ICS 35.040
L 72

中华人民共和国国家标准

GB/T 32392.2—2015

信息技术　互操作性元模型框架(MFI)　第2部分:核心模型

Information technology—Metamodel framework for interoperability(MFI)—Part 2:Core model

2015-12-31 发布　　2017-01-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　　次

前　言

GB/T 32392《信息技术　互操作性元模型框架(MFI)》包含以下几个部分：

——第1部分：参考模型；

——第2部分：核心模型；

——第3部分：本体注册元模型；

——第4部分：模型映射元模型；

——第5部分：模型构件六模型框架；

——第6部分：注册规程。

本部分按照GB/T 1.1—2009给出的规则起草。

本部分为GB/T 32392的第2部分。本部分参考国际标准最终委员会草案ISO/IEC FCD 19763-2：2007版编制。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本部分由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本部分起草单位：武汉大学软件工程国家重点实验室、中国电子技术标准化研究院。

本部分主要起草人：何克清、何扬帆、王翀、王健、王静。

引　言

随着电子业务和电子商务在因特网的广泛传播，跨国家和跨文化的业务贸易和其他相关信息交换已成为IT业内外人员主要关注的问题。他们致力于规范代表业务实践的领域特定业务过程模型和标准建模结构，如每个业务域的数据元素、实体轮廓和值域等。标准化的工作主要关注模型或者元模型的内容，使用UML和XML表示和交换业务的语义。

标准化活动推动了元模型和UML轮廓的发展，这些活动包括联合国贸易促进和电子商务中心(UN/CEFACT)、结构化信息标准促进组织(OASIS)、对象管理组(OMG)和卫生7级(HL7)标准。然而，不同的标准组织都按照自己的方式规定元模型模式。因此，需要为元模型的一致性开发和注册规定公共的基础，否则有可能产生重复和不一致。

用于分类和注册规范化模型元素的统一框架对建立独立开发的元模型的协调性来说是一个重要部分，并且能够促进它们在组织间的广泛重用。

元模型和模型能够描述可以共享的业务概念和部件，以支持系统之间的互操作。

因特网应用的发展使得信息共享变得越来越重要。然而，对于真实的信息来说，不同的开发者、不同的用户和不同的视角都会赋予其不同的解释。仅通过单词分类对信息进行整理是比较困难的。我们将这些本质复杂的信息整理成一个系统化的信息世界，并建议MFI能够通过信息注册来提高信息的共享和兼容性。本部分为解决这个领域的问题提供了帮助。

信息技术　互操作性元模型框架(MFI)
第2部分:核心模型

1　范围

GB/T 32392的本部分提供了核心模型。该核心模型将在GB/T 32392的其他部分得到应用。核心模型规定了为注册其他元模型和模型的规范化MFI元模型。

特定业务域中注册的业务概念和业务模型的标准化不在本部分的范围之内。

2　规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 18391.6—2009　信息技术　元数据注册系统(MDR)　第6部分:注册(ISO/IEC 11179-6:2005, IDT)

GB/T 32392.3—2015　信息技术　互操作性元模型框架(MFI)　第3部分:本体注册元模型(ISO/IEC 19763-3:2007, IDT)

GB/T 32392.4—2015　信息技术　互操作性元模型框架(MFI)　第4部分:模型映射元模型

ISO/IEC 19501:2005　信息技术　开放分布式处理　统一建模语言

ISO/IEC 19502:2005　信息技术　元对象设施

ISO/IEC 20944　信息技术　元数据注册系统互操作性和绑定

3　术语、定义和缩略语

下列术语、定义和缩略语适用于本文件。

注:3.1定义了本部分使用的统一建模语言UML(ISO/IEC 19501:2005)和元对象设施MOF(ISO/IEC 19502:2005)中的术语。3.2列出了本部分使用但未包含在3.1中的术语及其定义。

3.1　本部分使用的UML和MOF术语

3.1.1

抽象　abstraction

使实体区别于其他实体的本质特性。

注:抽象定义了观察者视角的边界。

3.1.2

抽象句法　abstract syntax

UML类图中用以定义构件及其相互关系的元类。

3.1.3

制品　artefact

软件开发过程中产生的一种物理形式的信息。

注：写制品可以用于实现可部署的部件。

示例：

文件、脚本和二进制可执行文件。

3.1.4

关联　association

两个或多个类元素之间的语义关系，规定了类元素实例之间的连接。

3.1.5

关联端点　association end

关联的端点，连接关联与类元素。

3.1.6

属性　attribute

MOF 建模中类元素的特征，描述了类元素实例的取值范围。

3.1.7

属性　attribute

对象或实体的元模型特性。

3.1.8

绑定　binding

通过为模板参数赋值的方式来创建基于模板的模型元素。

3.1.9

势　cardinality

集合中元素的个数。

对比：多重度(3.1.44)。

3.1.10

类　class

对一个对象集合的描述。这些对象共享相同的属性、操作、方法、关系和语义。

类可以使用接口集来说明它提供给环境的操作集合。

3.1.11

类元素　classifier

行为和结构特征的描述机制。类元素包括接口、类、数据类型和部件。

3.1.12

分类　classification

将一个对象赋给一个特定的类元素。

3.1.13

类图　class diagram

说明声明性(静态的)模型元素(如类和类型)及其相互关系的图。

3.1.14

协作图　collaboration diagram

描述围绕一个模型结构所展开的交互的图。图中使用类元素和关联，或实例和链接。

3.1.15

部件　component

系统的模块化、可配置的、可替换的部分。部件封装了实现部分，对外提供一组接口。

注：部件通常由一个或多个驻留在其中的类元素（如实现类）进行规约，可以通过一个或多个制品（如二进制可执行文件或脚本文件）来实现。

3.1.16

组合　composition

一种特定形式的聚合。要求部分的实例包含在至多一个组合的实例中。组合对象负责各个部分的创建和销毁。

注：组合可以是递归的。

3.1.17

约束　constraint

语义条件或限制。有些约束是 UML 预先定义的，其他的可以由用户定义。

对比：标签值（3.1.65）和构造型（3.1.60）。

注：约束是 UML 提供的 3 种可扩展机制之一。

3.1.18

容器　container

〈UML 建模数据〉，一个包含其他实例的实例，提供了访问和迭代其内容的操作。

示例：

数组、链表和集合。

3.1.19

容器　container

一个包含其他部件的部件。

3.1.20

语境　context

出于特定目的（如规定对操作）从特定角度观察一组相关的建模元素。

3.1.21

数据类型　datatype

一组值的描述符。这些值没有单独的标识。值的操作不具副作用。

注：数据类型包括基本的、预先定义的类型和用户定义的类型。预先定义的类型包括数字型、串型和时间型。用户定义的类型包括可枚举型。

3.1.22

依赖　dependency

建模元素间的一种关系，如果改变其中一个建模元素（独立元素）将影响另外一个（依赖元素）。

3.1.23

图（示）　diagram

一组建模元素的图形化表示，通常表现为弧（关系）和顶点（其他的某些元素）的连接图。

注：UML 支持下列图示：类图、对象图、用例图、顺序图、协作图、状态图、活动图、部件图和部署图。

3.1.24

（领）域　domain

由特定的知识或活动构成的论域，由一组能被该领域的从业者理解的概念和术语所刻画。

3.1.25

元素　element

模型的原子组成成分。

3.1.26

特征　feature

封装在类元素(如接口、类和数据类型)中的特性,如操作或属性。

3.1.27

框架　framework

〈MOF 建模〉,一个包含模型元素的构造型包。这些模型元素为系统的全部或部分定义了一个可重用的架构。

注:典型的框架包括类、模式或模板。在针对特定应用域进行特化后,框架通常被称为应用框架。

对比:模式(3.1.52)。

3.1.28

可泛化元素　generalizable element

泛化关系中的模型元素。

3.1.29

泛化　generalization

一般元素和具体元素之间的分类学关系。具体元素与一般元素保持一致,并包含额外的信息。具体元素的实例可以在一般元素适用的任何地方使用。

对比:继承(3.1.30)。

3.1.30

继承　inheritance

为更具体元素赋予更一般元素所具有的结构和行为的机制。

对比:泛化(3.1.29)。

3.1.31

实例　instance

一种特殊的实体,具有唯一标识符、一组可以应用其上的操作以及用以存放操作效果的状态。

对比:对象(3.1.47)。

3.1.32

层　layer

相同抽象层上的类元素或包的一种组织形式。层代表体系结构的水平切面,而划分代表体系结构的垂直切面。

对比:划分(3.1.51)。

3.1.33

链接　link

对象间的语义关联,是关联的实例。

对比:关联(3.1.4)。

3.1.34

链接端点　link end

关联端点的实例。

对比:关联端点(3.1.5)。

3.1.35

元类　metaclass

实例仍为类的类。元类通常用以构成元模型。

3.1.36

元元模型　meta-metamodel

定义元模型语言的模型。

注：元元模型与元模型的关系类似于元模型和模型的关系。

3.1.37

元模型　metamodel

〈MOF 建模〉定义模型语言的模型。

3.1.38

元模型　metamodel

〈MDR〉定义其他数据模型的数据模型。

3.1.39

元对象　metaobject

元建模语言中所有元实体的统称。

示例：

元类型、元类、元属性和元关联。

3.1.40

模型　model

针对特定目的对物理系统建立的一种抽象。

注：在描述元元模型的 MOF 规范中，为便于描述，元元模型经常被简单记做模型。

3.1.41

模型元素　model element

从待建模系统中抽象出的元素。

对比：视图元素(3.1.68)。

注：在 MOF 规范中，模型元素被看作元对象。

3.1.42

元对象设施　Meta Object Facility；MOF

用以描述元模型的公共设施。

3.1.43

遵循 MOF 的　MOF compliant

基于 MOF 的 MOF based

基于 MOF 模型描述的。

3.1.44

多重度　multiplicity

规定了集合可以容许的势的范围。

注 1：多重度可用于描述关联的角色、组合的部分等。

注 2：多重度实质上是非负整数的(有可能是无限的)子集。

3.1.45

(模型元素的)名称　name(of model element)

