

软件工程

CASE

数据交换模型

计算机科学 1995 Vol. 22 No. 3

49-52

CASE 数据交换模型

郭江 金茂忠

廖越红

(北京航空航天大学软件工程研究所) (中国科学院计算中心)

TP311.5

摘要 A very important problem in the CASE is data interchange. It is the key part of the components of CASE environment communication and coordination. The paper mainly discusses the architecture, representation, and future evolution of the CASE data interchange model.

关键词 CASE, Data interchange model, Entity relationship model.

一、引言

CASE 数据交换模型是 CASE 环境中工具间进行数据信息交换的关键。为了使 CASE 工具能够在控制集成、数据集成、显示集成这三方面成为一个整体,工具间的数据交换和通讯是必不可少的。为了能有效地实现这种通讯机制,势必要建立一个数据交换模型。

CASE 数据交换模型的目标是提供一个数据交换的标准集,以便 CASE 工具能够以标准的格式交换信息,这不仅需要格式本身的定义,而且需要保持交换信息的语义一致性,即建立一个数据交换模型。

数据交换模型必须是一个中性模型,不依赖于某个具体方法,但必须支持多种方法所需的语义,而且还必须能进行扩充以支持新的语义。

在数据交换模型方面还有许多工作要做,如怎样支持 PCTE 的某些概念(模式定义集 SDS 和成对关系)、数据交换模型的语法如何映射为仓库(Repository)的输入/输出语言等等。

二、模型的体系结构

1. 模型的语义分离

最初的建模工作是通过直接开发数据交换的语法完成的,因而没有将底层的模型和表达模型的语法分离开,所产生的结果是图形化表示和语义对象密切地结合在一起。但是,为了扩充和模型的灵活性,需要将语法、图形表示以及底层模型这三者加以分离。分离工作的首要任务是定义一个建模框架,以便定义要交换的信息语义。在这三者分离之后,就可以使用同一个建模框架来定义数据交换格式,而无

需参照用框架定义的语义集。这样,两方面的工作就可以独立完成了。这里我们提出了一个四层模型的建模框架,下面将进行详细地讨论。

2. 四层模型的建模框架

建模框架的每一层用于定义下一层的内容和语义。CASE 工具被开发人员、分析人员以及程序员用来定义正在开发的系统模型。该模型本身最后实例化为运行系统中的单个进程和数据对象,这一层称为数据层,是四层模型的最底层。由各工具用户所开发的模型处于第三层,这一层称为模型层。在开发过程中,CASE 工具所包含的系统模型信息自定义在第二层中,这一层称为元模型层。这是工具间要交换信息语义的形式化定义层,该层本身要使用的固定表示称为元元模型,这形成了框架的顶层。四层模型的建模框架如图 1 所示。

3. 交换信息的语义和表示的分离

CASE 工具以各种方式给工具用户提供信息。工具与工具的图形表示各不相同;如,实体模型目前就已开发出了多种表示。在建模的初期,通常要花许多时间来讨论对象间的准确关系,但这并不是因为对象的底层语义有多大的区别,而是因为不同工具所用的表示方法有较大差异。

因此,数据交换模型必须将交换信息的底层语义和它的表示进行分离,并提供一个工具来给用户显示底层的表示信息。这种方法有几个好处:(1)底层语义的建模由于不包含所有表示信息而得到大大简化。(2)表示信息可以用一种标准方式建模,而不必考虑要表示的语义。这样,一个底层语义集可以有多个表示信息集。

郭江 博士生,主要研究领域:软件工程,软件工程环境,数据库等。金茂忠 系主任,教授,主要研究领域:软件工程,软件工程环境,语言处理系统等。

4. 语法和编码的分离

在提供了固定的元元模型之后,所有语法就可以以它为基础来进行定义。该语法定义了元元模型框架中建模对象的语法表示形式。它将使用元元模型对象作为定义中的标识符,然后在具体的交换文件中用这些对象的实例进行替换。这样,该语法的定义就可以和实际的元模型进行分离。例如,元元模型定义了元实体,它的某个属性称为“Name”,那么语法就可以定义一个标识符(Meta-entity Name),该标识符的定义就代表了它的各种取值,在元模型中包含了带属性定义“Name”的元实体。

