

# 数据建模方法的比较与分析

## Comparison and Analysis of Data Modeling Method

朱 宇 (清华大学软件学院 100003)

**摘要:**本文选取了 Richard Barker 表示法、IDEFIX 表示法、UML 表示法三种典型的数据建模方法,试图从完整性和可读性两个方面,通过对相同角度上三种典型建模方法的不同特点进行比较和分析,找出它们的适用范围。

**关键词:**数据建模方法 适用范围 完整性 可读性

### 1 引言

数据模型是对信息系统中客观事物及其联系的数据描述,它是复杂的数据关系之间的一个整体逻辑结构图。数据模型不但提供了整个组织藉以收集数据的基础,它还与组织中其他模型一起,精确恰当地记录业务需求,并支持信息系统不断地发展和完善,以满足不断变化的业务需求。对于任何一个信息系统来说,数据模型都是它的核心和灵魂。

本文选取了 Richard Barker 表示法、IDEFIX 表示法、UML 表示法三种典型的数据建模方法,试图从完整性和可读性两个方面,通过对相同角度上三种典型建模方法的不同特点进行比较和分析,找出它们的适用范围。其中,对于完整性的衡量,主要是从实体类型与属性、关系、唯一标识、子类型与超类型以及关系间的约束等几个角度来分析的;对于可读性的衡量,则是从其对关系线和实体类型框的图形表示、图形设计优劣的一般准则等角度来考虑的。

### 2 三种典型建模技术的比较

为了便于比较,本文将用这三种方法考察同一个实例。

#### 2.1 Richard Barker 表示法

Richard Barker 表示法最初是由英国的 CACI 咨询公司提出的,现在是欧洲所采用的 SSADM 方法的一部分。后来得到了 Richard Barker 的大力倡导,为 Oracle 公司采用,成为其“CASE \* 方法”的一部分。

图 1 给出了该表示法的一个示例。

(1) 实体类型与属性。Barker 表示法中的实体类

型是用圆角矩形表示的。实体类型的名称及属性标注在矩形框内。在正式的表示法中,用#号表示唯一标识属性,开口小圆圈表示可选属性,闭合小圆圈表示必选属性,但在实际运用(包括本文示例)中,经常是用小原点表示所有可选或必选属性。

(2) 关系。Barker 表示法中的关系是用直线(一半是实线,一半是虚线)表示的。关系的名称是用直线两端标注的介词或介词短语来表示的。关系的基数性是用直线两端有没有乌鸦角符号来表示的,关系的可选择性是则用关系对应侧直线的虚实而确定的。

(3) 唯一标识。Barker 表示法中唯一标识是用属性名前的#号来表示的。

(4) 子类型。Barker 表示法中子类型是用超类型框中嵌套的方框来表示的。其中,各个子类型之间是互斥的,不允许出现子类型的交叠。同时各个子类型又都是完备的,认为所有子类型能组成超类型的全集。

(5) 关系间的约束。Barker 表示法中关系之间的唯一约束就是异或。横跨两条关系连线的弧线表示某实体类型的每次出现与 1 个或多个其他实体类型的出现相关。

#### 2.2 IDEFIX 表示法

IDEFIX 表示法是为美国联邦政府的很多部门所采用的一种数据建模技术。在 IDEF 方法家族中,IDEF1 是最早着手研究和制定的标准之一,它主要用作开发“信息模型”。而 IDEFIX 标准则是在 IDEF1 标准的基础上进行的扩展,它在表示符号和语义方面进

行了较大的改动,主要增加了概括和聚合语义,于1985年正式发布。1993年 IDEFIX 成为美国政府的处理标准文件 FIPS (Federal Information Processing Specification) 之一。