MOF 建模中用以识别模型元素的字符串。

3.1.46

名称空间　namespace

MOF 建模中模型的一个组成部分，用以定义名称。

注：在名称空间中，每个名称有唯一的含义。

对比：(模型元素的)名称(3.1.45)。

3.1.47

对象　object

MOF 建模中封装了状态和行为的实体，具有定义良好的边界和特性。

注 1：状态由属性和关系表示，行为由操作、方法和状态机表示。

注 2：对象是类的实例。

对比：类(3.1.10)和实例(3.1.31)。

3.1.48

操作　operation

对象可以提供的服务，该服务能够产生一定的影响。

注：操作有相应的接口，用以限制可能的实际参数。

3.1.49

包　package

对元素进行组织的常用机制。

注：包可以嵌套在其他包中。

3.1.50

参数化元素　parameterized element

模板　template

含有一个或多个自由参数的类描述符。

3.1.51

划分　partition

处于某层次化体系结构中相同抽象层次或者跨层次的一组相关类元素或者包。

注：划分代表体系结构的垂直切面，而层则代表水平切面。

对比：层(3.1.32)。

3.1.52

模式　pattern

模板化的协作。

3.1.53

轮廓　profile

需要指定的所依赖的模型库以及所扩展的元模型子集。

注：构造型化的包由使用构造型、标签值和约束等扩展机制得到的模型元素组成。这些元素服务于特定的域或者目的。

3.1.54

性质　property

〈MOF 建模〉，表达元素特性名称的值，性质有语义影响。

注：部分性质由 UML 预先定义；其他的可由用户定义。

3.1.55

参考/指针　reference /pointer

〈UML 建模〉，一种模型元素的指称方式。

3.1.56

参考/指针　reference /pointer

〈MOF 建模〉，类元素中名称的槽，用以导航到其他类元素。

3.1.57

关系　relationship

模型元素间的语义连接。

示例：

关联和泛化。

3.1.58

存储库　repository

存储对象模型、接口和具体实现的机制。

3.1.59

角色　role

特定语境中实体的命名行为。

注：角色可以是静态的(如关联端点)或者动态的(如合作角色)。

3.1.60

构造型　stereotype

扩充了元模型语义的新型建模元素。

注 1：构造型必须基于元模型中已有的类型或类。

注 2：构造型可以扩展已有类型和类的语义而不是其结构。

注 3：一些构造型预先在 UML 中给出了定义，其他的则由用户定义。

注 4：构造型是 UML 提供的 3 种扩展机制之一。

对比：约束(3.1.17)和标签值(3.1.65)。

3.1.61

子类　subclass

泛化关系中超类的具体化。

对比：泛化(3.1.29)和超类(3.1.63)。

3.1.62

子类型　subtype

在泛化关系中超类型的具体化。

对比 1：泛化(3.1.29)。

对比 2：超类型(3.1.64)。

3.1.63

超类　superclass

在泛化关系中子类的泛化。

对比 1：泛化(3.1.29)。

对比 2：子类(3.1.61)。

3.1.64

超类型　supertype

在泛化关系中子类型的泛化。

对比 1：泛化(3.1.29)。

对比 2：子类(3.1.61)。

3.1.65

标签值　tagged value

将性质显式地定义为名称-取值对。在标签值中，名称被称为标签。

注：某些标签在 UML 中预先给出了定义，其他的则由用户定义。标签值是 UML 提供的 3 种扩展机制之一。

对比：约束(3.1.17)和构造型(3.1.60)。

3.1.66

类型　type

类的一种构造型，用以指定实例域(或对象域)以及实例(对象)上可以实施的操作。

注：类型可以不包括任何方法。

对比：类(3.1.10)和实例(3.1.31)。

3.1.67

视图　view

模型的投影，从给定的视图或有利位置观察所得，忽略了与该视图不相关的实体。

3.1.68

视图元素　view element

一组模型元素的文本化或图形化的投影。

3.1.69

XML 元数据交换　XML metadata interchange；XMI

由遵循 MOF 的模型生成的 XML 模式。

3.2　本部分使用的通用术语

3.2.1

管理项　administered item

管理记录中管理信息的注册系统项。

3.2.2

特性　characteristic

对象或对象集的性质的抽象。

注：特性用以描述概念。

3.2.3

概念　concept

由唯一的特性组合创建或规定的知识单元。

3.2.4

概念数据模型　conceptual data model

表示现实世界抽象视图的数据模型。

3.2.5

数据　data

信息的可重解释的表示，通常采用形式化方式进行描述，以便于交流、解释或处理。

注：数据可以由人工或自动的方式进行处理。

3.2.6

数据模型　data model

数据的图形化或词汇方式的表示，规定了它们的性质、结构和相互之间的关系。

3.2.7

定义　definition

通过描述性的语句来表达概念，以便将其与相关概念区分开来。

3.2.8

指定　designation

通过指代其记号得到的概念表示。

3.2.9

实体　entity

任何存在的(包括确实存在和有可能存在)具体的或抽象的东西,包括实体之间的关联。

示例:

人、对象、事件、思想、过程等。

注: 实体可以存在于可获得相关数据的地方,也可以存在于不可获得相关数据的地方。

3.2.10

语言　language

为了交流而建立的记号体系,通常包括词汇表和规则。

3.2.11

强制的　mandatory

总是必须的。

注 1: 描述元数据项属性强制程度的三种状态之一,指出需要该属性的条件。

注 2: 应用到元数据项上时,注册状态应该为"已记录"或者更高。

对比:可选的(3.2.35)。

3.2.12

元数据　metadata

用以定义和描述其他数据的数据。

3.2.13

元数据项　metadata item

元数据对象的实例。

注 1: 在 GB/T 32392 中,本术语仅适用于本部分第 4 章所描述的元数据对象。

示例:

模型概念、模型域概要和模型部件集的实例。

注 2: 元数据项有相关的属性,适用于它所实例化的元数据对象。

3.2.14

元数据对象　metadata object

元模型定义的对象类型。

注: 在 GB/T 32392 中,该术语仅适用于本部分中定义的元模型的元数据对象。

示例:

模型概念、模型域轮廓和模型部件集等。

3.2.15

元数据注册库　metadata register

由元数据注册系统维护的信息存储库或数据库。

3.2.16

元数据注册系统　metadata registry;MDR

注册元数据的信息系统。

注: 相关的信息存储库或数据库称为元数据注册库。

3.2.17

模型记号　model sign

指派名称空间中某个名称的元类。

3.2.18

模型概念　model concept

说明某个概念含义的元类。

3.2.19

模型类元素　model classifier

分类模型概念的元类。

3.2.20

模型域轮廓　model domain profile

规定模型概念定义的语境的元类。

3.2.21

模型部件　model component

规定注册元素的元类。

3.2.22

模型部件集　model component set

规定注册元素集合的元类。

3.2.23

模型选择　model selection

用以说明选中的注册元素集合的元类。

3.2.24

模型规范　model specification

规定域规范的文档的元类。

3.2.25

模型构件　model construct

规定建模构件的元类。

3.2.26

遵循 MOF 的模型　model by MOF

规定遵循 MOF 规范的模型或元模型的元类。

3.2.27

模型关联　model association

规定模型类元素之间关联的元类。

3.2.28

模型关联端点　model association end

规定模型关联的端点的元类。

3.2.29

建模参考　modeling reference

规定模型关联的相关信息的元类。

3.2.30

建模构件　modeling construct

建模的记法单元,如 UML 中的命名元素。

3.2.31

互操作性元模型框架　metamodel framework for interoperability;MFI

注册基于元模型和模型的制品的框架。

3.2.32

对象名称　name of object

用语言表达式来指代一个对象。

注：见管理项名称（3.2.33）。

3.2.33

管理项名称　name of administered item

在特定语境中指派给管理项的名称。

注 1：对应的元模型构件是“指定的属性”

注 2：(模型元素的)名称(3.1.45)。

3.2.34

对象　object

在元模型中，表示可以感知或想象的任何事物。

注：对象可以是物质的(如一个发动机、一片纸或一个钻石)、非物质的(如变换比率或项目方案)或想象的(如独角兽)。

3.2.35

可选的　optional

允许但非必须的。

注 1：元数据项属性强制程度的三种状态之一，指出需要该属性的条件。

注 2：应用到元数据项上时，注册状态应该为“已记录”或更高。

对比：强制的(3.2.11)。

3.2.36

注册机构　registration authority

负责维护注册库的组织。

3.2.37

注册系统项　registry item

元数据注册系统中记录的元数据项。

3.2.38

注册系统元模型　registry metamodel

规定元数据注册系统的元模型。

3.2.39

记号　sign

用特定语言或者标记来指示概念。

3.2.40

统一资源标识符　uniform resource identifier；URI

用以标识资源(特别是因特网上的)的格式化串。

注：其句法需遵循《因特网资源定位符的功能建议》[RFC 1736]和《唯一资源名称的功能要求》[RFC 1737]中的建议。

3.2.41

XML 模式　XML schema

可扩展置标语言 XML 的模式定义语言。

3.3　缩略语

URI	统一资源标识符 (Uniform Resource Identifier)
UML	统一建模语言 (Unified Modeling Language)
MOF	元对象设施 (Meta Object Facility)
MFI	互操作性元模型框架(Metamodel framework for interoperability)
MDR	元数据注册系统 (Metadata Registry)
UN/CEFACT	联合国贸易便捷化与电子商务标准委员会 (United Nations Centre for Trade Fa-

	cilitation and electronic Business)
OASIS	结构化信息标准促进组织（Organization for the Advancement of Structured Information Standards）
OMG	对象管理组（Object Management Group）
HL7	卫生7级（Health Level Seven）
XML	可扩展置标语言(eXtensible Markup Language)
RAI	注册机构标识符(Registration Authority Identifier)
DI	数据标识符(Data Identifier)
VI	版本标识符(Version Identifier)
IRDI	国际注册数据标识符(International Registration Data Identifier)

4 互操作性元模型框架(MFI)核心规范

4.1 概述

本章描述了互操作性元模型框架MFI核心所定义的模型。MFI核心提供了一组建模元素以及它们的使用规则，根据这些规则可以注册模型。

4.1.1 核心模型概述

本部分提供了一个基础设施，该设施能够有效促进电子商务中各企业共享合作所需的信息元素和业务模型。本部分有助于实现独立开发的元模型和模型的共享。本部分可以按照MOF的4层体系结构来描述。MOF提供了一组建模元素和相应的使用规则以支持元模型的开发。这一关注点使得MOF能够作为特定域的建模环境，为定义领域元模型提供支持。MOF不是一个具有通用目的的建模环境。

4.1.1.1 MFI的4层元数据体系结构

MFI的4层元数据体系结构基于ISO/IEC 19502:2005中描述的传统的4层元数据体系结构，而ISO/IEC 19502:2005中的框架又以GB/T 16647—1996为基础。

ISO/IEC 19502:2005中描述的4层元数据体系、结构如下：

a) 信息(对象)层(M0)的信息包含了用户希望描述的数据。

b) 模型层(M1)包含了描述实例层数据的元数据。元数据聚在一起形成了模型。

c) 元模型层(M2)包含定义了元数据结构和语义的描述信息(即元元数据)。元元数据聚在一起形成的元模型是一种抽象语言(即没有具体句法或者记法)，可以用来描述不同类型的数据。

d) 元元模型层(M3)包含描述元元数据结构和语义的描述信息。换句话说，这是定义不同类型元数据的“抽象语言”。

在本部分中，术语“元模型”泛指模型和元模型。UML是一种通用的建模语言，独立于特定的对象域或实现环境。可以使用构造型、标签值或约束来定义轮廓以实现对UML语言的扩展。UML轮廓可以在现有的元模型基础上规定，添加新的模型元素。对于使用UML轮廓描述的域而言，需要提供模型元素的构造型信息以清楚地表达模型的含义。

图1示出了域模型可以使用基于MOF的建模构件(如模式或者构造型)来描述。

例如，在电子商务消息交换中，各方交互所遵循的业务过程模型可以看做模型或者元模型。在这种情况下，制品(如消息格式和协议)可以作为注册目标。此外，概念和相应的实例(包括词汇表和分类)也可以使用核心模型进行注册。

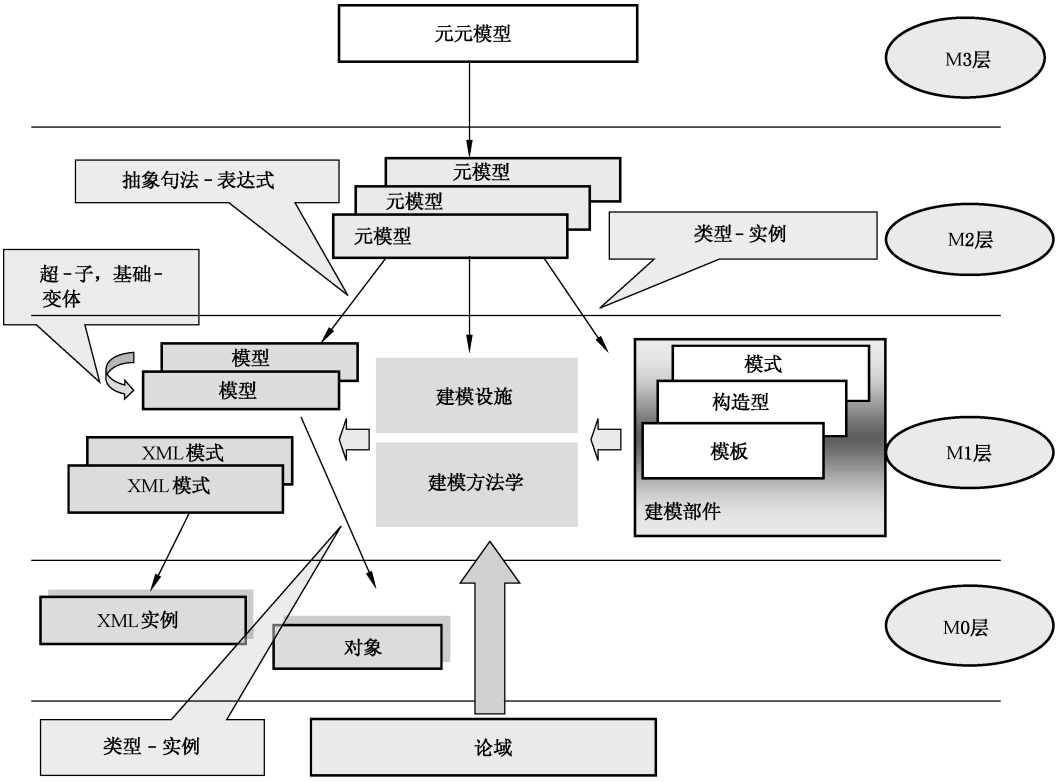

图 1　MFI 的元数据体系结构和待注册的制品

4.1.1.2　互操作性元模型框架一览

本部分使用元模型来描述 MFI 元数据注册库的结构。MFI 注册系统元模型是一个概念化模型，即描述现实世界中相关信息如何组织的模型。本部分不涉及任何实现模型。然而，实现应该严格遵循 MFI 标准，以实现对元模型和相关模型的公共管理。

在 MFI 标准中，使用 MOF 元数据体系结构层来制定框架的目的是为了辅助定义元元模型中元素的含义及其相互之间的关系。本部分对 MOF 体系结构的 M3 层进行了扩充以支持这些元素的注册。

为便于描述，MFI 中的模型被组织为以下 5 个包：

a)　遵循 MOF 的 MDR：管理项（见附录 A）；

b)　遵循 MOF 的对象：包中元素和名称的元素（参见附录 B）；

c)　核心模型（见 4.3、4.4、4.5 和参见附录 C 和附录 D）；

d)　本体注册元模型（见 GB/T 32392.3—2015）；

e)　模型映射元模型（见 GB/T 32392.4—2015）。

如图 2 所示，核心模型包含 5 个包，分别描述目标、注册系统、关系、层对和模型类元素。图 2 中未标上阴影的两个包——遵循 MOF 的对象包和遵循 MOF 的 MDR 包以本部分之外的标准（GB/T 18391 和 ISO/IEC 19502：2005）为基础。此图描述了这些包与 MOF 的 4 层体系结构之间的对应关系。

