

基于 UML 类图模型的一致性检查方法

董庆超, 王智学, 张爱辉, 陈 剑

(解放军理工大学 指挥自动化学院, 江苏 南京 210007)

摘 要: UML 中的类图采用直观的图形化表示方法, 有效描述了待建系统的静态特征, 为系统设计人员发现系统模型中存在的非一致性和冗余等问题, 提供了有效的分析工具。但是对于复杂的系统, 完全依靠系统分析人员发现模型中存在的非一致性和冗余等问题是不现实的, 应当为建模工具赋予模型自动一致性检查功能。SHOIQ(D) 是描述逻辑家族中可判定的子集, 它在保证推理可判定的同时, 具备较强的描述知识能力。鉴于上述特点, 通过从 UML 类图图元中抽取语义, 用 SHOIQ(D) 形式化描述类图图元, 借助自动推理引擎, 从而使基于 UML 类图模型的自动一致性检查功能得到实现。根据该方法改进后的建模工具, 可以自动发现基于 UML 类图模型中存在的非一致性和冗余等问题。

关键词: UML 类图; 描述逻辑; SHOIQ(D)

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2008)10-0085-04

A Method of Inconsistency Detecting of Models Based on UML Class Diagram

DONG Qing-chao, WANG Zhi-xue, ZHANG Ai-hui, CHEN Jian

(Institute of Command Automation, PLA University of Science and Technology, Nanjing 210007, China)

Abstract: The class diagram of UML uses a visual manner to describe the static character of systems effectively. It provides software engineers a tool to find the inconsistency and redundancy of models. But a complex software system contains a great many of elements. It is hard to find the problems of models. So tools of modeling should have an automatic inconsistency detecting function. SHOIQ(D) is a decidable subset of description logic. It has a strong ability of knowledge description. Thereby, by extracting the semantic of components of UML class diagram, this paper uses description logic SHOIQ(D) to describe components of UML class diagram, and realizes the formalization of models. The transformed models can be automatically detected the inconsistency by the automated reasoning systems.

Key words: UML class diagram; description logic; SHOIQ(D)

0 引言

目前, 统一建模语言(UML)在领域知识描述、软件系统建模与分析等方面得到了广泛的应用。UML 中的类图通过图形化的表示方法, 直观地表示出待建系统的静态结构特征。UML 类图为系统开发人员在系统开发初期发现系统模型中存在的非一致性和冗余等问题, 提供了有效的分析工具。但是诸如 ROSE 这种支持 UML 统一建模语言的建模工具, 都没有提供模型的自动一致性检查功能。对于综合电子信息系统, 这样的大规模复杂系统建模与分析来说, 仅仅依靠模型设计人员检查模型中存在的非一致性和冗余等问

题是不可行的。因此, 具备模型自动检查功能, 将是今后系统建模工具发展的必然方向。为了形式化描述 UML 类图, 并且使模型的自动一致性检查得以实现, 文中在理解描述逻辑^[1]表示领域知识方法的基础上, 结合国内外的相关研究成果^[2-5], 提出一种由描述逻辑(SHOIQ(D))编码 UML 类图图元的规则(这里不考虑 OCL 约束和多元关联), 该规则不但可以有效地实现基于 UML 类图模型的形式化描述, 而且借助自动推理引擎, 从而使基于 UML 类图模型的自动一致性检查得以实现。

1 描述逻辑 SHOIQ(D)

描述逻辑是一族知识表示语言, 它以结构化和易理解的形式化来表示领域知识(Baader et al. 2003)。描述逻辑能很好地满足本体语言在语义、表达能力, 以及复杂性上的要求, 它具有正式的基于逻辑的语义和很

收稿日期: 2008-03-02

基金项目: 国家高技术研究发展计划(863 计划)(2007AA01Z126)

作者简介: 董庆超(1982-), 男, 硕士研究生, 研究方向为指控系统运筹分析及应用; 王智学, 博士生导师, 教授, 研究方向为指挥自动化理论与技术。

强的表达能力,是一阶谓词逻辑可判定的子集。当前的本体描述语言普遍将描述逻辑作为其逻辑基础^[1]。由描述逻辑构建的本体具备严格的形式化语义,消除了语义二义性和不确定性等问题。

