

面向对象方法中的整体与部件关系*

宋柔 王鑫

(北京工业大学计算机学院, 100044)

摘要 在面向对象的方法中, 引入了关于整体和部件的关系, 并实现在特定领域知识库建造工具OOKS之中.

关键词 面向对象的方法, 知识库建造工具, 整体, 部件

分类号 TP18

设计知识库建造工具或专家系统生成工具中, 最关键的问题之一是如何把握好通用与专用的关系. 我们研制了一种面向对象的特定领域知识库建造工具(OOKS). 它的基本思想是: 在尽量照顾通用性的同时, 根据所面对的领域、任务的特点, 提炼出一些该范围内专用的概念, 扩充和修改面向对象的知识表示方法, 达到方便地表示知识、高效地利用知识的目的. 我们的工作包括: 在面向对象的方法中引入关于整体和部件关系的专用表示方法和处理方法, 引入实例状态的概念和实例状态化方法, 采用黑板模型的自动归约控制, 等等. 本文介绍关于整体和部件关系方面的工作.

1 整体—部件关系的性质

1.1 与父类—子类关系相似的特征

(1) 普遍存在性. 只要个体有结构, 个体内就有整体—部件关系.

(2) 传递性. 任一个体的部件的部件仍是该个体的部件.

由传递性, 父类—子类关系和整体—部件关系分别构成树形关系(严格地说, 它们都不是真正的树形关系, 因为一个子结点可能有多个父结点, 但它们都不带环). 这两种树形关系向两个不同方向伸展, 又相互关联, 形成一种立体的网状结构. 图1和图2例示了这种结构, 其中向下的箭头表示父类—子类关系, 向右的箭头表示整体—部件关系.

用小写的 a、b、c 表示个体(即实例), 用大写的 A、B、C 表示个体的集合(即类), 则这种立体网状结构中的关系可以表述如下:

非形式定义: whole-part(a, b) \longleftrightarrow b 是 a 的部件 \longleftrightarrow a 是 b 的整体

公理: whole-part(a, b) \wedge whole-part(b, c) \rightarrow whole-part(a, c)

定义: FATHER-SON(A, B) \longleftrightarrow A \supset B \longleftrightarrow $\forall (b) [b \in B \rightarrow b \in A]$

CLASS-INSTANCE(A, a) \longleftrightarrow a \in A

WHOLE-PART(A, B) \longleftrightarrow

* 清华大学化学工程联合国家实验室资助项目.

收稿日期: 1994-09-29.

$$\forall (a)[a \in A \rightarrow \exists (b)[b \in B \wedge \text{whole-part}(a, b)]]$$

定理: FATHER-SON(A, B) \wedge FATHER-SON(B, C) \rightarrow FATHER-SON(A, C)
 FATHER-SON(A, B) \wedge CLASS-INSTANCE(B, b) \rightarrow CLASS-INSTANCE(A, b)
 WHOLE-PART(A, B) \wedge WHOLE-PART(B, C) \rightarrow WHOLE-PART(A, C)
 WHOLE-PART(A, B) \wedge CLASS-INSTANCE(A, a)
 $\rightarrow \exists (b)[\text{CLASS-INSTANCE}(B, b) \wedge \text{whole-part}(a, b)]$
 FATHER-SON(A, B) \wedge WHOLE-PART(A, C) \rightarrow WHOLE-PART(B, C)