本部分的主要目的是实现建模制品的共享。如图 2 所示，核心模型位于 MOF 体系结构之中，作为遵循 MOF 的一个元模型。遵循 MOF 的其他元模型也可以置于 MOF 体系结构之中。从本部分的角度看，这些元模型仅仅被称作使用 MOF 和 UML 定义的部件。

为便于描述，核心模型被组织为以下 5 个包：

a)　注册系统包（规范性的；见图 3 和 4.3）；

b)　目标包（规范性的；见图 4 和 4.4）；

c) 关系包(规范性的;见图 5 和 4.5);

d) 模型类元素包(资料性的;参见附录 C);

e) 层对包(资料性的;参见附录 D)。

本部分与 GB/T 32392.1—2015 的关系:

GB/T 32392.1—2015 描述了整个标准的框架,包括本部分、GB/T 32392.3—2015 和 GB/T 32392.4—2015。本部分以 GB/T 32392.1—2015 为基础。

本部分、GB/T 32392.3—2015 和本体定义元模型(ODM)的关系:

GB/T 32392.3—2015 为参考本体和本地本体规定了一个注册其管理信息和其他信息的元模型。本部分对 MOF 进行了扩充。GB/T 32392.3—2015 需要对本部分的一些内容(如模型概念、模型域轮廓、模型部件等)进行扩展。OMG 发布的 ODM 为本体描述语言规定了一个与 MOF 兼容的元模型。ODM 与 GB/T 32392.3—2015 可以很好地结合起来,为本体注册提供支持。

本部分与 GB/T 32392.4—2015 的关系:

GB/T 32392.4—2015 对本部分做了扩充,以支持模型映射。可以使用遵循 MOF 的查询/视图/转换规范(QVT)来描述模型映射规则。QVT 可以与 GB/T 32392.4—2015 很好地协调,共同为模型映射与转换规则的注册提供支持。

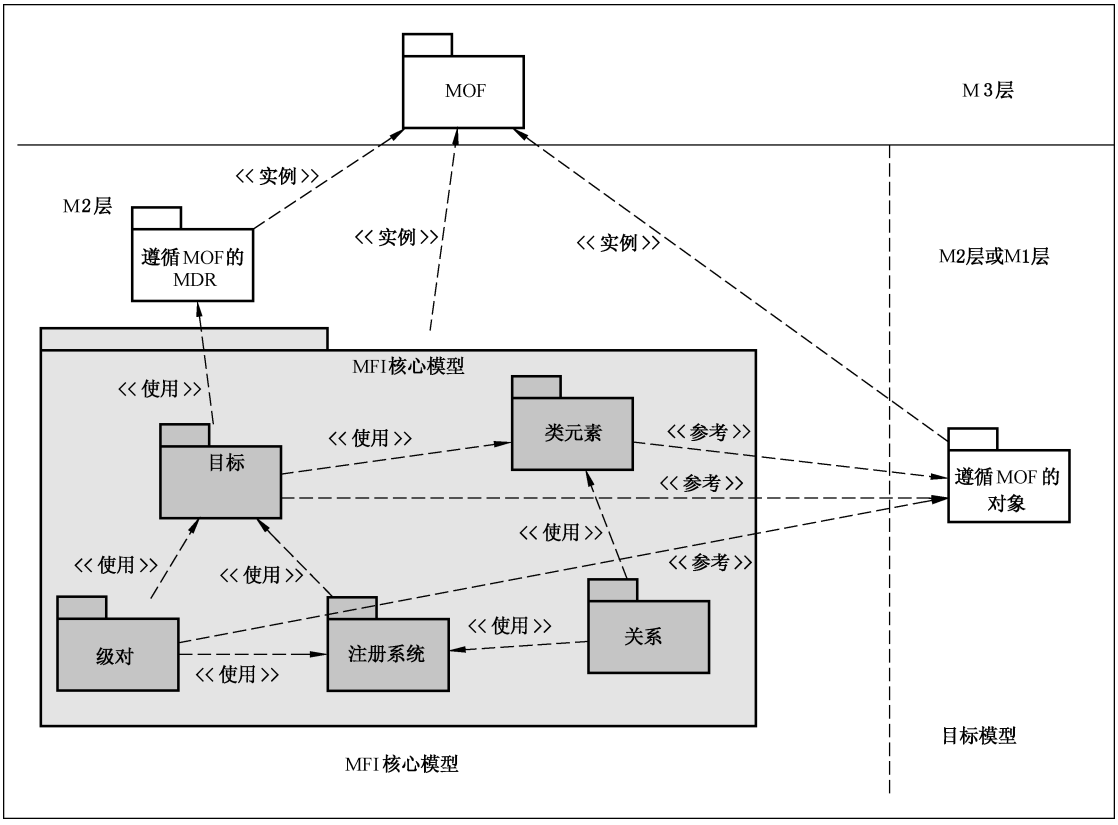

图 2 MFI 核心包和目标模型

4.1.2 核心模型的符号约定

核心模型的制定以 MOF 和 MDR 定义的管理项为基础。使用的 MOF 元模型构件包括类、关系、关联类、属性和参考。这些术语在 3.1 中有相应的定义,其模型在附录 B 有具体的描述。

核心模型表示为一系列 UML 类图,每个类按照如下方式进行描述:

a) 超类

直接继承的类。

b） 属性

属性名称：数据类型和势，强制的或可选的

属性名称的内容及目的的描述。

c） 参考

参考名称：类名和势，强制的或者可选的。

参考名称的内容及目的的描述。

d） 约束

如果需要，可以使用自然语言来表达约束。

模型示出了属性和参考的最小势及最大势。最大势约束在所有时间内有效，而最小势约束仅在元数据项的注册状态为“已记录”或者更高的情况下生效。“已记录”或更高的注册状态在 GB/T 32392 中有详细的定义。“已记录”表明，所有强制属性的值均已记录。

4.1.3 核心模型

图 3 示出了核心模型的高层概览。本部分规定了如下类型的管理项。图中的管理项在后文有详细描述。下述管理项的记法示例参见附录 E：

——模型记号（见 4.2.1）；

——模型概念（见 4.2.2）；

——模型部件集（见 4.2.3）；

——模型选择（见 4.2.4）；

——模型域轮廓（见 4.3.1）；

——模型部件（见 4.3.6）。

此处的管理项符合 GB/T 32392 的定义。本部分中元对象的标识符具有唯一性。

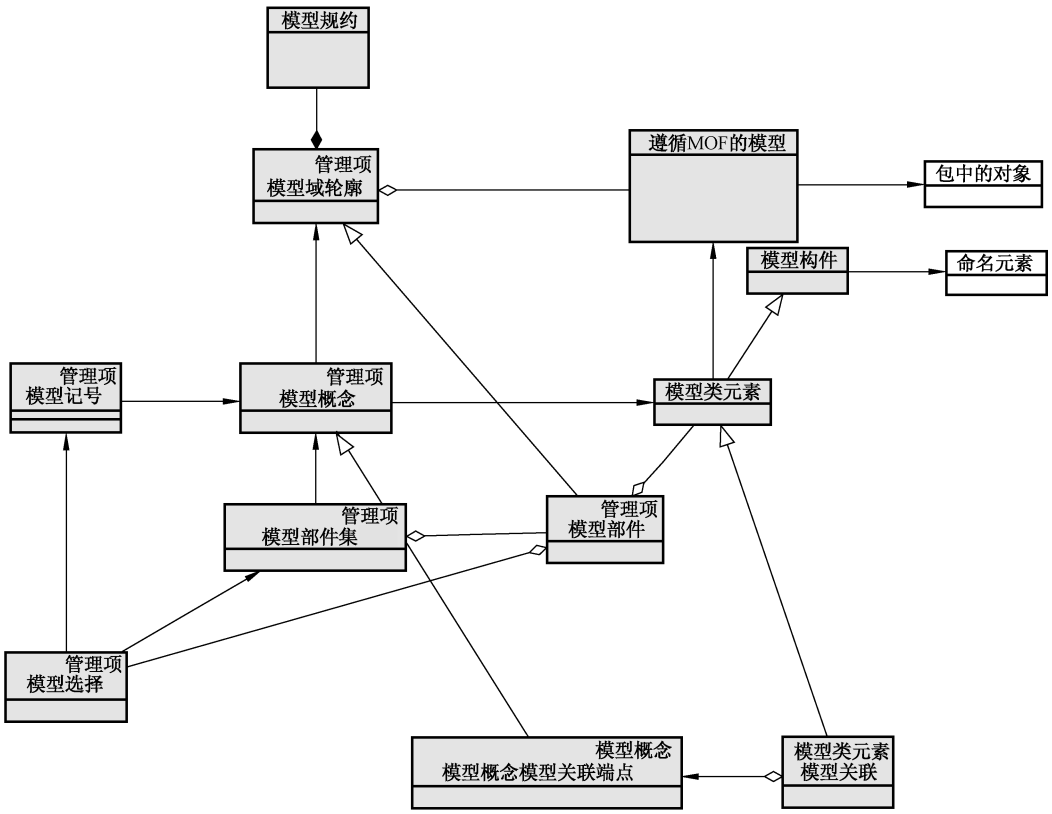

图 3 核心模型概述

4.2 注册系统包(注册系统的结构)

图 4 示出了核心模型中注册包的基本结构。

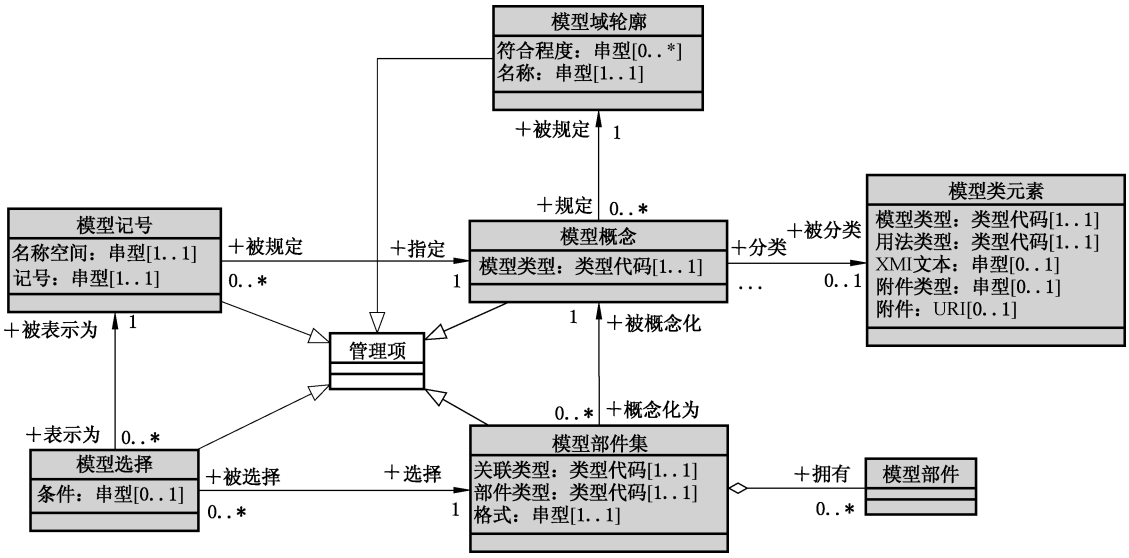

图 4 MFI 核心的注册系统包

4.2.1 模型记号

模型记号是一种元类,具有记号和名称空间。该记号在名称空间中是唯一的。模型记号实例能够辅助识别出特定的、用户感兴趣的事物,如包、类和关联。模型记号由相关的模型概念作出规约,可以用来表达多个模型选择。模型记号是一种管理项。

每个类按照如下方式进行描述:

a) 超类

管理项(源自 MDR)。

b) 属性

名称空间:串型[1..1],强制的

用于识别唯一定义了该记号的名称空间的串。

记号 :串型 [1..1],强制的

用于识别引发模型概念的命名元素的串。

c) 参考

指派:模型概念[1..1],强制的

提供了一个指向模型概念的指针,模型记号的含义由该模型概念规定。

d) 约束

表达 :模型选择[0..*],可选的

模型选择由模型记号表达。

名称空间需遵循该空间所处的特定环境(如 MOF、XML 等)的规则。在注册 XML 元素名称时,需使用 XML 名称空间。在注册 MOF 模型元素时,要使用包的名称。但是,也可以使用全局唯一标识符,如 URI。