SHOIQ(D)是描述逻辑家族中的可判定子集,它由 16 个概念构造子、6 个关系构造子以及具体域(D)组成^[1]。在描述逻辑众多可判定子集中,SHOIQ(D)具有较强的描述知识能力,并且已经有学者提出基于 tableaux 的 SHOIQ(D)的推理算法,证明了该算法的正确性^[6]。由于 UML 类图具有丰富的语义和大量的约束条件,在实现模型的形式化描述过程中,必须完整地保留原模型语义,因此文中采用 SHOIQ(D)作为形式化描述 UML 类图图元的语言。

2 用 SHOIQ(D)编码 UML 类图图元

文中主要从概念角度考虑 UML 类图的形式化描述规则,一些从工程实现角度考虑的 UML 类图特征,例如:公有、保护、私有等,这里不作讨论。下面介绍 UML 类图图元基于 SHOIQ(D)编码规则。

2.1 类 Class

类是一系列具有相同属性、操作、关系的对象集合的总称^[7]。在类图中,类由一个矩形表示,矩形分为三部分,分别对应于类的名称、属性和操作。UML 类图中的类与 SHOIQ(D)的原子概念具有良好的对应关系,不同的类在 SHOIQ(D)中可以分别用不同的原子概念编码。类 C 具有一个 T 类型属性 a , 在 SHOIQ(D) 中编码为:

$$C \subseteq a.T$$

其中类 C 在 SHOIQ(D) 中用原子概念 C 编码,类的属性 a 用原子关系 a 编码,属性的类型 T 用原子概念 T 编码(如果类型 T 是一些常见的类型(诸如整型、字符串型), T 可以用具体域 D 所提供的相应数据类型编码)。

类 C 具有无参数且返回值为 R 的操作 $f():R$, 在 SHOIQ(D) 中编码为:

$$C \subseteq \forall r_{f()}.R \cap (\leq 1r_{f()}.T)$$

其中类 C 的无参数操作在 SHOIQ(D) 中用原子关系 $r_{f()}$ 编码,操作的返回值用原子概念 R 编码。

类 C 具有一个含多元参数,多个返回值的操作 $operation = (op, C, (p_1, \dots, p_m), (t_1, \dots, t_n))$, 在 SHOIQ(D) 中编码为:

$$op \subseteq \forall r_1.c \cap (\leq 1r_1) \cap \forall r_2.p_1 \cap (\leq 1r_2) \cap \dots \cap \forall r_{m+n+1}.t_n \cap (\leq 1r_{m+n+1})$$

其中类 C 用原子概念 c 编码,类的操作用原子概念 op 编码,参数和返回值分别用原子概念 p_i, t_i 编码,

操作与类、参数、返回值之间的所属关系用原子关系 r_i 编码。

2.2 二元关联 Binary association

二元关联是一种结构化的关系,指一种对象和另一种对象有联系,使一个对象可以访问另一个对象类的公共属性和操作^[7]。类 C_1, C_2 之间存在有向二元关联(方向由 C_1 指向 C_2),在 SHOIQ(D) 编码为:

$$C_1 \subseteq \forall A.C_2$$

其中类 C_1, C_2 分别用原子概念 C_1, C_2 编码,关联用原子关系 A 编码, A 的定义域是概念 C_1 ,值域是概念 C_2 。

关联双方的重数约束可以编码为:

$$C_1 \subseteq (\geq \min A.C_2) \cap (\leq \max A.C_2)$$

$$C_2 \subseteq (\geq \min A^{-}.C_1) \cap (\leq \max A^{-}.C_1)$$

上述表达式的语义为:在关联关系 A 中,定义域概念 C_1 实例的重数大于最小值 \min ,小于最大值 \max ;值域概念 C_2 实例的重数大于最小值 \min ,小于最大值 \max 。

在关联关系中存在一类特殊的关系——链接属性,它实际上把链接关系也看成了一个类^[7]。例如,图 1 中每对相关的学生对象和课程对象都包含同一个年级对象,它们是因拥有同一个年级对象发生关联的。