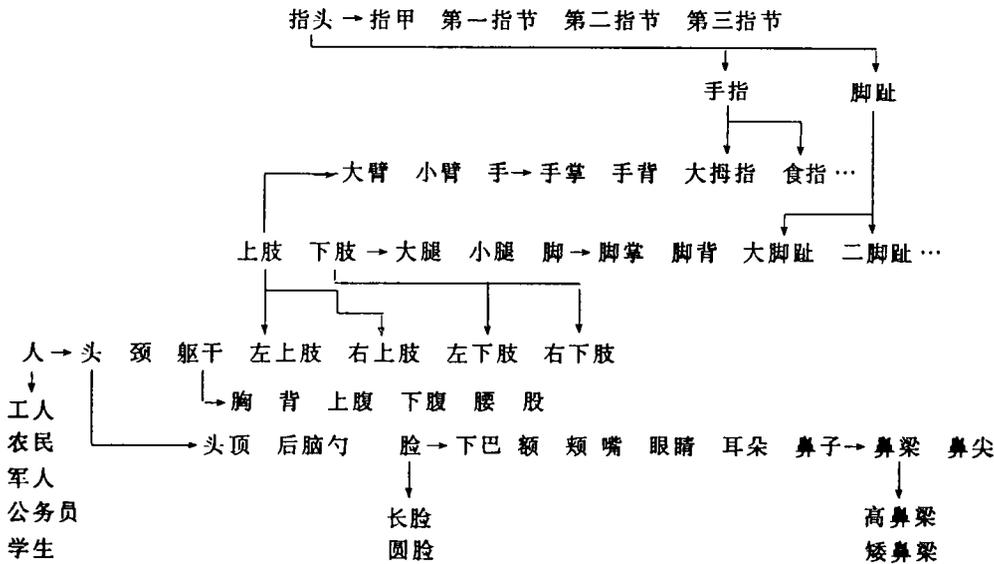


图1 人体

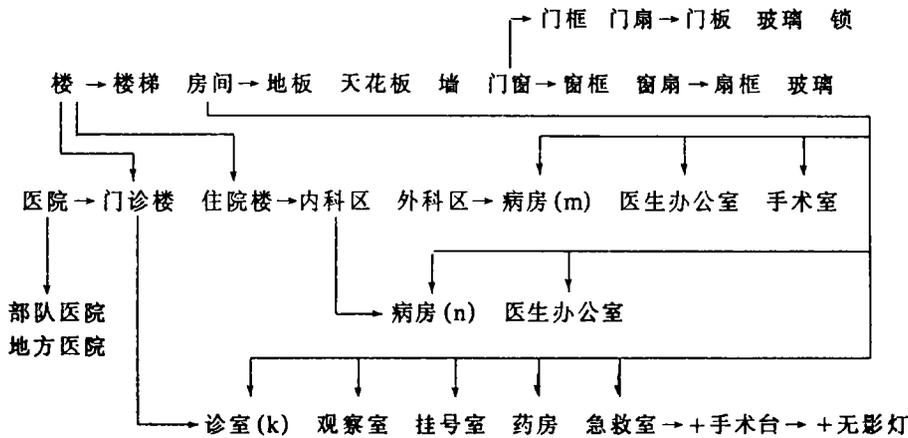


图2 医院

(3) 划分的不唯一性

父类-子类关系的划分是不唯一的。如,“人”的子类可以按人种分,按国籍分,按性别分等等。整体-部件关系的划分也不唯一。如:人的部件可分为头、颈、躯干、左上肢、右上肢、左下肢、右下肢,躯干又分为胸、上腹、下腹、背、腰、股。但人也可分为头、颈、上身、腰、下身,上身又分为胸、上腹、背、左上肢、右上肢,等等。

1.2 相对于父类-子类关系的继承性

一般情况下,同类的个体具有相同的结构性质,于是整体-部件关系可以在类的层面上描述,从而也能在父类和子类的关系中继承。即,子类可以继承父类的整体-部件关系,或者说,若一个集合内的所有个体都具有某种结构性质,则该集合的任意子集中的所有个体也都具有这种结构性质。这一点是与类中个体的其它性质相同的。另一方面,从父类继承的整体-部件关系也可能被屏蔽,这与类中个体的其它性质也是相同的。

2 引入整体-部件关系的好处

因为该关系有以上特性,故应把该关系抽象出来,在面向对象系统中专门给定一种表示方法和实现方法,以便利用这些特性。

(1) 用简化路径来引用部件

在图1所示的人体结构中,若不引入整体-部件关系,部件将作为类的对象成员对待。当我们引用张三的鼻尖时,一般需要用类似于下面的形式:

(((张三.头).脸).鼻子).鼻尖)

即必须说清楚从张三到其鼻尖要经过结构路径“张三→头→脸→鼻子→鼻尖”。这种表示法带有传统程序设计语言的风格。

我们的系统中引入了整体-部件关系,并利用其传递性和继承性设计实现了一个由整体找部件的算法,故可使用“张三→鼻尖”这种既简单又明确的引用方式,这是知识表示的风格。

路径的简化需避免歧义。如“张三→左上肢→手”不能简化成“张三→手”,因为这可能同“张三→右上肢→手”相混淆。图2中“人民医院→急救室→门→玻璃”不能简化成“人民医院→急救室→玻璃”,更不能变成“人民医院→玻璃”,因为急救室还有窗玻璃,其它房间也有玻璃。当然,可以借助这种歧义发展一套基于整体-部件关系的模糊查询的理论和技巧,但那是本文以外的工作。

注意,如果不在整体-部件关系范围内,当引用某实例的普通对象成员的对象成员时,一般不能省略中间路径,如“张三→朋友→母亲”不能简化成“张三→母亲”。

(2) 用多种方式来说明部件

由于整体分成部件的方式并不唯一,采用一般的面向对象的方法来说明部件有时会遇到困难。如1的例子,把躯干和上身这两个部件都列作“人”类的普通对象成员是有问题的,因它们都有对象成员“胸”,但没有方便的手段说明这两个“胸”是同一个对象。可是如果这两个对象成员中只说明一个,则另一个对象就无法被引用。

我们的系统允许同一对象有多种部件分解方式,同时约定不同分解方式下名称相同的

部件表示同一对象, 如此便解决了这一问题.

3 引入整体—部件关系的方法

在 OOKS 中, 类定义中包含整体说明和部件说明, 它们都是类名的表. 在机内存结构的层面上, 这种说明相应于整体和部件之间的双向指针. 如果该类的个体中包含多个某种部件, 则需有部件个数的说明, 其数值在实例化时被给出.

实例成员的表达方式是一个路径表, 其第一项是实例名(许多情况下可省略), 最后一项是要访问的成员名, 中间各项是部件名. 如由图 2, 人民医院住院楼外科区 3 号病房墙的颜色简化表示为

(人民医院 外科区 (病房 3) 墙 颜色)

基于这种简化的表示方式, 对成员的访问需用一种搜索算法. 这样做方便了建立知识库的专家, 但降低了知识库系统的运行效率. 为弥补后一方面的缺陷, OOKS 提供了一种翻译机制, 可以静态地把成员路径表中被省略的中间部件名和提供继承的类名补充进去.

例如, 上面的例子补全后成为

(人民医院 (部件 住院楼) (部件 外科区) (部件 (病房 (3))
(父类 房间) (部件 墙) 颜色)

于是运行时不必搜索, 从而提高了效率.

The Relationship Between the Whole and Part Domains Object-oriented Approach

Song Rou Wang Xin

(Computer Institute, Beijing Polytechnic University, 100044)

Abstract This paper adds the concepts of whole and part domains and their relationship to the object-oriented approach. Based on this approach, OOKS, a tool for building knowledge bases of special domains, is developed.

Keywords object-oriented approach, building toll for knowledge base, domain