名称空间由用户社区中的注册机构负责管理和维护。

模型记号实例包括从其超类(管理项)继承而来的名称,在注册时应该遵循管理项语言部分的规定。

4.2.2 模型概念

模型概念是一个具有模型类型的元类。模型概念由模型域轮廓作出规约，依据模型类元素进行分类。模型概念是一种管理项。

模型概念建立了被模型类元素分类的概念和模型域轮廓提供的语境之间的关联。

每个类按照如下方式进行描述：

a） 超类

管理项（源自 MDR）。

b） 属性

模型类型：类型代码[1..1]，强制的

代码可以用于区分模型类元素的不同类型。表 1 提供了一组预先定义的模型类型。

c） 参考

由...规定：模型域轮廓[0..*]，强制的

提供模型概念语境的模型域轮廓。

概念：模型类元素[0..1]，可选的

对模型概念进行分类的模型类元素。

d） 约束

由…指派：模型记号[0..*]，可选的

概念化：模型部件集[0..*]，可选的

模型类元素及其代码系统应该使用相同的模型域轮廓。

模型概念实例包括从其超类——管理项继承的名称，在注册时都应该遵循管理项语言部分的规定。

模型概念由用户社区中的注册机构负责管理和维护。

表 1 模型类型的代码集（资料性的）

代码	名　称	描　述
STY	构造型（Stereotype）	构造型以元模型的方式进行定义和声明，对已有模型元素的含义做扩充或限制。构造型的实例需声明为特定构造型的对象。参见 C.1
CDV	代码值（CodedValue）	代码值适用于特定的数据类型，这些类型使用编码值来描述模型域轮廓和模型部件集。参见 C.2
PAT	模式（Pattern）	模式是一种模型构件元素，具有可复用性，通常可以在类似的模型中使用。可以当作一种类型（或者模板）。参见 C.3
COM	通信（Communica- tion）	通信提供了一种设施，用于建立基于模式的组合型、嵌套型交互。如果需要，可以将嵌套的、层次化交互中的各个部分标准化。参见 C.4
CMP	部件（Component）	部件提供了一种设施，用于建立基于模式的组合型、嵌套型部件。如果需要，可以将得到的部件中的各个部分标准化。在对部件进行组装时，可以使用模式附带的操作。参见 C.5
FRM	框架（Framework）	框架提供了一种设施，用于建立基于模式的组合型、嵌套型框架。如果需要，可以将得到的框架中的各个部分标准化。在对部件和框架进行组装时，可以使用模式附带的操作。参见 C.6

4.2.3 模型部件集

模型部件集是一种具有概念类型、部件类型和格式信息的元类。模型部件集由一组部件组成，可以由模型概念加以概念化。

使用模型记号指派特定的模型概念,可以指代相应的模型部件集。模型部件集可以按照模型类元素进行分类。概念化类型指定了模型类元素和模型部件的关联。模型部件集是一种管理项。

每个类按照如下方式进行描述:

a) 超类

管理项(源自 MDR)。

b) 属性

概念化类型:类型代码[1..1],强制的

概念化类型定义了被包含的模型部件和标识为“概念化为某个模型”的模型类元素之间有可能存在的关系。表 2 列举了 4 种预先定义的概念化类型。允许对该表进行扩充。

部件类型:类型代码 [1..1],强制的

代码表示了模型部件的类型。

不同的标准开发组织颁布了多种部件类型。部件类型代码集由 MFI 注册系统负责注册与管理。表 G.1 列出了一些可选的部件类型。

格式:串型[1..1],强制的

用以识别模型部件格式的串。允许的格式(如 XML 模式)应该在 MFI 注册系统中进行注册和管理。

c) 参考

被…概念化:模型概念[0..1],强制的

模型概念为模型部件集提供了概念化。

拥有:模型部件[0..*],强制的

模型部件集中包含的模型部件。

d) 约束

被…选择:模型选择[0..*],可选的

模型部件集包含一组由模型类元素得到的模型部件。

部件类型和代码系统应该遵循相应的模型域轮廓。

模型部件集实例包括从其超类——管理项继承的名称,在注册时应该遵循管理项语言部分的规定。

模型部件集由用户社区中的注册机构负责管理和维护。

表 2 概念化类型的代码集

代码	名 字	描 述
T-I	类型-实例 (Type-Instance)	“类型-实例”关系是类型与其实例之间关系的一种概念化类型。应该提供模型包中的类图和对象图
S-S	超-子(Super-Sub)	“超-子”关系是超类和子类之间关系的一种概念化类型。应该提供模型包及其子包
B-V	基础-变体关系 (Base-Variant)	“基础-变体”关系是基础模型和使用可允许的操作对基础模型实施修改后得到的变体模型之间关系的一种概念化类型。见表 F.1。 在概念化类型“基础-变体”中,可以在基础模型的基础上多次实施各种操作。最终,从上层基础模型可以得到一个下层模型。对于每个注册目标而言,指定操作细节的规约应该作为一个模型规约。 “超-子”关系是“基础-变体”关系的一种特例。“超-子”关系仅作用在类上
A-E	抽象句法-表达式 (Abstract Syntax-Expression)	“抽象句法-表达式”关系是上层元模型和下层模型之间关系的一种概念化类型。在这种背景下,模型域轮廓中遵循 MOF 的模型提供了一个元模型。下层模型的描述必须遵从上层模型规定的抽象句法。通常,UML 构造型定义为元模型(如 UML 轮廓)。下层模型应该以这些构造型为基础

4.2.4 模型选择

模型选择是一个具有选择条件的元类。模型选择选定了一个模型部件集，使用模型记号来表达。模型选择代表了模型记号和模型部件集之间的链接。

根据模型部件集的条件，可以访问由模型部件构成的的子集。通过外部参考，模型部件可以包含其它的模型选择作为子部件。模型选择是一种管理项。

每个类按照如下方式进行描述：

a） 超类

管理项(源自 MDR)。

b） 属性

条件：串型[1..1]

用以指定对模型部件集实施过滤的条件的串。

c） 参考

由…表达：模型记号[1..1]，强制的

表示模型选择所选定的模型部件集的模型记号。

选择：模型部件集[0..1]，强制的

模型选择所选定的模型部件集。

d） 约束

模型选择由用户社区中的注册机构负责管理和维护。

模型选择实例包括从其超类——管理项继承的名称，在注册时应该遵循管理项语言部分的规定。

4.3 目标包(注册目标的结构)

图 5 表示的是核心模型的目标包。

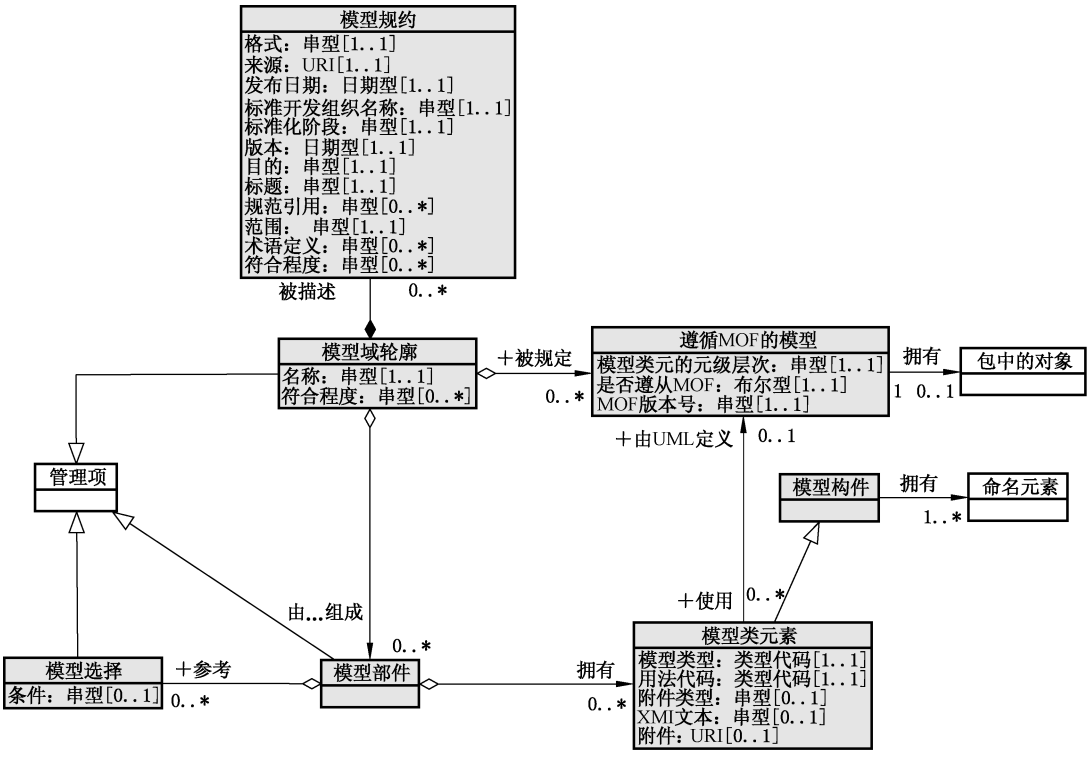

图 5 核心模型中的目标包

4.3.1 模型域轮廓

模型域轮廓是一个具有名称和符合程度说明的元类。符合程度通过符合声明来表达。模型域轮廓由相应的模型规约作出描述，包含多个模型部件，通过遵循 MOF 的相关模型进行规范。模型域轮廓是一种管理项。

每个类按照如下方式进行描述：

a） 超类

管理项(源自 MDR)。

b） 属性

名称：串型[1..1]，强制的

用以识别模型域轮廓中每个实例的串。

符合程度：串型[0..*]，可选的

用以识别模型规约中定义的符合程度的串。

c） 参考

被...描述：模型规约[1..*]，可选的

标准或者轮廓的模型规约。

被...规定：遵循 MOF 的模型[0..*]，可选的

基于 MOF 的模型或者元模型。

包含：模型部件[0..*]，可选的

模型域轮廓中聚集的模型部件。

d） 约束

属性“名称”的取值无须是全局唯一标识符。

4.3.2 模型规约

模型规约是一种元类，它包含一些元数据以描述标准及相关轮廓。这种描述信息便于人阅读和理解。模型规约可包含遵循 MOF 的模型文档和模型部件文档。

每个类按照如下方式进行描述：

a） 超类

无。

b） 属性

格式：串型[1..1]，强制的

表示文档类型(如 ppt、pdf、http 等)的串。

来源：URI[1..1]，强制的

表示材料来源位置的统一资源标识符。

标准开发组织名称：串型[1..1]，可选的

表示标准开发组织的串。

发布日期：日期型[1..1]，可选的

表示发布的日期。

版本：数字型[1..1]，可选的

表示标准或轮廓的版本号。

标准化阶段：串型[1..1]，可选的

表示标准化过程所处的阶段。

标题：串型[1..1]，强制的

表示标准或者轮廓的标题。

目的:串型[1..1],可选的

表示制定文档的目的。

范围:串型[1..1],可选的

表示文档的应用域和范围。

规范引用:串型[0..*],可选的

表示规范化引用的串。

术语定义:串型[0..*],可选的

表示描述术语定义的串。

符合程度:串型[0..*],可选的

表示描述符合程度的串。

4.3.3 模型构件

模型构件是一种用于表示 UML 命名元素集的元类。模型构件可以包含的命名元素包括模式(模板)、构造型(标签值)、代码值和约束等。

每个类按照如下方式进行描述:

a) 超类

无。

b) 参考

拥有:命名元素[0..*],可选的

命名元素(源自 MOF)。

注:模型层的标准建模构件可以在元数据注册机制中进行注册以促进互操作。这些建模构件的共享和管理非常重要。建模构件包括模式、构造型、模板、标签值、词汇表和其他类型的元数据。

4.3.4 模型类元素

在核心模型中,模型类元素是一个具有模型类型、用法类型、XMI 文本、附件类型和附件等多种属性的元类。模型类元素将建模单元和类型化模型标识为概念。模型类元素的定义遵循 UML 和 MOF。