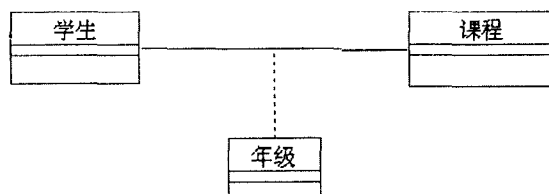


图 1 链接属性

链接属性在 SHOIQ(D) 中编码为:

$$Assoc \subseteq (\geq 1R_1) \cap (\geq 1R_2) \cap \exists R_1.C_1 \cap \exists R_2.C_2$$

其中链接类用原子概念 $Assoc$ 编码,类 C_1, C_2 用原子概念 C_1, C_2 编码,链接类与类 C_1, C_2 之间的关联分别用原子关系 R_1, R_2 编码。

关联双方的重数约束可以编码为:

$$Assoc \subseteq (\geq \min R_1.C_1) \cap (\leq \max R_1.C_1)$$

$$Assoc \subseteq (\geq \min R_2.C_1) \cap (\leq \max R_2.C_1)$$

上述表达式的语义为:在链接属性 $Assoc$ 中,概念 C_1 实例的重数大于最小值 \min ,小于最大值 \max ;概念 C_2 实例的重数大于最小值 \min ,小于最大值 \max 。

2.3 概括关系 Generalization

概括是一种继承关系,它是指两个模型元素之间的继承关系,使一个类可以继承另一个类的公共和保护属性和操作。通常称继承类为子类,而被继承类为

父类^[7]。概括关系 Generalization 在 SHOIQ(D)中编码为:

$$C_1 \subseteq C$$

其中子类 C_1 用原子概念 C_1 编码,父概念用原子概念 C 编码。

2.4 弱聚合关系 Aggregation、强聚合关系 Composition

强聚合关系与弱聚合关系(如图 2,3 所示)实际上是一种特殊的关联关系,它们都描述了一种整体与部分之间的关系,区别在于强聚合中要求部分类的实例只能出现在唯一的整体类实例中^[7]。



图 2 弱聚合关系

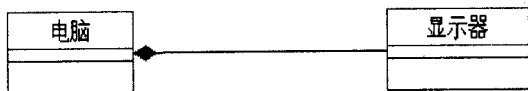


图 3 强聚合关系

弱聚合关系在 SHOIQ(D)中具有与有向二元关联关系相同的编码,原子关系 A 的定义域是整体概念,值域是部分概念。

强聚合关系的编码要在弱聚合关系编码的基础上增加以下约束:

$$C_1 \subseteq (= 1 A^{-1}. C) \cap \forall A^{-1}. C$$

其中整体类用原子概念 C 编码,部分类用原子概念 C_1 编码, C_1 与 C 的聚合关系用原子关系 A 编码, A 的定义域是整体概念 C ,值域是部分概念 C_1 。SHOIQ(D) 表达式的语义为:成员概念 C_1 只能聚合在整体概念 C 中,并且聚合关系 A 中定义域的重数等于 1,也就保证了一个部分实例仅能聚合于唯一的整体实例之中。

2.5 约束条件

不相交约束 Disjointness constraint 和覆盖约束 covering constraint 是限定类图模型要素之间关系的重要约束条件。

不相交约束在 SHOIQ(D)中编码为:

$$C_i \subseteq \bigcap_{j=i+1}^n \neg C_j \quad 1 \leq i \leq n-1$$

SHOIQ(D) 表达式的语义为:对于概念 C_i 与 C_j ,如果 $i \neq j$,那么 C_i 与 C_j 是互不相交的概念,即: C_i 的实例集与 C_j 的实例集交集为空。

覆盖约束在 SHOIQ(D)中编码为:

$$C \subseteq \bigcup_{i=1}^n C_i$$

SHOIQ(D) 表达式的语义为:概念 C 的实例集是某一组概念 C_i 实例集并集的一个子集。

3 UML 类图转换实例

下面用一个实例说明如何应用上述编码规则,实现基于 UML 类图模型的形式化描述。

图 4 描述了一个 Pizza 的组成模型,每个 Pizza 由唯一的 PizzaBase 和 PizzaTopping 组成。PizzaTopping 具体分为多种不同类型。每一个类型的 PizzaTopping 有唯一的口味,但是不同类型的 PizzaTopping 口味各不相同。