由于对要传递的信息语义进行了分离,数据交换模型的体系结构支持多重语法的概念。同样,在任何语法中必须提供多重编码。语法是一个规则集,表示了给定模型实例的终结符产生式。编码是终结符怎样表示为实际字符或字节的定义,因此为了达到不同的目标就必须进行不同的编码。例如,“正文编码”就是指信息没有压缩成缩写形式,而是以人们可阅读的方式出现的编码方式。“字符编码”就是将所有关键字以号码的方式表示,以减少交换文件的大小,但所有信息均以 ASCII 字符方式保存,这些字符可以在通讯线路上传输。“二进制编码”是将信息压缩为一种更紧凑的形式,力图在计算机系统内的通讯进程间进行交换。

为了达到这个目标,语法在不同编码层次上就要采取相应的文法定义形式,以表示语法的有关细节部分。例如,就文法的 BNF 描述来讲,语法将某些标识符作为语法定义中的终结符,但这些标识符会在编码中进一步分解,以提供对象的特定编码表示。

三、CASE 数据交换的元元模型

元元模型的目的是为那些用可扩充性来定义要交换的 CASE 数据信息语义的人员提供一个工具,但并不希望成为一个通用的建模表示。

1. 元元模型

我们这里所使用的元元模型是由实体关系属性(ERA)建模语言组成的,此外还包括元元模型对象的一个相关图形表示。元元模型要定义的术语有:

(1)实体:是可以包含属性的实体。必须有唯一的标识符和在实体内声明的属性。这一概念提供了一个预定义的数据类型集。

(2)关系:支持二元的多对多关系。这些关系可以带有若干属性,并支持可选择性和明确的度的概念,用来表示源和目标实体的多重性。

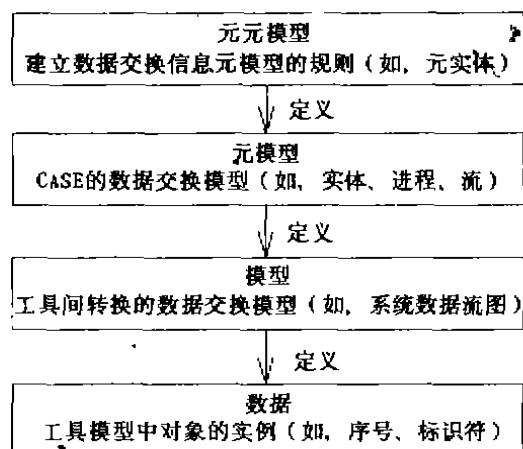


图1 数据交换信息模型的四层框架

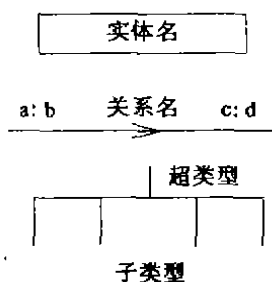


图2 CASE数据交换信息模型的表示图元

(3)子类型:在元元模型里要支持实体的子类型,还要允许进行多重继承。子类型继承超类型的所有属性和关系,超类型既可以是具体的也可以是抽象的,具体的超类型可以在信息交换中实例化,但抽象的超类型却不能实例化,仅是其子类型可以进行实例化。

由于数据交换模型力图给出的是一个交换信息的定义而不是动态环境,因而在继承的属性名上发生冲突时的处理规则都比较简单。在许多情况下,一个对象有多个超类型,但它们都共享一个共同的根;这样,子类型将“继承”每个超类型的相同属性集,它们在子类型中仅出现一次。两个同名属性的其它继承形式均定义为非法,因而数据交换模型的用户不能定义这样的属性。

(4)复杂关系:由于仅直接支持二元关系,因此就必须提供一个机制来记录建模的底层语义,用以表示更复杂的关系,可以对实体进行分类,以表示 n 维关系和相关对象。这样,概念工具就可以捕获相关的信息。

(5)图形表示:在这里所使用的表示方法建立在现有的表示基础之上,各组成部分如图2所示。关系上的箭头定义了读取名字的方式以及源和目标实体,但它没有更深的语义含义。元元模型的图形表示如图3所示。

