生许多文档, 这些文档为建模中产生的各种模型提供说明, 并指导各种模型的开发。因此从理论的观点来看, KADS 被广泛认为是非常优秀的知识工程建模方法^[1]。目前, KADS 方法已成为支持构建知识工程框架的主流方法, 是欧洲知识分析和知识密集型系统开发的标准, 它被欧洲、美国和日本的许多大公司全部地或是部分

地采用。

KADS 进一步发展形成 commonKADS, 它提供了各种支持知识分析和开发的模型, 每一个模型都各自侧重于解决知识系统的某一个限定的方面, 这些模型构成了 commonKADS 知识工程方法学的核心^[2]。

1 KADS 建模方法的主要特点

KADS 方法是知识工程最杰出的建模方法之一, 它的主要特点包括: 为支持知识分析和开发提供各种模型、层次化的专家模型(知识模型)及结构保留的设计方法。

1. 为支持知识分析和开发提供各种模型。

KADS 含有一个预先定义好的模型集合, 集合里的每一个模型都各自侧重于解决知识系统的某一个限定的方面, 这些模型包括^[2]: (1) 组织模型 (organization model): 一个组织模型可以对一个组织的主要特征进行分析, 以发现知识系统的问题和机会, 确定它们的可行性; (2) 任务模型 (task model): 分析全局任务布局、输入和输出、先决条件和性能准则以及所需要的资源和能力; (3) 主体模型 (agent model): 描述主体的特征、能力、行使的权力和权力的约束; (4) 知识模型 (expertise model)

收稿日期: 2005-12-05

作者简介: 程志(1965-), 男, 广东河源人, 北京师范大学教育技术学院知识科学与工程研究所硕士研究生, 研究方向: 信息技术教育应用。

el): 描述执行任务时用到的知识类型和知识结构; (5) 通讯模型(communication model): 描述相关主体之间的通讯事务; (6) 设计模型(design model): 提供技术上的系统规范说明, 包括在实现知识和通讯模型时所需要的体系结构、实现平台、软件模块、表示构造和计算机制等。

组织、任务和主体模型共同分析了组织环境和相应知识系统中的重要成功因素; 知识和通讯模型为问题求解以及知识系统要处理和提交的数据生成概念上的描述, 而设计模型则将这些描述转换成技术规范说明, 作为软件系统实现的基础。各模型之间的关系如图 1 所示。

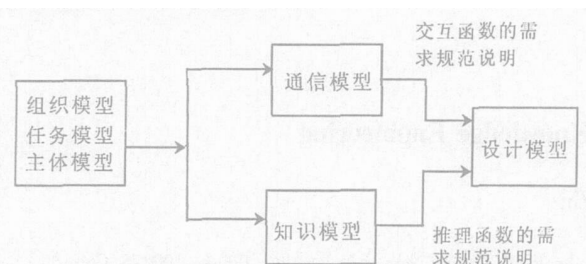


图 1 KADS 各模型之间的关系

2. 层次化的专家模型。

KADS 方法的一个主要贡献是提出了建构层次化的专家模型(expertise model), 这在知识层次上区分了用于解决给定任务的不同类型的知识。该模型分为三层^[3]: (1) 领域层(domain layer)。描述了一个应用领域的静态信息和知识对象, 它由领域的概念、概念的属性、概念之间的关系和规则知识类型组成。(2) 推理层(inference layer)。描述了使用领域知识进行推理的过程。推理知识主要包括推理、知识角色(knowledge role)和传递函数(transfer function)。知识角色又可分为动态角色和静态角色: 动态角色是系统运行时推理的输入和输出; 静态角色是指用于推理的领域知识集合。传递函数是推理主体与外部环境之间转换一条信息项的函数。(3) 任务层(task layer)。描述了一个应用所要达到的目标是什么以及如何通过将任务分解成子任务和推理来实现这些目标。任务通常以层次方式来描述: 高层任务分解成子任务, 这些子任务又可分解成更小的任务。在任务分解的最低层是与任务有关的推理和传递函数。