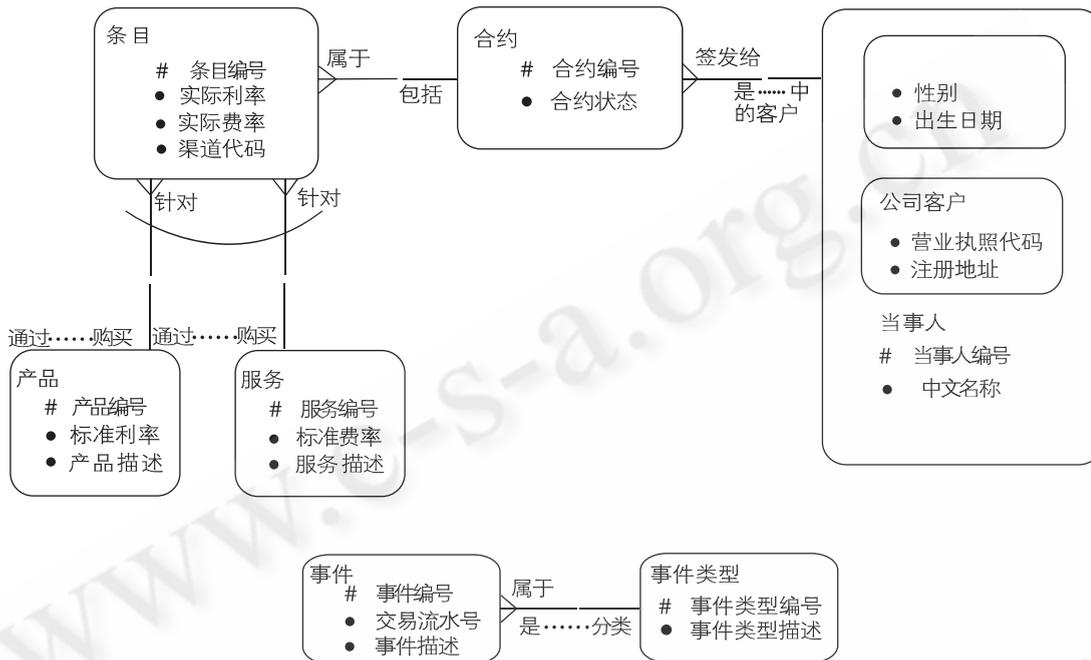


图 1

图 2 给出了该表示法的一个示例。

(1) 实体类型与属性。IDEFIX 表示法中的实体类型是用圆角或直角矩形表示的,不同的形状表示不同的实体类型。圆角矩形表示依赖型实体,它们的唯一标识中至少包括 1 个到另一个实体类型的关系。直角矩形表示独立型实体,它们的唯一标识中没有来自其他实体类型成分的实体类型。实体类型的名称标注在矩形框外,实体类型的属性标注在矩形框内。矩形框内分为两个部分,上部内是起标识作用的属性,或者称为主键,下部内则是不做标识用的属性。

(2) 关系。IDEFIX 表示法中关系是用直线(全实线或者全虚线)表示的。关系的名称是用直线两端标注的动词或动词短语来表示的。在 IDEFIX 表示法中,关系的基数性和可选择性是不能向 Barker 表示法那样分别解析的,全实线或全虚线以及两端的圆形、菱形符号的 22 种组合分别代表不同的相邻实体类型的基数性和可选择性关系。

(3) 唯一标识。IDEFIX 表示法中唯一标识是用实体类型框上部内的属性来表示的,它将作为在关系型数据库中实现的主键。

(4) 子类型。IDEFIX 表示法中子类型是用专门的

实体类型框来表示的。这些表示子类型的矩形框是与表示超类型的矩形框相分开的,二者之间通过“是一个”关系连接起来。子类型有两种:完备的子类型,下面带有两条水平线的圆圈,它表示父类型的所有出现都必须是其某个子类型的出现;不完备的子类型,下面

带有一条水平线的圆圈,它表示子类型并不能表示其所有超类型的所有可能出现。

(5) 关系间的约束。IDEFIX 表示法中没有明确的表示关系间约束的手段。在这种表示法中,不是声明“A”与“B”或“C”有关,而是要定义实体类型“D”,然后再运用子类型的表示方法来表述出相关约束。

### 2.3 UML 表示法

面向对象已成为软件领域的主流技术,UML 的诞生与发展更是对于整个软件开发流程的改善起了相当关键的作用。20 世纪 80 年代,出现了一大批面向对象的分析与设计方法,如 Booch 方法、Coad/Yourdon 方法、Firesmith 方法、Jacobson 的 OOSE、Martin/Odell 方法、Rumbaugh 等人的 OMT、Shlaer/Mellor 方法等等,设计与建模技术比较纷乱,众说纷纭。1997 年 Rational 公司与其他公司一起共同推出了 UML1.0,形成了统一的建模语言,2003 年又推出了 UML2.0,2.0 可