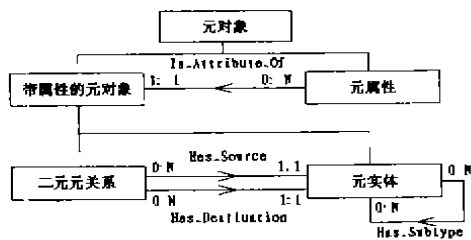


图3 CASE数据交换信息的元元模型

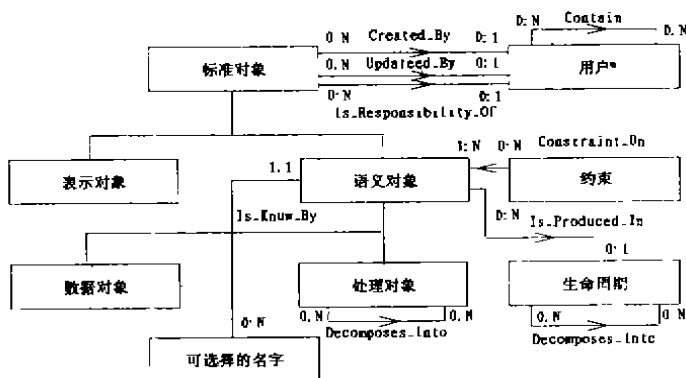


图4 核心主题域

2. 可扩充性

可扩充性就是在现存对象上定义新的实体、关系和属性,但不能去掉预定义的实体、关系和属性,也不能进行重新命名。

预定义在CASE数据交换元模型中的单个对象称为根对象,它的唯一属性就是数据交换模型标识符,这是数据交换中对象的唯一标识符。这个根对象是所有其它实体的根,即其它实体必须是该根某一层的子类型。这既可以用到那些定义在数据交换模型中的标准实体上,也可以用到使用可扩充性增加的所有实体上,但不能使用扩充性为根对象定义超类型。

四、CASE 数据交换的元模型

为了讨论方便,可以将元模型分成两个部分:语义模型和表示模型。语义模型覆盖了信息的底层定义,而表示模型则给用户提供了信息表示方式。

CASE 数据交换元模型的目标是定义一个对象、属性以及关系的集合,用以覆盖在CASE工具间所需的数据交换信息,将精确的含义传递给所有的对象。语法定义本身不能在工具间进行通讯,因此而

元模型的目标就是定义使用该语法的语言含义。

1. 语义元模型

在语义元模型中,使用主题域这个概念将语义模型分成相关的几个领域以便理解和解释。本文定义了四个主题域:核心主题域、数据流建模主题域、实体关系属性(ERA)建模主题域和数据清单主题域。

核心主题域描述了在CASE数据交换模型中交换对象所需的有关信息,这里定义的有关内容包括创建或修改对象的用户细节、对象上的约束以及相关的通用信息。这里要提供一个工具来定义用户的层次结构和生命周期。它包括两类对象:语义对象和表示对象。所有语义模型和表示模型中的对象分别是它们的子类型。核心主题域如图4所示。

数据流建模主题域覆盖了由数据流图技术所表示的语义。这种表示的底层语义并没有太大差异。主要的差别是在不同对象类型上所允许的分解程度。一些方法仅允许对处理进行分解,而另一些则允许对存储、外部系统以及流进行分解。在一些方法中,一些语义信息包含了提供给用户的信息表示图符,这不仅是一个图形交换的问题,而且该信息还必须在语义模型中进行表示。数据流建模主题域如图5所示。

实体关系属性(ERA)建模主题域覆盖了信息系统开发中所遇到的ERA建模的主要形式。这里还没有覆盖面向对象的方法,这是一个要进行扩充的领域。它覆盖的各种形式从仅支持二元非属性关系到n维属性关系,甚至还包括带属性的角色。这个主题域定义了各种技术在CASE数据交换元模型中的表示方式,以便使用其它形式的工具能够对它进行理解。ERA建模主题域如图6所示。

数据清单主题域覆盖了对其它主题域数据对象的属性定义以及底层数据类型的定义(包括复杂的结构和域定义)。

2. 表示元模型

表示模型定义了工具给用户信息的方式。表示和语义的分离是CASE数据交换模型所采取的基本方法,这样就使底层语义能以其单纯的方式进行建模,而不需要考虑对象、关系及度的表示方式。表示模型使数据信息产生工具能一次定义和交换信息表示的一种类型,如数据流图,而且也能传递表示的实际例子,在实际例子中可能有几个不同的数据流图都使用定义的表示方式。这种方法使得每个实

例所需的信息量大为减少,而且也可以配置工具来捕获交换数据所需的信息。我们在这里所讨论的表示元模型可使用图符链图(ICON_LINK)这种表示形式,元模型提供了各种图类型的有关定义信息,包括图上允许的图符和链。图符可以有各种复杂的结构,由多种组成部分组合而成。在表示模型中还可以定义有关底层语义元模型的信息显示规则,以附加到所定义的图符和链上。如,在一个定义的图符内显示一个实体名以表示一个实体。这些规则可以包含图符所表示的对象内包含属性的大多数情况。