按照上述 UML 类图图元编码规则,该模型对应的 SHOIQ(D)知识库是:

$$\text{Pizza} \subseteq \forall \text{HasBase}. \text{PizzaBase}$$

$$\text{Pizza} \subseteq = 1 \text{HasBase}. \text{PizzaBase}$$

$$\text{Pizza} \subseteq \forall \text{HasTopping}. \text{PizzaTopping}$$

$$\text{Pizza} \subseteq = \text{HasTopping}. \text{PizzaTopping}$$

$$\text{PizzaBase} \subseteq (= 1 \text{HasBase}^{-1}. \text{Pizza}) \cap \forall \text{HasBase}^{-1}. \text{Pizza}$$

$$\text{PizzaTopping} \subseteq (= 1 \text{HasTopping}^{-1}. \text{Pizza}) \cap \forall \text{HasTopping}^{-1}. \text{Pizza}$$

$$\text{MeatTopping} \subseteq \text{PizzaTopping}$$

$$\text{CheeseTopping} \subseteq \text{PizzaTopping}$$

$$\text{SeaFoodTopping} \subseteq \text{PizzaTopping}$$

$$\text{Hot} \subseteq \text{Spiciness}$$

$$\text{Medium} \subseteq \text{Spiciness}$$

$$\text{Mild} \subseteq \text{Spiciness}$$

$$\text{Hot} \subseteq \neg \text{Medium} \cap \neg \text{Mild}$$

$$\text{Medium} \subseteq \neg \text{Mild}$$

$$\text{PizzaTopping} \subseteq \forall \text{HasSpiciness}. \text{Spiciness}$$

$$\text{PizzaTopping} \subseteq = 1 \text{HasSpiciness}. \text{Spiciness}$$

$$\text{Spiciness} \subseteq \geq 1 \text{HasSpiciness}^{-1}. \text{Spiciness}$$

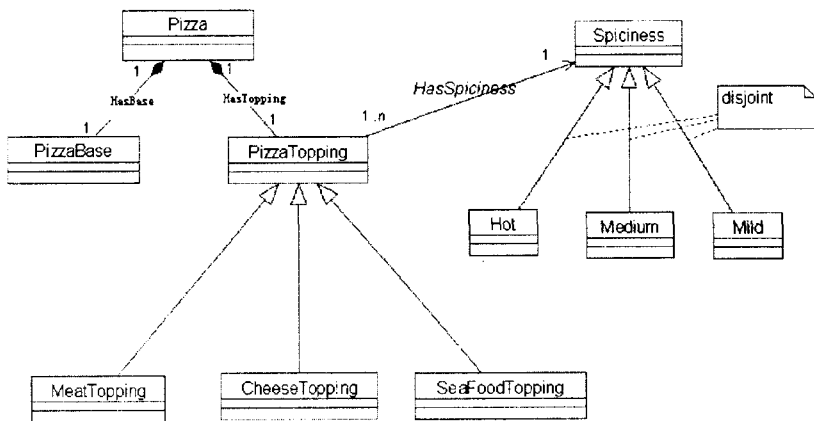


图 4 Pizza 模型片段

上述 SHOIQ(D)知识库完整地保留了原模型的全部信息,实现了模型由图形化向形式化描述的转换过程。

转换后的模型,借助自动推理引擎,该引擎可以通

过文献[7]提供的算法实现,不但可以自动划分模型中类之间的包容、继承、等价等层次关系,而且对于模型中存在的 inconsistency、冗余等问题提供了自动检查功能。这些特点减轻了系统建模与分析人员的工作负担,为确保系统开发初期准确构建系统模型提供了可靠的技术支持。

4 结束语

在系统建模与分析工作中,UML 凭借它可视化的建模方式,赢得了系统设计人员的普遍青睐。但是这种方法构建的模型形式化程度比较低,不容易被计算机所理解,复杂模型中更加容易出现语义上的矛盾和冲突。这些问题对领域知识复用、需求获取、信息检索等应用都会带来不利影响。