以让开发者在设计软件的过程中分解成不同层次进行设计,同时 2.0 完整度非常高,弥补了 1.0 版在做软件设计开发方面的缺陷。

图 3 给出了该表示法的一个示例。

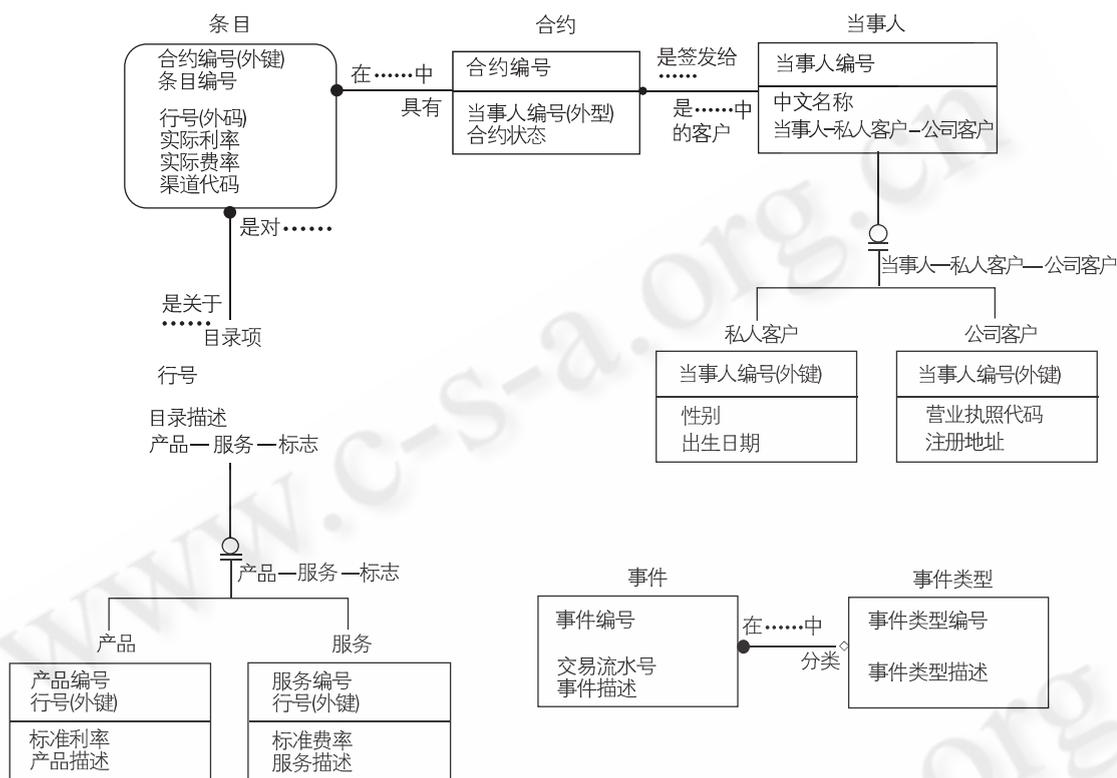


图 2

(1) 实体类型与属性。UML 表示法中的实体类型被称为类,是用直角矩形表示的。矩形内部分为三个部分,顶部是类名称,中间列出若干个属性,底部是对行为的描述。在 UML 表示法中,属性除名称外还可以有选择地标注构造型、可视性、多样性、类型、初始值等内容。

(2) 关系。UML 表示法中关系被称为关联,不是用图形符号,而是用文字来表述所有信息的。关联的命名主要有两种形式:一种是用简单的动词短语来对整个关联命名;另一种则是在每一端定义描述该类在此关联中所扮演的角色。关联的基数性和可选择性是通过: <下限>...<上限>来表示的,其中 <下限>表示的是可选择性,<上限>表示的是基数性。

(3) 唯一标识。UML 表示法中很少提及唯一标

识,当某个类中的对象的行为要求确定另一类的某个特定出现时,将在需要它的那个实体类型旁边的方框中标出用于确定该出现的那个属性。

(4) 子类型。UML 表示法中子类型是用专门的实

体类型框来表示子类型的。每个子类型框都放在其超类型框之外,子类型框通过“是一个……”关系连接到超类型,每个子类型的出现都是其超类型的一次出现。

(5) 关

系间的约束。UML 表示法中用关联对间的虚线来表示约束。如果标有 { xor } 或 { or },则表示

为异或。

### 3 三种典型建模技术的分析

#### 3.1 Richard Barker 表示法

在需求分析中宜采用 Barker 先生的数据建模表示法,这是因为它外观简洁,最易于向用户进行表述,对于非技术人员来说,清晰的模型图更容易理解。在具有同等完整性的表示法中,Barker 表示法是最简单、最清楚的一种。它使用的符号种类相对较少,采用表示出的模型图比较清楚,要求用户认知的成分种类也比较少。