每个类按照如下方式进行描述:

a) 超类

模型构件。

b) 属性

模型类型:类型代码[1..1],可选的

指定模型类元素类型的类型代码(见 4.2.2 的表 1)。

用法类型:类型代码[1..1],可选的

指定使用目的的类型代码。

用法代码集由注册机构负责维护。

XMI 文本:串型[0..1],可选的

指定用以描述 XML 模式或者 XML 格式的串。

附件类型:串型[0..1],可选的

指定附件格式(如 ppt、pdf 等)的串。

附件:URI[0..1],可选的

指定附属材料位置的统一资源标识符。

c) 参考

用 UML 定义的:遵循 MOF 的模型[0..1],可选的

遵循 MOF 的模型是一个使用 UML 语言定义的包。

d) 约束

模型类元素由一个包组成,可以是模型构件的子集。一个模型单元包含一个包。

包也可在建模时作为组织构件。可以使用嵌套、导入和泛化等方式来管理模型的复杂度。

模型类元素和相应的代码系统需遵循特定的模型域轮廓。

4.3.5 遵循 MOF 的模型

遵循 MOF 的模型是一个元类,具有模型层次、是否与 MOF 兼容、MOF 版本等属性。遵循 MOF 的模型可以有一个对应的包中对象。

每个类按照如下方式进行描述:

a) 超类

无。

b) 属性

模型层次:串型[1..1],强制的

指示模型或元模型所在元级层次(如 M2 层、M1 层)的串。

是否遵从 MOF:布尔型[1..1],强制的

表示元模型是否遵从 MOF 的真假值。

MOF 版本号:串型[1..1],可选的

表示 MOF 版本号码的串。

c) 参考

拥有:包对象[0..1],强制的

表示元模型的包对象。建议使用遵从 MOF 的模型。

d) 约束

使用:模型类元素 [0..*],可选的

遵循 MOF 的模型对应于 MOF 元数据体系结构中 M2 层或者 M1 层的元模型。

下层模型应该遵循上层模型的抽象句法和相关约束。

4.3.6 模型部件

在核心模型中,模型部件是一个具有模型类元素和模型选择的元类。模型选择扮演了以插拔方式与外部元素建立关联的角色。模型部件是一种管理项。

注:模型部件可包含一些模型类元素作为模型元素。与此同时,模型部件可以由模型概念进行独立的概念化,这个模型概念由其他的模型类元素负责分类。

每个类按照如下方式进行描述:

a) 超类

管理项(源自 MDR)。

b) 参考

拥有:模型类元素[1..*],可选的

模型部件中使用的模型类元素。

参考:模型选择[0..*],可选的

在模型部件中使用、并参考已注册模型部件集的模型选择。

c) 约束

模型选择的使用需依据该模型选择所指向的模型部件集。

4.4 关系包(注册目标的关系)

图 6 描述了核心模型的关系包。

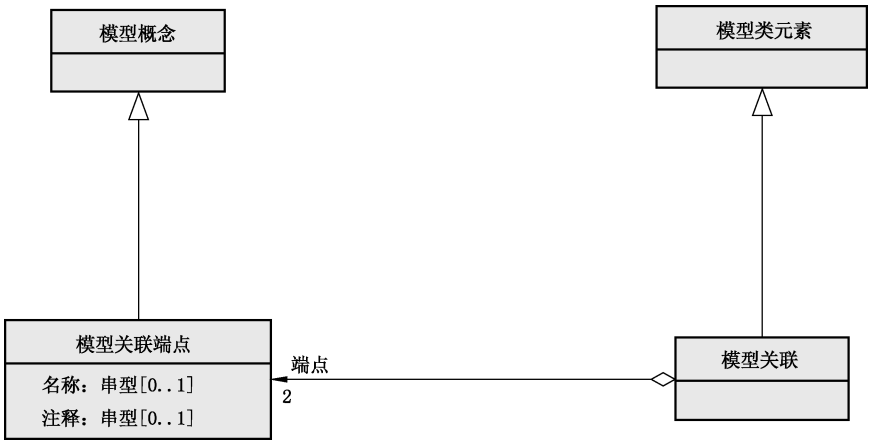

图 6 核心模型的关系包

4.4.1 模型关联

模型关联是一个元类,用以定义模型类元素之间关联。模型关联为模型类元素上的链接集合提供了一种分类方式。作为模型关联的实例,链接表达了模型类元实例之间的联系。

每个类按照如下方式进行描述:

a) 超类

模型类元素。

b) 参考

端点:模型关联端点[2..*],强制的

指定模型类元素间链接端点的角色。

c) 约束

定义模型关联需要规定两个模型关联端点。

如果模型关联是有向的,应该使用能够指示方向的名称。

即使没有直接的模型关联,从元模型得到的间接关联仍有可能存在。

4.4.2 模型关联端点

模型关联端点是一个元类,用以描述模型关联中的两个端点。每个模型关联端点定义了模型概念在模型关联中扮演的角色以及模型概念集上的约束。

每个类按照如下方式进行描述:

a) 超类

无。

b) 属性

名称:串型 [1..1],可选的

用以识别模型关联端点角色的串。

注释:串 [1..1],可选的

模型关联端点的非形式化描述串。

c) 参考

无。

d) 约束

模型关联端点的实例是链接端点。链接端点定义了在模型关联实例中,链接与模型关联端点的模型概念之间的关系。

4.4.3 模型交换的标准格式

本部分没有指定具体的元数据格式。但是,建议在互操作中使用核心模型的 XMI 模式作为元数据交换格式。如果提供了合适的 XML 模式作为轮廓,绑定了诸如 ISO/IEC 20944 的其他元数据格式也是允许的。

5 符合性

5.1 概要

本部分描述了一个概念元模型,而非物理实现。因此,元模型的实现不需要与规定的完全相符。但是,必须保证元模型和实现之间的双向映射是无二义性的。

符合的实现需要:

——满足 4.3 的要求;

——确定符合性程度(5.2);

——确定符合性级别(5.3)。

5.2 符合性的程度

MFI 核心模型的符合性从 3 个方面进行规定:第一个是值的要求;第二个是元数据交换的互操作性;第三个是注册内容(如上层模型和下层模型)之间的符合。

每个类按照如下方式进行描述:

a) MFI 核心模型的符合在表 4 中规定;

b) 交换格式的符合在表 4 中规定;

c) 注册内容的符合在本部分中没有规定。

“级别 1”和“级别 2”实现之间的区别对于同时解决互操作性和扩展性是必要的。本部分描述了增强互操作性的规范。扩展是根据用户、销售商、协会和行业需要而实施的。实现符合程度见表 3。

表 3 实现符合性的程度

扩展	实现	
	严格符合	符合
本部分未直接规定	应该支持所有强制或者可选的属性和参考	支持所有强制和可选的属性和参考
被规定、并与 GB/T 32392 的其他部分保持一致	不得支持使用、测试、访问和探测任何扩展特征和扩展属性	在实现时允许使用、测试、访问和探测任何扩展特征和扩展属性
有可能会作为本部分后续版本的试用	不得承认、使用或者允许产生依赖于任何未规定、未定义或者实现定义的行为的属性	允许识别、使用或产生依赖于实现定义的行为的属性

注：使用元模型或基本属性的扩展有可能会造成未定义的行为。所有严格符合的实现也是符合的实现。

级别 1 的实现在功能上有局限性，但是能够最大程度的实现关于本部分的互操作。级别 2 的符合实现可以具备更多的功能，但其互操作性较差。

5.3 符合性的级别

实现需要遵循本部分定义的符合性两个级别之一。表 4 表示符合性级别的信息。

表 4 符合性级别

符合性视图	符合性级别	
	级别 1	级别 2
数值要求	只支持和使用在第 4 章中规定的元数据元素、关系和属性	可以支持和使用第 4 章中规定的元数据元素、关系和属性
交换格式	只支持和使用 XML 格式	未规定

5.4 承诺

两种承诺状态之一将被应用于元数据项属性，表示请求该属性的条件。实施到元数据项的承诺状态是“已记录”或更高状态。本部分中规定的属性和关系均被声明为强制的或可选的。

为了达到符合性的目的：

a) 强制的属性和关系必须存在，并且这些属性和关系应符合本部分的规定；

b) 可选的属性和关系不是必须存在的，但是如果存在，则应符合本部分的规定。

当且仅当相关元数据项的注册状态是“已记录”或更高状态时，上述承诺将被强制实施。

5.5 实现符合性的声明

本部分的实现应包括一个实现符合性声明，说明：

a) 该实现是否符合或者严格符合(5.2)；

b) 符合性级别 1 或级别 2(5.3)，或二者皆是；

c) 支持或使用哪些扩展。

5.6 注册的作用和责任

符合需要在注册机构的任务和责任中考虑，正如 GB/T 18391.6—2009：数据元素的注册中说明的。系统的扩展符合性需要规程的形式化、各机构的任务和职责达成一致、软件产品的使用指南和其他系统的转换指南。这些方面的形式化应和上述条款中符合性需求一致，也应与 GB/T 18391.6—2009 等注册机构的作用保持一致。

附 录 A
（规范性附录）
遵循 MOF 的 MDR

A.1 概述

元数据注册系统元模型在 GB/T 18391 中有相应的定义。数据建模的理论基础是：所有数据均用以描述自然世界（即论域）中对象的属性，数据的基本单元是数据元素。该元模型使用了许多与数据建模相同的概念数据结构。

元数据注册系统的关键概念如下所示：

——数据元素概念：一个可以用数据元素的形式表示的思想，以独立于任何特定表示的方式来表达。

——概念域：数据元素可能取值的内涵构成的集合。

——值域：可允许的取值构成的集合。

——数据元素：数据单元，其定义、标识、表示和允许值可使用属性集合进行规定。

GB/T 18391 并不期望该注册元模型能够完全符合所有用户的需求。某些部分，比如元模型和模型的注册，要用到其他标准定义的一些元数据属性。如果没有违背 GB/T 18391 中元模型在结构和内容上潜在的规则，那么这种扩充与 GB/T 18391 保持了一致。可以向 GB/T 18391 的概念模型添加新的类、关系和属性。标准的实现者可以在实现中直接使用这些扩充，或者他们可以提供设施让注册系统的使用者定义自己的扩展。

本部分是基于 GB/T 18391 开发的。然而，本部分对注册目标进行了扩充以便能够注册复杂的模型和对象。于是，4 个关键的概念被重新定义为本部分所描述的：模型记号、模型概念（包括模型域轮廓）、模型部件集和模型选择。

至于管理过程和规程，本部分遵从 GB/T 18391 体系的公共设施。目标对象有一个管理项，该管理项由遵循 MOF（如图 A.1 和图 A.2）的元模型进行规约。在本部分中，公共设施按照下列方式应用到所有管理项：

——管理项在注册系统中只能够被标识一次，并且作为单独的项进行管理。

——管理项至少在一个语境中进行命名和定义，允许在多个语境中进行命名和定义。在每个语境中，名称和定义可以用一种或多种语言来规定。

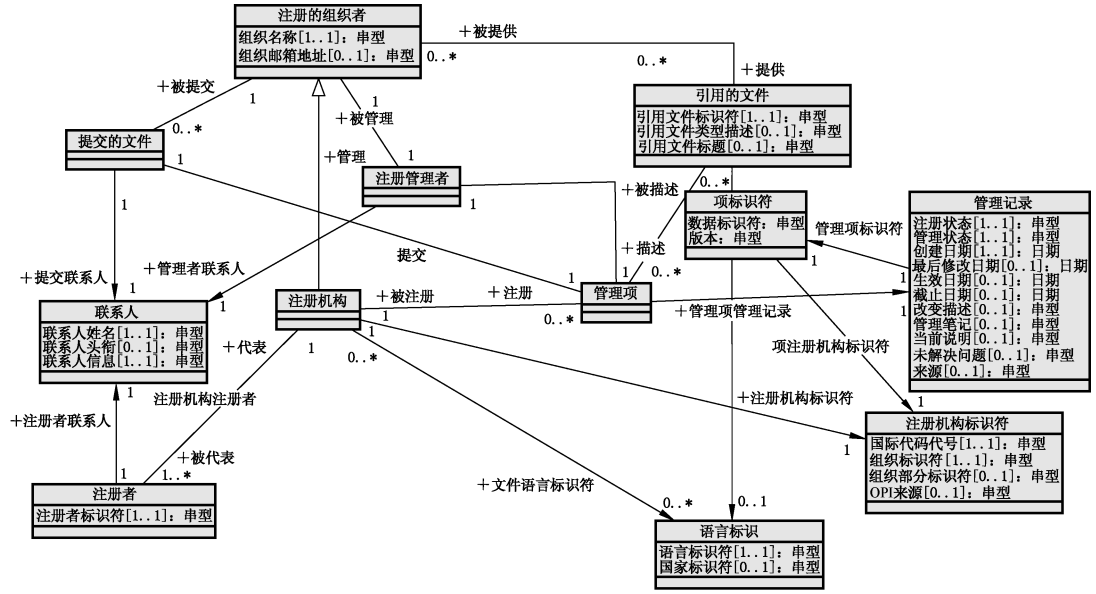

图 A.1　MDR 包(管理和标识)

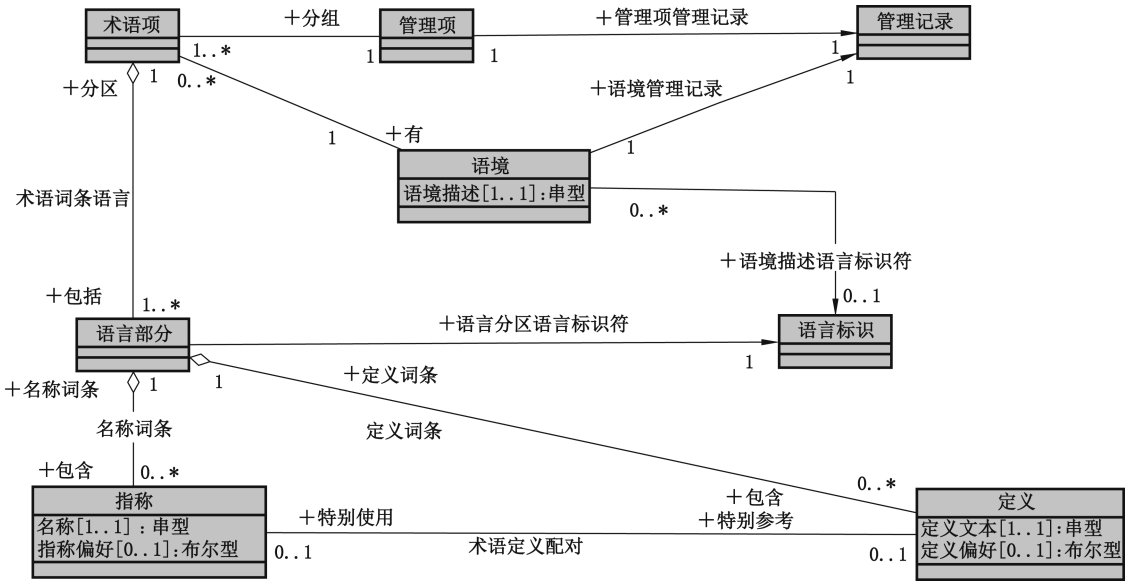

图 A.2　MDR 注册包(语境、命名和定义)