表示模型也可以用来定义基于底层属性值的图

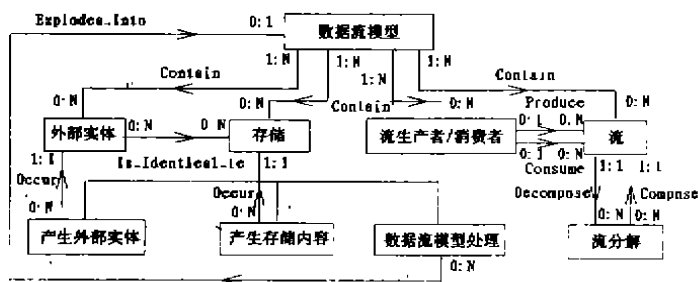


图5 数据流模型主题域

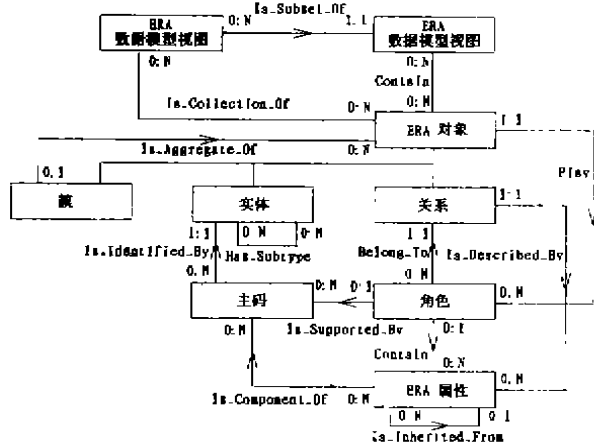


图6 实体关系属性模型主题域

形表示形式,这包括图符链图的色彩、大小、字符方式等等,例如,通过映射定义对象的对象集来定义,表示形式就可以用图符链图来表示。定义对象包含了图符链图的实际协调信息,以及对由图符或链所表示的底层语义实例的调用关系信息。在这里还可以增加其它信息,如规则系统没有定义的细节。

在这里要进行的一个扩充是增加从底层语义信息获得的属性值导出规则,力图不论约束定义采用何种语言,都能用于属性导出规则的定义。

五、CASE 数据交换模型

CASE 数据交换模型的体系结构提供了多重语法,每个均可以进行多重编码。由于定义了元元模型,因而可以将元模型与数据交换模型的定义分离开来,并可以由元元模型来定义语法和编码规则,语法定义了语言的文法,并将标识符定义为某种表示形式,如 ASCII 表示成二进制形式。

所有 CASE 数据交换模型的语法和编码必须提供一个标准的“信封”,将内容定义为数据交换信息,并定义交换中所使用的语法和编码。这个数据交换信息的主要内容三部分组成:头部、元模型定义以及模型定义。

在头部定义了信息生产工具的标识,字符集以及在信息交换中使用的文本格式和产生的日期、时间细节。在模型中对元模型的调用需要给出元模型的版本号,以便信息消费工具在接受信息时能检查出在扩充中预定义内容的非法重用。尽管预定义的根对象不是在元模型中定义的,而是在模型框架中定义的,但这种情况仍有可能发生。如果只使用了标准的元模型(即系统提供的元模型),那么就不需扩充了。模型定义了两个工具间要交换数据信息的实际例子。元元模型定义了元模型层所具有的对象、对象具有的属性、以及属性的数据类型,同时还定义了元模型所允许的扩充形式。

在给出了以上这些定义之后,就可以定义一个语法,该语法要表示所有数据交换所需的信息。这些信息在元模型层上用于进行扩充,在模型层上用于工具间所需的信息交换。元元模型中对象的属性可以用作语法定义中的标识符,这些标识符随后在一个给定的元模型中实例化为一个实际值。元元属性名在元模型定义部分里作为关键字出现,而且要给出相应的定义。这些均建立在模型中元模型所允许的结构基础之上。

参考文献

- [1] M. Imber, CASE Data Interchange Format Standards, Information and Software Technology, Vol 33 No. 9, 1991
- [2] CDIF, CDIF-Framework for Modelling and Extensibility, July 1991, Interim Standard EIA/IS-81 Electronic Industries Association, USA, 1991
- [3] CDIF, CDIF Standardized CASE Interchange Metamodel, July 1991, Same to [2]