在实体的类型与属性的表示上,其他一些表示法额外地添加了一些符号,如 IDEF1X 表示法中,用不同的

符号来表示“依赖”和“独立”实体类型,同时还需要在不同的实体类型端使用不同的关系符号。UML 表示法中根据参照完整性的不同而用两种特殊的符号来致命“属于”和“包括”的关系。这些所添加的符号都增加了模型图的复杂性,使其更难以理解,而对于用户而言,却并没有增加通过 Barker 表示法的简单符号和名称就能表示的信息之外的有效内容。

确、易于理解的介词或介词短语,这方面其他表示法都是采用动词或动词短语作为关系名称的。由于在自然语言中,介词就是用以描述关系的,所以 Barker 表示法更为合适。动词描述的是动作而不是关系,用动词来描述关系其实是对两个实体类型间的动作来下定义,在数据模型中简单地描述关系本身更为合适,而动词则更适用于功能模型中。

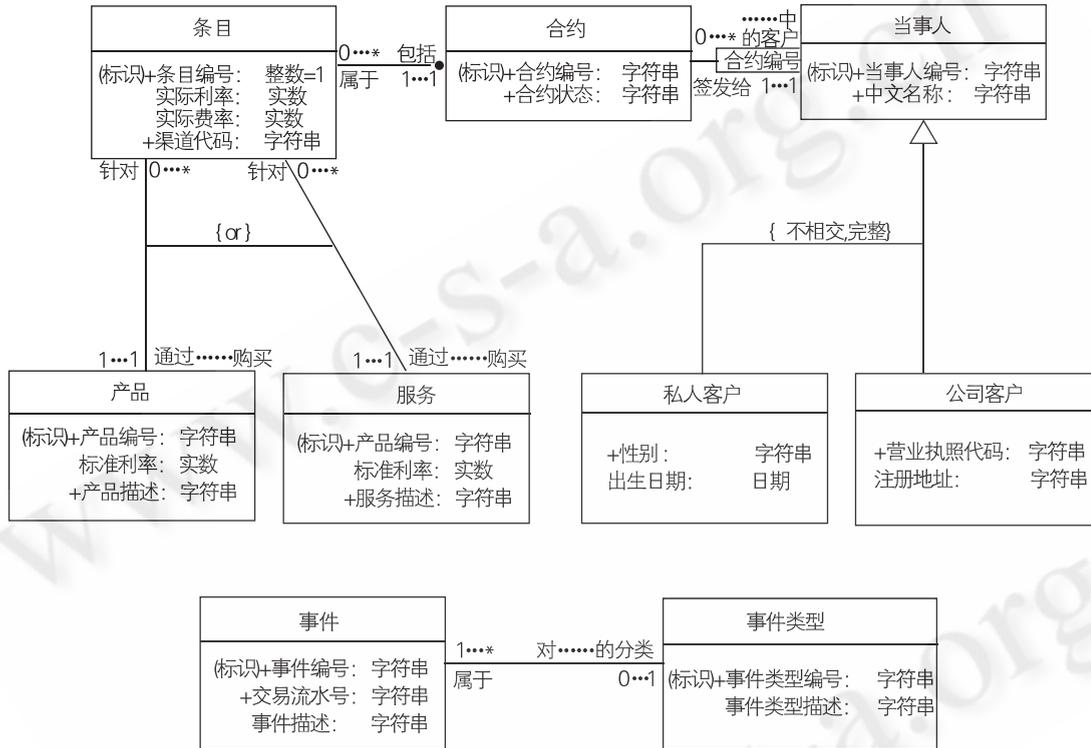


图 3

在关系的表示上,对于大多数用户来说,可选择性是最重要的特性,在 Barker 表示法中,可选择性是通过最明显的外观特点——关系线的虚实来表示的。IDEFIX 表示法中虽然也利用了关系线的虚实,但却是用来表达某个关系在多大程度上是唯一标识。对于关系的基数性,Barker 表示法只需要用乌鸦脚符号的有无就完全可以表示出某个关系的上限,而在其他一些表示法中,表示手段就要复杂得多。例如 IDEFIX 表示法中通过虚实线与其两端不同的圆形、菱形符号的 22 种组合分别表示不同的基数性和可选择性关系;UML 表示法中也需要通过详细定义上下限来表示。