文中借鉴描述逻辑(SHOIQ(D))描述知识的方法,对 UML 类图图元定义了基于 SHOIQ(D)的编码规则,通过该规则实现了基于 UML 类图模型的形式化描述,借助推理引擎,从而使模型的自动一致性检查得以实现。

下一步的工作是在建模工具中具体实现模型转换算法,根据基于 tableaux 的 SHOIQ(D)的推理算法^[7],

开发自动推理引擎,为建模工具添加模型自动形式化描述与一致性检查功能。

参考文献:

- [1] 陆建江,张亚非,苗 壮,等.语义网原理与技术[M].北京:科学出版社,2007.
- [2] Na H S, Choi O H, Lim J E. A Method for Building Domain Ontologies based on the Transformation of UML Models [C]//SERA2006. New York: IEEE, 2006.
- [3] Na H S, Choi O H, Lim J E. A Metamodel - Based Approach for Extracting Ontological Semantics from UML Models[C]//WISE 2006. Chongqing: [s. n.], 2006.
- [4] Van Der Straeten R. Inconsistency Management in Model - Driven Engineering—An Approach using Description Logics [D]. Brussel: Vrije Universiteit Brussel, 2005.
- [5] Berardi D, Calvanese D, De Giacomo G. Reasoning on UML class diagrams [J]. Artificial Intelligence 2005 (168): 70 - 118.
- [6] 蒋运承,汤 庸,王 驹,等.面向语义 Web 的描述逻辑 [J]. 模式识别与人工智能, 2007, 20(1): 48 - 54.
- [7] 王智学. ROSE 对象建模方法与技术[M]. 北京:机械工业出版社, 2003.

(上接第 84 页)

类 2 $\{x_1, x_2\}$;
 类 3 $\{x_5, x_8, x_9\}$;
 类 4 $\{x_4, x_6, x_7, x_{10}, x_{12}\}$;
 类 5 $\{x_{13}, x_{15}, x_{16}, x_{17}, x_{19}\}$;
 类 6 $\{x_{11}, x_{14}, x_{18}, x_{20}\}$;
 $x_i, i = 1, 2, \dots, 20$ 表示第 i 个数据行。

从聚类结果中可以看到,“丹阳市”被单独分为一类,原因在于,财务总收入作为一个重要特征指标被赋予极大的权值,而该市的年度财务总收入为 16.62 亿,远远高出其它各市。该实验突出体现了加权对聚类结果的影响,当然,在具体使用中,使用者在对特征指标加权时往往不会这样特殊。通过加权,使用者可以根据自己对各特征指标不同的关注度取得合理的聚类结果,而当各聚类指标的权值相等时,该算法与传统的 FCM 聚类结果相同。

3 结束语

提出了一种基于用户需求的加权模糊 C 聚类算法,它是以加权 Euclid 距离作为聚类的标准。通过加权,增加了聚类的灵活性,使聚类结果能更好地满足使

用者的需求。最后通过实验,验证了特征指标加权对聚类结果的影响。

参考文献:

- [1] Bezdek J. Pattern Recognition with Fuzzy Objective Function Algorithms[M]. New York: Plenum Press, 1981.
- [2] Xie X L, Beni G. A validity measure for fuzzy clustering[J]. IEEE Trans Patt Anal and Machine Intell, 1991, 13(8): 841 - 847.
- [3] 岳士弘,李 平,宋执环,等.自适应模糊聚类[J]. 浙江大学学报, 2004, 38(10): 1280 - 1284.
- [4] 诸克军,苏顺华,黎金玲.模糊 C-均值中的最优聚类与最佳聚类数[J]. 系统工程理论与实践, 2005(3): 52 - 61.
- [5] 胡宝清. 模糊理论基础[M]. 武汉:武汉大学出版社, 2004: 165 - 169.
- [6] Bezdek J C, Pal S K. Fuzzy models for Pattern recognition [M]. New York: Plenum Press, 1992: 94 - 96.
- [7] 李为民,朱永峰,付 强.基于自适应模糊聚类分析的目标冗余信息处理[J]. 计算机应用, 2005, 25(4): 949 - 951.
- [8] 陈 治. 使用 SPSS 软件进行因子分析和聚类分析的方法 [J]. 市场研究, 2006(6): 45 - 48.