A.2　管理项的命名规则

数据标识符可以被指派给已注册的管理项。在注册机构中,每个管理项应该有一个唯一的数据标识符。注册机构标识符,数据标识符和版本标识符共同组成了管理项的唯一标识符。更详细信息见GB/T 18391.6—2009。

注册机构标识符(RAI)、数据标识符(DI)和版本标识符(VI)组成了国际注册数据标识符。每个管理项都需要有一个国际注册数据标识符。

数据标识符需要由注册机构指定;数据标识符在一个注册机构的辖域内应该是唯一的。由于每个注册机构有可能会确定它自己的数据标识符分配模式,因此无法保证数据标识符本身能够唯一的识别

管理项。为了识别管理项,数据标识符和注册机构标识符都是必需的。管理项的每个名称都需要在特定的语境中来规定。表 A.1 中提供了核心模型中管理项的命名传统。

表 A.1　国际注册数据标识符的命名传统

序号	管理项	国际注册数据标识符				
		数据标识符 1	数据标识符 2	数据标识符 3	后缀 1	后缀 2
1	模型记号	(记号)	(概念)	(域)	(注册机构标识符)	(版本)
2	模型概念	(概念)	(域)		(注册机构标识符)	(版本)
3	模型部件集	(部件集)	(概念)		(注册机构标识符)	(版本)
4	模型选择	(选择)	(记号)	(部件集)	(注册机构标识符)	(版本)
5	模型域轮廓	(域)			(注册机构标识符)	(版本)
6	模型部件	(部件)			(注册机构标识符)	(版本)

A.3　命名的例子

命名的例子如下:

a)　名称(记号/概念/域/注册机构标识符/版本)

“国家/国家和地区/全球疆界/日本国/1.0 版”

b)　模型概念(概念/域/注册机构标识符/版本)

“国家和地区/全球疆界/联合国/2.0 版”

c)　模型部件集(部件集/概念/注册机构标识符/版本)

“ISO/IEC JTC1 参加国成员/国家和地区/日本国/1.0 版”

d)　模型选择(选择/记号/部件集/注册机构标识符/版本)

“情报理学会/国家/ISO/IEC JTC1 参加国成员/日本国/2.0 版”

e)　模型域轮廓(域/注册机构标识符/版本)

“全球疆界/联合国/3.0 版”

f)　模型部件(部件/注册机构标识符/版本)

“加拿大/联合国/2.0 版”

“日本/联合国/2.0 版”

“美国/联合国/2.0 版”

“韩国/联合国/2.0 版”

“英国/联合国/2.0 版”

“澳大利亚/联合国/2.0 版”

“中国/联合国/2.0 版”

附 录 B
（资料性附录）
遵循 MOF 的对象

B.1 概述

MOF 是由 OMG 和 ISO/IEC 共同采用的技术，用来定义元数据。元数据是描述数据或信息的数据，因此它也可以被其他的元数据所描述。在 MOF 术语系统中，描述元数据的元数据被称为元元数据，包含元元数据的模型被称为元模型。

元模型在 MOF 中起着重要的作用。MOF 元模型定义了遵循 MOF 的模型中元数据的抽象句法。因为在典型的系统中有多种元数据，所以 MOF 框架需要支持多种不同的 MOF 元模型。通过为元模型定义公共的抽象句法，MOF 为这些元模型的集成提供了一种途径。这种抽象句法与 MOF 模型相关联，它是元模型的模型，即元元模型。MOF 元数据框架通常被描述为一个 4 层体系结构（M0 层、M1 层、M2 层、M3 层）。基于 MOF 的元模型和 UML 轮廓具有如下特征：

a） 使用指定的 UML 记法子集；

b） 提供定义元模型的公共风格；

c） M2 层的元模型是 M1 层模型的抽象句法；

d） M1 层的模型是元模型的一种表示；

e） UML 轮廓（构造型、标签值等）是元模型的附加约束；

f） 元模型和 UML 轮廓间的映射是进行工具开发的基础；

g） 促进不同工具间交换元模型/模型/数据。

构造型、标签值和约束可以定义为一个轮廓，用以扩展 UML 语言。建议创建相应的元模型来描述构造型的含义。UML 轮廓可以在已有元模型的基础上，通过增加新的模型元素而得到。在使用 UML 轮廓描述的域中，需要附上模型元素的构造型以准确表达模型的含义。

图 B.1 示出了 MOF 模型包的片段。

注：在遵循 MOF 的对象包中，包中元素和命名元素这两个类被定义为外部参考元素。包中元素应该被定义为包的子类，并且不带有附加属性。同样，命名元素应该被定义为名称空间的子类，并且不带有附加属性。包和名称空间在 MOF 中有相应的定义。

B.2 和 B.3 总结了名称空间和包的定义。

B.2 名称空间

名称空间是一个元类，用以识别带有名称空间信息的模型元素。

每个类按照如下方式进行描述：

a） 超类

模型元素

b） 属性

/名称[MOF]：串型[1..1]

元建模者提出的名称，该名称在名称空间的辅助下，可以唯一识别模型元素。

/限定名[MOF]：串型[0..1]

最外层包中为模型元素提供的唯一名称。

/注释[MOF]:串型[0..1]

模型元素的非形式描述。

/容器[MOF]:名称空间[0..1]

用以识别包含模型元素的名称空间。

/提供者[MOF]:提供者[0..*]

模型元素的定义所依赖的模型元素。

/约束[MOF]:受限元素[0..*]

施加于模型元素的约束集。约束将施加到模型元素类及其子类的所有实例。

/标签[MOF]:标签[0..*]

提供一种轻量级的扩充机制,允许将映射、供应者、甚至客户特有的信息添加或关联到元模型。

/被包含的元素[MOF]:模型元素[0..*]

被包含的每个模型元素。如果是最顶层的包,那么该包只能够作为关联的被包含元素。

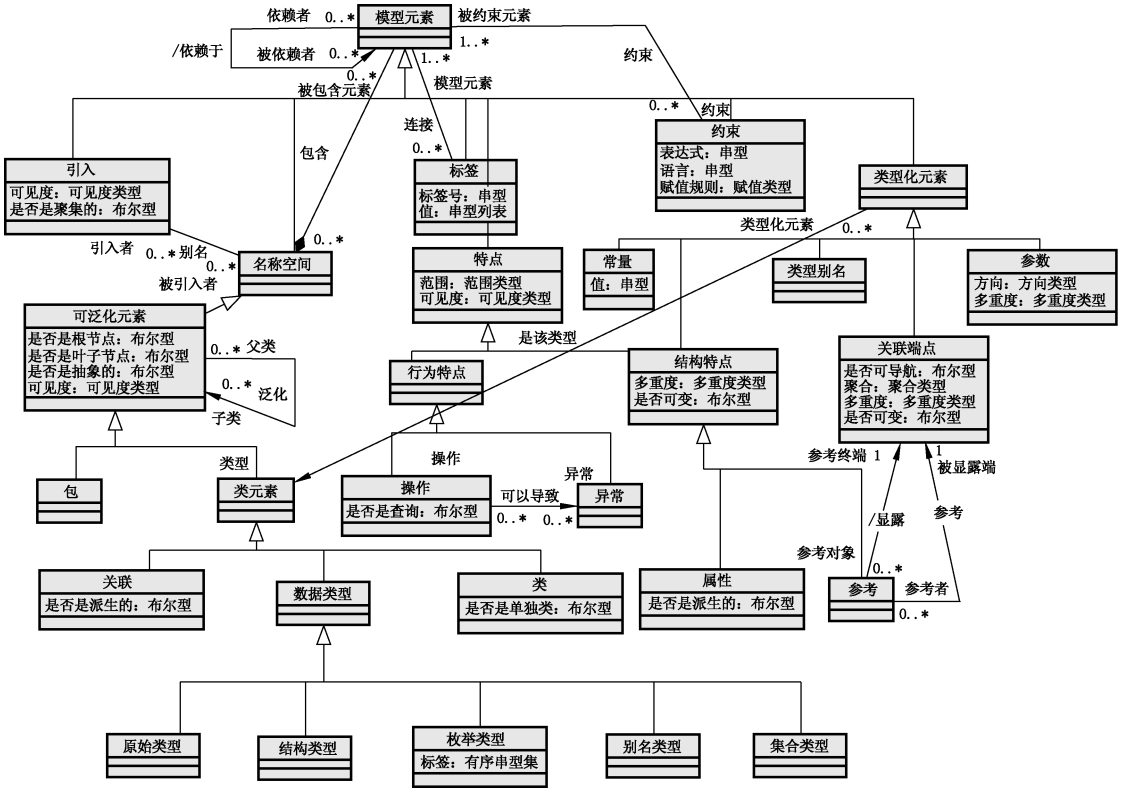

图 B.1　MOF 包

c)　约束

当选择模型元素的名称时,元模型建模者应该考虑将名字翻译为相应标识符时应该遵循的规则。为了减小移植性问题,建议使用以 ASCII 字母开头的名称。名称可以包含 ASCII 字母、数字、空格和下划线。尽量避免使用大写字母、空格和下划线具有重要意义的名称。

限定名是一个串,包含了模型元素的名称、容器、容器的容器等信息,直到得到一个不被包含的元素为止。这个串要以不被包含的元素的名称开头。

由于包含关联是一个组合关联,因此任何模型元素至多含有一个容器,并且包含图是严格树状的。

名称空间定义了一个组合其他模型元素的模型元素。因为名称空间有几个子类,所以名称空间—模型元素组合有可能是一个相当大的集合。

B.3 包

包是一个元类，用以识别包的实例。

每个类按照如下方式进行描述：

a） 超类

泛化后的元素

b） 属性

/名称[MOF]：串型[1..1]

元建模者提出的名称，可用于唯一识别名称空间中的模型元素。

/限定名[MOF]：串型[0..1]

在最外层包中，为模型元素提供的唯一名称。

/注释[MOF]：串型[0..1]

模型元素的非形式化描述。

/容器[MOF]：名称空间[0..1]

用以识别包含模型元素的名称空间。

/提供者[MOF]：提供者[0..*]

模型元素的定义所依赖的模型元素。

/约束[MOF]：约束[0..*]

施加于模型元素的约束集。约束将施加到模型元素类及其子类的所有实例。

/标签[MOF]：标签[0..*]

提供一种轻量级的扩充机制，允许将映射、供应者、甚至客户特有的信息添加或者关联到元模型。

/被包含的元素[MOF]：模型元素[0..*]

被包括的元素是指除了参与到关联的顶层包之外的所有模型元素。

/是否是根节点[MOF]：布尔型[1..1]

用于指出该可泛化元素是否存在超类型。取值为真，表明它没有可能有超类型；取值为假，则表明它有可能有超类型(不管实际有或没有)。

/是否是叶子节点[MOF]：布尔型[1..1]

用于指出该可泛化元素是否是另一个可泛化元素的超类型。取值为真，则表明它没有可能是超类型；取值为假，则表明它有可能是一个超类型(不管实际是或不是)。

/是否是抽象的[MOF]：布尔型[1..1]

用于指出该可泛化元素是否可以有实例。当该属性取值为真时，由可泛化元素表示或分类的实例同时也是该可泛化元素特化后得到的实例。不需要专门的操作来支持可泛化元素的实例创建过程。

/可见度[MOF]：可见度类型[1..1]

在将来，这个属性将用于约束在模型元素容器之外如何使用该元素。目前，MOF 模型元素的可见性规则还没有正式的标准。

/超类型[MOF]：可泛化元素[0..*]

可泛化元素的超类集。注意，可泛化元素没有指向其子类的参考。

附 录 C
（资料性附录）
模型类元素

C.1 概述

模型元素具有不同的粒度。对于 MFI 而言，它们可以定义为模型类元素。例如，模型类元素的子类、方法学、产品、网址、标签值、数据类型、模式（通信、消息、部件和框架）和词汇表（术语和代码值）是可供选择的注册模型类型。模型类元素以 UML 描述的包、类图或者基于 XMI 的 XML 实例来表达。本附录提供了模型类素元的一些子类作为示例。