在关系名称的表示上,Barker 表示法要求分析师对各种关系做出精炼的表述,而且要使用清楚、语法正

在子类型的表示上,Barker 表示法是唯一之中在超类型内部表示子类型的表示法,这样做一方面可以强调子类型是超类型的子集这一事实,另一方面也可以节省绘图空间。

在模型的可读性方面,其他表示法中都允许关系线任意弯折,从而使模型图显得更为复杂。而在 Barker 表示法

中对模型图的布局有着特定的要求,可以使关系线尽可能短、直。

### 3.2 IDEFIX 表示法

在关系数据库设计中宜采用 IDEFIX 表示法,这是因为对于技术人员而言,IDEFIX 表示法能表示出实现方式上的复杂细节,例如突出强调的外键存在,关系的可选择性和基数性的完整组合等。但对非技术人员而言,这种方法并没有遵从设计优良图形的原则,该方法可能较难掌握和使用。

在关系的表述上,IDEFIX 表示法中所用的符号并不与它们所要表达的模型中的概念完全相符。在这种表示法中,本应由一个符号表示的概念却要求同时使

用多个符号来表示,而且某一特定情况可以由不只一组符号来表示,而同一符号在不同上下文中又可能表示不同的含义。某一特定情况到底该用哪个符号来表示在很大程度上取决于上下文和该关系的实现方式,而不是取决于该情况本身。这些做法导致要教会非技术人员读懂 IDEFIX 模型图是极为困难的。

IDEFIX 表示法中,直线的虚实是用来表示某一实体类型是否构成另一实体类型的唯一标识(主键)。这将要求分析师在开始考察模型中各关系的可选择性或基数性之前,先分析依赖性。但在实际建模时,分析师通常总是先考察哪些实体类型是其他实体类型所需要的,以及其中涉及多少次出现等问题,通常是在后期才考虑主键或标识等细节。另外对这种模型的修改也是非常困难的,如果基数性或可选择性方面出现了某个小错误,就必须改动好几个符号。

### 3.3 UML 表示法

在面向对象的设计中宜采用 UML 表示法,这是因为它更完整、更详尽。通过其他数据建模方法所不具备的额外表达能力,使其不仅适合于系统的逻辑分析模型的表述,也同样适用于物理设计模型的表述。

在实体属性的表述上,UML 表示法较其他表示法能够更详细地对属性做出描述。可以标注出包括构造型、可视性、名称、多样性、类型、初始值等一个或多个内容,而在 Barker 及 IDEFIX 中对于实体属性仅标注出名称。

在关系的表述上,对并非是两个关联间的简单关系的商业规则,UML 表示法引入了一种小标志,该标志中可以包括描述任何商业规则的文字。对关系的可选择性和基数性的表示,UML 表示法能够表述更为复杂的上限,可以具体说明某个实体类型的出现可能与另

一个实体类型的 1、7 ~ 9 或 10 次出现有关。

在关系间约束的表述上,UML 表示法用两个关联之间的简单直线代替了 Barker 表示法中对关系之间约束的表述,而且可以在这种直线上加评注,从而描述两个关联之间的任何关系。

## 4 总结

不同的数据建模方法适用于不同使用者对数据模型的不同要求,在需求分析工作中采用 Barker 方法更为合适,为支持面向对象的设计,宜采用 UML 方法,如果要采用关系型数据库,则应采用 IDEFIX 方法。

## 参考文献

- 1 Data Modeler's Workbench – Tools and Techniques for Analysis and Design, Steve Hoberman, 2001.
- 2 Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe.
- 3 Object – Oriented Analysis and Design with Application, Booch G. , 1994.
- 4 Requirements Analysis, David C. Hay, 2003.
- 5 A Framework for Information Systems Architecture, J. A. Zachman, 1987.
- 6 The Information Framework, R. Evernden, 1996.
- 7 信息系统建模, 张维明、刘忠、肖卫东, 2001。
- 8 IDEF 建模分析和涉及方法, 陈寓六, 1999。
- 9 面向对象软件工程, 陈市鸿、彭蓉, 1999。
- 10 数据仓库模型构建技术, 曹东启、曹翊, 2002。