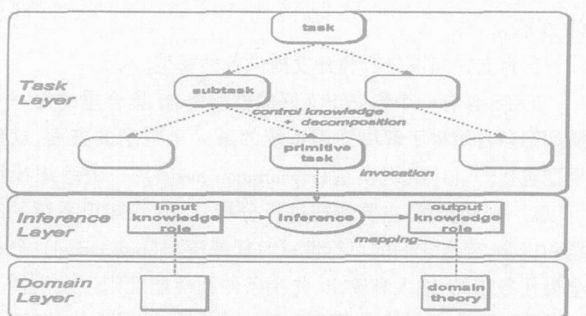


图 2 KADS 的专家模型

专家模型的三层结构是互相联系的, 任务层的任务影射到推理层的推理, 而推理层的知识角色又要影射到领域层里的知识^[4], 如图 2 所示。因此, 将专家模型分成不同的层次可最大化地实现知识的重用。例如, 通过影射可以将相同的领

域知识应用到不同的推理和任务层。当然, 反过来也一样。

3. 结构保留的设计方法。

KADS 的概念模型(专家模型和通讯模型)详细地描述了所要开发的知识系统应有的结构和功能, 所以设计阶段采用结构保留的设计(structure-preserving design)方法, 即设计模型的结构应尽可能地反映概念模型的结构, 以确保系统开发的准确性^[5]。结构保留的设计方法可以有以下几点: (1) 易于理解。由于采用结构保留的设计, 概念模型、系统设计和程序代码有着相似的结构。因此, 对概念模型熟悉的人可以很容易地理解系统设计和程序代码。(2) 可重用代码。由于采用结构保留的设计, 相应于概念模型中某一可重用部分的程序代码也因此是可重用的。程序代码的可重用性可提高程序代码的质量及加快知识系统的设计和开发。(3) 易于维护。由于采用结构保留的设计, 概念模型和程序代码之间保持着清晰的映射, 所以程序代码的出错可以跟踪到概念模型中的某一特定部分。反过来也一样, 概念模型的冗余或者矛盾也很容易被定位到程序代码的某一位置。因此, 大大地简化了系统的维护。

2 KADS 建模方法的形式描述语言

在许多知识工程建模方法中, 专家模型的构建是非常重要的。为了清楚地、无歧义地描述专家模型, 用形式描述语言将专家模型形式化是很有必要的, 而且在知识工程领域里, 人们对形式描述语言在建模中的作用也普遍认可^[6]。将专家系统形式化描述的主要优点有: 首先, 可以克服自然语言描述的模糊性和不确定性; 其次, 可以通过形式论证, 来确认系统的完全性和同一性; 第三, 可以减少系统建模人员和系统实现人员的误解; 第四, 可以将系统的形式化语言影射到操作语言, 以实现用原型方法开发系统。

形式化 KADS 专家模型的语言有: FORKADS、KARL、(ML)²、Model K、Momo 等等。以下简要介绍 KARL 和 (ML)² 两种形式描述语言。

1. KARL 形式描述语言。

KARL(Knowledge Acquisition and Representation Language)是用来形式化和操作化 KADS 专家模型的语言。用对象和关系来形式化描述专家模型是 KARL 致力解决的基本问题。KARL 包含三个部分^[7]: (1) 数据定义语言: 用来定义领域层的关系, 这些关系按父类和子类的层次关系来安排。(2) 逻辑语言: 用于领域层和推理层。用在推理层时, 逻辑语言用来描述通用知识是怎样与领域层中特定知识相关的以及哪一个推理可能进行; 用在领域层时, 逻辑语言用来表达各关系里的成员是怎样相关的以及应满足哪一条限定条件。(3) 描述语言: 用来描述任务层在什么时候有哪一条推理将被启动。这个语言是用来描述控制流。

2. (ML)² 形式描述语言^[8]。

(ML)² 形式描述语言是通过使用不同的逻辑形式来形

化KADS专家结构模型。尽管 $(ML)^2$ 的子集能够操作化以实现系统的原型开发,但 $(ML)^2$ 主要还是用来形式化KADS专家模型。 $(ML)^2$ 包含了用来描述领域知识表征的一阶谓词逻辑;用来表达如何使用领域知识的元逻辑(Meta logic);用来描述控制信息的动态逻辑。

由于逻辑语言本身可以准确地表示领域层所声明和定义的那些领域知识,所以 $(ML)^2$ 用一阶谓词逻辑来形式化KADS专家模型中的领域层。推理的过程离不开领域知识,所以推理层被认为是关于如何使用领域层的理论,这使推理层成为领域层的元层(meta-layer)。因此, $(ML)^2$ 用元逻辑来形式化KADS专家模型中的推理层。任务层的作用是对推理层上的推理步骤进行控制, $(ML)^2$ 用量化动态逻辑(Quantified dynamic logic)来形式化KADS专家模型中的任务层。