为了改善对象模型的可共享性和可复用性，必须使用规范化的建模构件，比如构造型和模式，以支持下列需求：

——模型必须由规范化的建模构件组成，不仅包括建模方法还有建模记法。

——预先定义的建模构件应该包括公共原子对象（如日期、货币和国家代码），它们的使用无须讨论。

——表示业务实体的公共聚合对象（如顾客、公司或订单）也应该被预先定义为标准的建模构件。

——业务概念（如交易、发票或结账）通常被表示为对象之间的关系，它们应该被定义为公共基本聚合对象或简单对象的聚合。它们也应该被预先定义为规范化的建模构件。

——通过这种聚合方式（在定义中使用更基础的模式）得到的结果可以定义为对象模式。

——模式可以用来表示业务概念。可以通过聚合更加基础的模式来生成新模式。因此，在模式中必须提供聚合或组合机制。

——指导业务概念的业务规则可以用封装了约束的模式来表示。因此，必须提供模式间的约束继承机制。

图 C.1 给出了模型类元素包的总体视图。图中的部分元类在本附录有具体描述。然而，应该为每个已注册对象定义更加准确的规约。

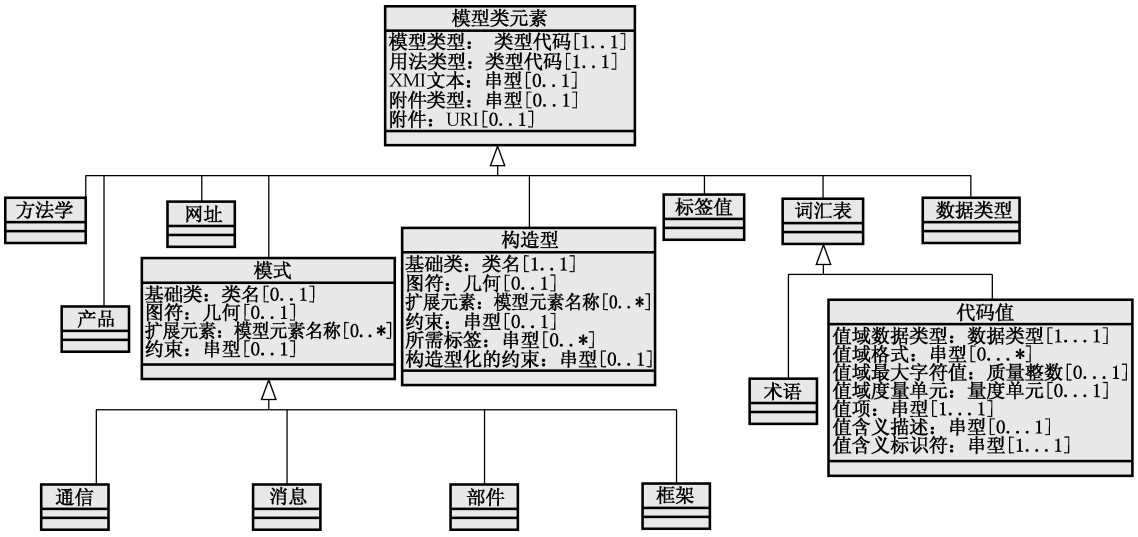

图 C.1 模型类元素包的示例

C.2 构造型

MOF 中没有与“构造型”对应的元类。然而,UML 元模型提供了与构造型相关的规范作为扩展机制。构造型是 UML 的 3 种扩展机制之一。每个构造型的含义用轮廓和元模型作出规约。它类似于一个虚拟的元模型构件的实例。实例具有与非构造型模型元素相同的结构(如属性、关联和操作)。构造型可以为实例规定额外的约束和所需的标签值。构造型也可以用来指出两个相同的结构化模型元素在含义和用法上的区别。

每个类按照如下方式进行描述:

a) 超类

模型类元素。

b) 属性

基础类:类名[1..1]。

图符:几何图形[0..1]。

被扩展的元素:模型元素的名称[0..*]。

约束:串型[0..1]。

所需标签:串型[0..*]。

构造型约束:串型[0..1]。

C.3 代码值

代码值是一个元类,它将代码值指派为模型构件。MOF 中没有与“代码值”对应的元类。元模型“代码值”通过模型构件从 MOF 中继承了类元素。可以为模型元素的枚举数据类型规定代码值。

在 GB/T 18391 中,枚举数据类型的每个值及其含义可以被定义和注册为一个值域。值域和代码值之间具有“指代”型链接,代码值及其含义可以注册在 GB/T 18391 的框架中。

每个类按照如下方式进行描述:

a) 超类

模型类元素。

b) 属性

值域数据类型:数据类型[1..1]。

值域格式:串型[0..*]。

值域最大字符长度:整数[0..1]。

值域的度量单元:度量单元[0..1]。

值项:串型[1..1]。

值含义标识符:串型[1..1]。

值含义描述:串型[0..1]。

C.4 模式

模式是一种元类,用以指代一种将模式定义为模型元素并应用其机制的设施。元元模型元素(如交互、部件和框架)是模式的子类。

模式用包和参数化合作图的方式予以定义。在模式定义的模型元素中,类名、属性、关联名、关联端点和操作名可以被规定为形式化参数。允许出现嵌套的模式定义。应用具有实际值的模式可以得到新

的基于模式的交互图。在展开过程中，如有必要，可使用重命名和过滤机制（对名称实施修改或者隐藏）。

从元模型的抽象句法中导出的表达式可以被定义为模式。

模式从 MOF 中间接继承了包和类元素。

每个类按照如下方式进行描述：

a） 超类

模型类元素。

b） 属性

基础类：类名称[1..1]。

图符：几何图形[0..1]。

被扩展的元素：模型元素名称[0..*]。

约束：串型[0..1]。

C.5 通信

通信是一种类元，它采用模式机制来指导交互建模。MOF 中没有与“通信”对应的元类。然而，元类“通信”间接继承了 MOF 中的包。

每个类按照如下方式进行描述：

a） 超类

模型类元。

b） 属性

目标系统交互：串型[1..1]。

C.6 部件

部件是一种类元，它采用模式机制来指导交互建模。MOF 中没有与“部件”对应的元类。然而，元类“部件”间接继承了 MOF 中的包。

每个类按照如下方式进行描述：

a） 超类

模型类元素。

b） 属性

目标系统部件：串型[1..1]。

C.7 框架

框架是一种类元，它采用模式机制来指导交互建模。MOF 中没有与“框架”对应的元类。然而，元类“框架”间接继承了 MOF 中的包。

每个类按照如下方式进行描述：

a） 超类

模型类元素。

b） 属性

目标系统框架：串型[1..1]。

附 录 D
（资料性附录）
级 对

D.1 概述

本附录规定了作为 MFI 注册库系统目标模型的元数据对象的级对。MFI 注册系统中包含大量的元数据对象实例。这些元数据定义在不同的层级和视图上。元数据实例指定了应用层数据构成的级对。根据特定层级专有的数据，可以对真实世界的数据进行注册。这些真实世界的数据可以作为各个层级和视图上的实例。

例如，业务过程名称和相应的元模型应分别注册为模型概念和处于上层的模型域轮廓。处于下层的、使用规范化建模构件定义的业务过程模型可注册为“指示物”。在下层，要对满足元模型规约的具体建模制品或产品实施注册。注册由建模制品或产品的开发者或提供者（负责开发和调用）来完成。

注：框架解释了不同建模层级的概念。

可选的、特定类型管理项的描述：

a） 上层（见 D.2）；

b） 上层模型（见 D.3）；

c） 上层模型元素（见 D.4）；

d） 下层（见 D.5）；

e） 下层模型（见 D.6）；

f） 下层模型元素（见 D.7）；

g） 模型视图（见 D.8）；

h） 模型层级（见 D.9）。

图 D.1 表示层级包。

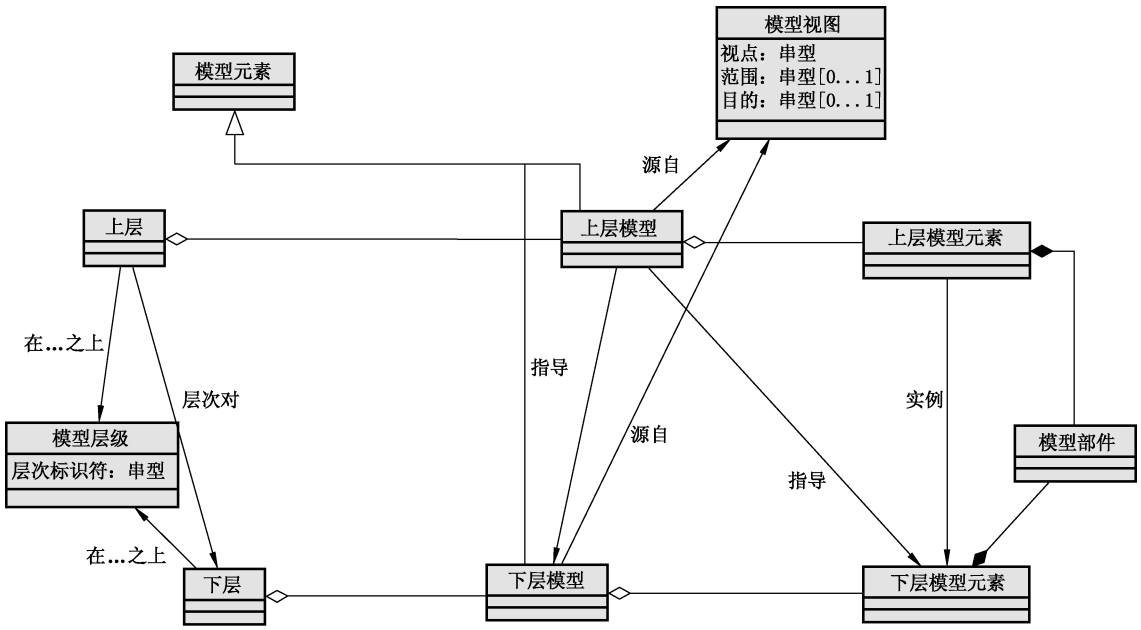

图 D.1 层级包

D.2 上层

上层是一种元类，描述了“模型层级”所指的元层。根据其层对，上层应该具有一个模型层级。上层聚合了相应元级上的元模型。

上层对应于 MOF 元数据体系结构的某个元级。如果上层是 M3 级，则下层是 M2 级；类似的，如果上层是 M2 层，则下层是 M1 层。

每个类按照如下方式进行描述：

a） 超类

无。

b） 属性

无。

c） 参考

在...之上：模型层级[1..1]。

级对：下层[0..*]。

D.3 上层模型

上层模型是一种元类，它规定了位于上层的模型或元模型。上层模型可以从不同的模型视图进行定义。通过关联和参考，上层模型之间可以建立链接。

上层模型对应于 MOF 元数据体系结构中某个元级的模型。MOF 具有处理自反操作的设施。在 MFI 中，元模型可以扩展为元对象的集合，下层模型的操作允许处理上层元模型的信息。

每个类按照如下方式进行描述：

a） 超类

模型类元素。

b） 属性

无。

c） 参考

源自：模型视图[1..1]。

d） 约束

上层模型必须与相应的下层模型关联。

下层模型应该根据上层模型定义的抽象句法和其他约束进行描述。

D.4 上层模型元素

上层模型元素是一种元类，它指定了上层模型的复合元素。上层模型元素是模型部件的聚合，模型部件有可能包括选中的模型。

每个类按照如下方式进行描述：

a） 超类

无。

b） 属性

无。

c） 参考

无。

d） 约束

下层模型元素应该是相应的上层模型元素的实例。

D.5 下层

下层是一种元类，描述了“模型层级”所指的模型层。下层具有指向上层建模目标链接。下层对应于 MOF 元数据体系结构中的某个元层级。

每个类按照如下方式进行描述：

a） 超类

无。

b） 属性

无。

c） 参考

在...之上：模型所处的层级 [1..1]。

级对：上层[0..*]。

D.6 下层模型

下层模型是一种元类，它规定了位于下层的模型或元模型。下层模型可以从不同的建模视图进行定义。通过关联和参考，下层模型之间可以建立链接。

下层模型对应于 MOF 元数据体系结构中某个元层级的模型。MOF 具有处理自反操作的设施。在 MFI 中，元模型可以扩展为元对象的集合，下层模型的操作允许处理上层元模型的信息。