3 KADS建模的实现环境

KADS建模的实现环境可以选用Prolog和AionDS两个实现环境中的任一个或者是其他的实现环境。Prolog和AionDS的应用程序都具有以下共同特点:

(1) 面向对象:应用程序的所有功能,包括:用户接口、数据访问、应用过程和知识的表示,全由对象来提供。
(2) 基于知识:应用程序使用基于规则的知识来解决日常事务问题,且规则和规则操作的数据被定义成对象的一部分。
(3) 基于组件:组件是软件模块,它可以通过预先定义好的接口与其他组件通讯。组件可以用一个库或者一组库来表示。

1. Prolog实现环境。

Prolog是programming in logic的缩写,意思就是使用逻辑的语言编写程序,它是由法国的学者柯尔迈伦和他的助手提出的一种高效率的逻辑语言。Prolog本身就是一个演绎推理机,具有表处理功能,能通过合一、置换、消解、回溯和匹配等机制来求解问题。Prolog语言是人工智能与专家系统领域最著名的逻辑程序设计语言^[9]。

Prolog实现的软件结构由三层组成,即两个结构层和一个应用层^[10],如图3所示。

(1) 第一层(实现平台):在Prolog上实现一个面向对象的概念,并应MVC体系结构的要求提供潜在的视图-更新工具。
(2) 第二层(结构工具层):实现从KADS对象到MVC结构的映射。
(3) 第三层(应用实现层):实际应用的实现层。

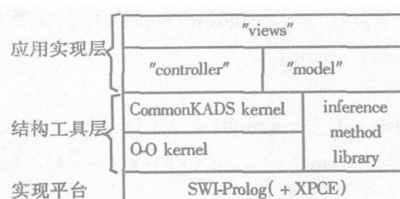


图3 Prolog实现的软件结构

2. AionDS实现环境。

AionDS集成了面向对象和基于规则的方法,主要用于业

务实践的环境。与Prolog相比,AionDS实现环境多了一个任务模板层。任务模板层提供的模板可以作为某类任务的通用实现模型,且对每个特定的任务,需要对其进行实例化。

4 结束语

尽管KADS方法学已作为欧洲开发知识系统的标准并得到广泛应用,但仍然有一些人认为KADS方法工作量太大,以至很难应用于实际的知识系统开发。因此,为了减轻开发人员的任务,在保持系统的结构化方法的同时,用于使知识系统开发过程全部自动化的多种开发工具必须要被开发出来。另外,用于指导设计和开发人员的知识系统也应被开发出来,以指导他们用KADS设计和开发知识系统时的决策^[1]。例如,指导他们为实现知识的分析选择适当的技术,为系统的实现选择适当的工具,等等。

参考文献:

- [1] John K C Kingston. Applying KADS to KADS: knowledge based guidance for knowledge engineering[J]. Expert Systems: The International Journal of Knowledge Engineering, 1995, 12(1).
- [2] Robert de Hoog, Rob Martil, Bob Wielinga, Rob Taylor, Clive Bright, Walter van de Velde. The CommonKADS model set [R]. ESPRIT Project P5248 KADS, 1993.
- [3] Rudi Studer, V Richard Benjamins, Dieter Fensel. Knowledge engineering: Principles and methods [J]. Data&Knowledge Engineering, 1998, 25(1-2).
- [4] Dieter Fensel, Frank van Hamelen, Wolfgang Reif, Annette ten Teije. Formal support for development of knowledge based systems[J]. Journal of Information Technology Management, 1998, 2(4).
- [5] Piet Hein Speel, Manfred Aben. Preserving conceptual structures in design and implementation of industrial KBSs[J]. Human-Computer Studies, 1998, 49(4).
- [6] D Fensel, F van Hamelen. A comparison of languages which operationalize and formalize KADS models of expertise[J]. The Knowledge Engineering Review, 1994, 9(2).
- [7] Angele J, Fensel D, Landes D, Studer R. Explorative Prototyping in KADS[R]. Research Report No. 214, University of Karlsruhe, Germany, 1991.
- [8] F van Hamelen, J Balde. $(ML)^2$, a formal language for KADS conceptual models[J]. Knowledge Acquisition, 1992, 4(1).
- [9] 王岚, 乐毓俊. 计算机自动推理与智能教学[M]. 北京: 北京邮电大学出版社, 2005.
- [9] Guus Schreiger. 知识工程和知识管理[M]. 史忠植, 等译. 北京: 机械工业出版社, 2003.