每个类按照如下方式进行描述：

a） 超类

模型类元素。

b） 属性

无。

c） 参考

源自：模型视图[1..1]。

d） 约束

下层模型必须与相应的上层模型关联。

下层模型应该根据上层模型定义的抽象句法和其他约束进行描述。

D.7 下层模型元素

下层模型元素是一种元类，它指定了下层模型的复合元素。下层模型元素的作用主要是对模型选择进行组装。

下层模型元素由模型选择和 MOF 中的元模型构件组成。

每个类按照如下方式进行描述：

a） 超类

无。

b） 属性

无。

c） 参考

无。

d）约束

下层模型元素应该是相应的上层模型元素的实例。

D.8 模型视图

模型视图是一种元类，它指定了建模视点。模型视图可以根据特定的本体进行分类和识别。在GB/T 18391中，应该根据分类模式注册本体。

每个类按照如下方式进行描述：

a） 超类

无。

b） 属性

视点：串型。

范围：串型[0..1]。

目的：串型[0..1]。

c） 参考

无。

d） 约束

无。

D.9 模型层级

模型层级是一种元类，它指定了元数据体系结构的元级。模型层级具有指向建模目标的建模语境的链接，以及指向模型元素名称空间的建模概念的链接。

元类“模型层级”对应于MOF元数据体系结构的某个元层级。

每个类按照如下方式进行描述：

a） 超类

无。

b） 属性

级标识符：串型[1..1]。

c） 参考

无。

d） 约束

无。

附 录 E
（资料性附录）
记 法 概 述

E.1 概述

核心模型提供了一个基于 GB/T 18391 的分类机制，以及一个增强型的含义三角形作为注册系统的基础。

在 MFI 的注册系统中，注册上层概念时需要定义相应的记号及其含义。作为上层概念的实例，下层概念可以注册为一组部件。

E.2 基本记法

“模型记号”“模型概念”“模型部件集”等记法对于部件分类是十分必要的。

a） 指示概念的记号（模型记号）

记号是概念的指代，可以引发概念。模型记号指派了概念的特定对象。模型记号可以使用不同的方式来表达，如字符串、句子、代码、图符等。模型记号提供了命名设施，以便通过符号来指代已注册的目标对象。

b） 域中的概念（模型概念）

概念由特定域的模型类元素进行规范。模型类元素规定了概念的含义。概念由记号（可采取模型或规约的形式）引发。模型类元素在模型域轮廓中有具体定义。可以通过元模型（一种上层模型）给出概念的形式化定义，同时提供多个相关标准和文本的轮廓。

c） 记号所指代的概念对应的模型部件集（模型部件集）

模型部件集是模型概念对应的一组构件，通过模型概念对应的模型类元进行规约。模型概念对应的模型部件集（由相应的模型记号指派），可以注册为上层元模型（或者上层模型）推导出来的模型部件集，如一组对象、模型、模式、构造型、数据元素等。

d） 根据用户目的（模型选择）选定的模型部件

模型选择建立了模型记号和模型部件集之间的关联。用户可以根据特定的目的从模型部件集中选择实例并进行注册。例如，可以注册一组来自特定域的模型部件；在其他域复用这些部件集时，就需要进行模型选择。一般来说，对象由选中的元素所组成。模型部件可以通过组合选定模型部件的方式得到。另外，模型选择的实例可以被模型支持工具使用。

按照这种方式，部件可以根据有概念规约的记号进行分类。记号只是部件的附加物。这里没有对记号本身进行分类。但是，可以分析记号和已注册部件之间的关系，而这些部件是依据记号进行分类的。通常，很难定义记号的分类，因为记号可以有很多含义，因此，创建词汇表样式的含义树比较困难。

图 E.1 表达了核心模型的主要思想。在这个图中，对上面提到的概念进行了元建模。在这个例子中，记号“国家”引发了相应的“国家”部件。模型域轮廓“国家包”和模型类元素“国家和地区”类被赋予了模型概念。

E.3 元类的角色

可以为同一个模型概念注册多个不同的模型部件集。模型选择记录了一组模型记号和模型部件集，以描述制品在特定社区中的共享。模型选择的条件属性可以用于从模型部件集中选择合适的实例。条件属性所扮演的角色类似于过滤器。在规约中，不允许选择一个模型概念的所有实例。但是，应该允

许对一个特定的模型概念的所有实例提供遍历服务。此外，模型选择还扮演了连接器的角色，在模型部件组合中发挥作用。

模型类元素可以通过模型域轮廓中遵循 MOF 的模型来描述。模型类元素扮演了类型化模型的角色，模型域轮廓起到了指定特定域知识的作用。

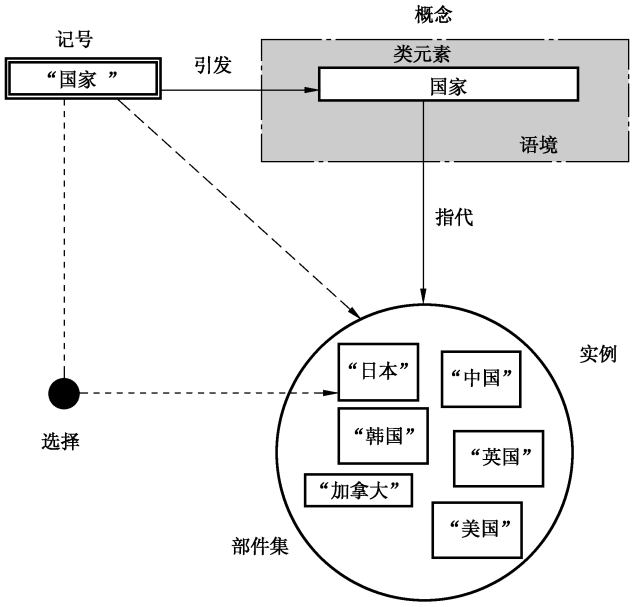

图 E.1　增强的意义三角形和选择

图 E.2 示出了本部分的基本框架。在这个框架中：

——模型记号有一个名称空间和一个命名元素的记号。

——模型概念拥有模型类元素和模型域轮廓。模型类元素的含义在模型域轮廓中做出了定义。

——模型部件集由一组模型部件组合而成。模型部件和模型类元素之间的关系可以通过概念化类型予以分类。

——模型选择可以用于连接模型记号和模型部件集。模型记号指代特定的模型部件集。模型记号和模型部件集可以连接到同一个模型概念。

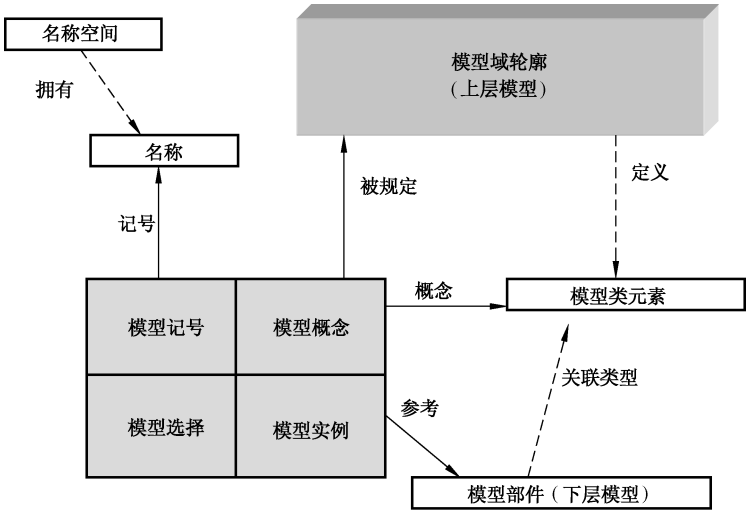

图 E.2　已注册目标对象的基本关系

E.4 简单的例子

图 E.3 是一个关于“车辆”的例子。

这个图被分为 4 部分，包括基于 MFI 的注册系统、目标模型存储库、目标名称空间和选定的模型区域。MFI 注册系统的区域与图 E.2 相同。实际的目标模型位于目标模型注册区域中。

图中的模型包括“车辆包”和几个模型部件（如“小轿车”“公共汽车”等）。“车辆包”定义了“车辆”类，“车辆”是一个模型类元素。记号“车辆”位于名称空间区域。

指向这些目标模型的指针用带箭头的线来表示，这些指针应被注册到基于 MFI 的注册系统中：

a) 通过一个有向箭头，部件（源自模型部件）可以反向追溯到特定的概念（源自模型概念）。

b) 实例（源自模型部件集）包含特定的类元素（源自模型类元素）和选择（源自模型选择）。

c) 模型选择可以用于从记号值（源自模型记号）和实例（模型部件集）中选取合适的配对。

如果需要，用户可以使用其他可选的值来组合自己的模型。通过模型选择，多部件的结构可以被表示为已注册的目标对象。

图 E.4 示出了与模型选择相关的已注册目标对象的例子。与上层模型相关联的下层模型本身也可以被注册为某个上层模型的模型记号。

图 E.5 示出了一个分层结构的例子，如下层模型充当上层模型。

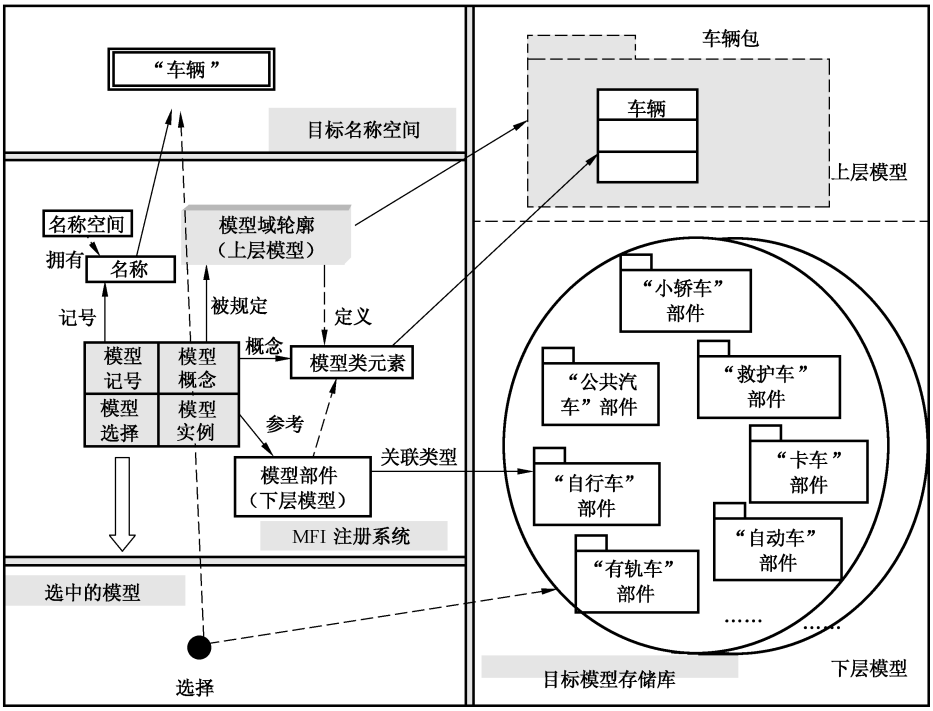

图 E.3 关于“车辆”的已注册目标对象

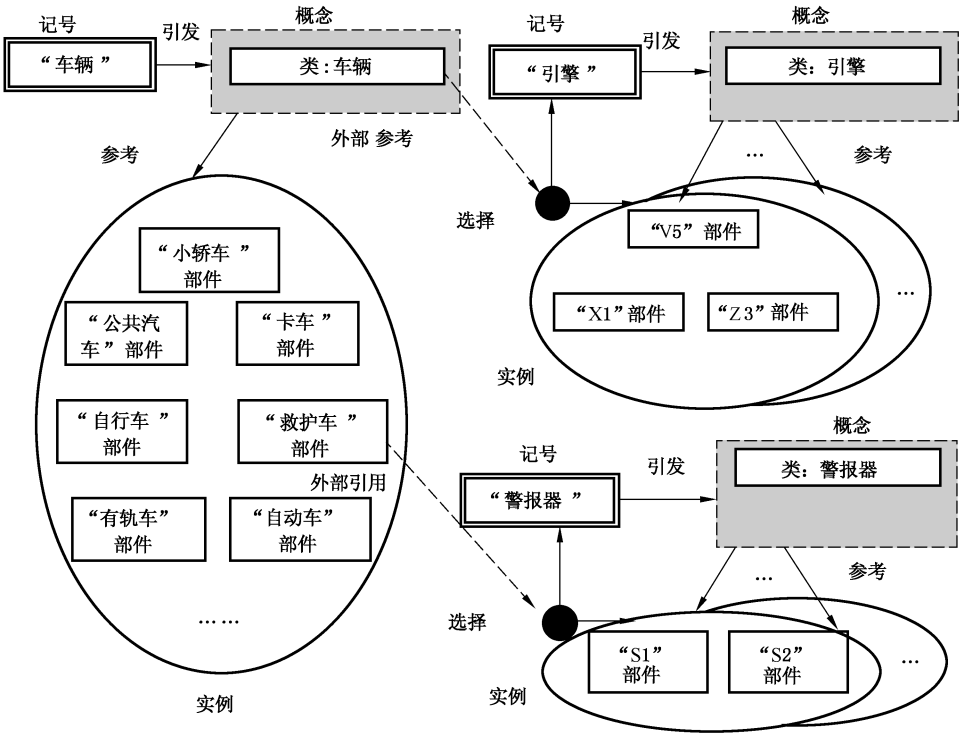

图 E.4　与选择有关的已注册目标对象

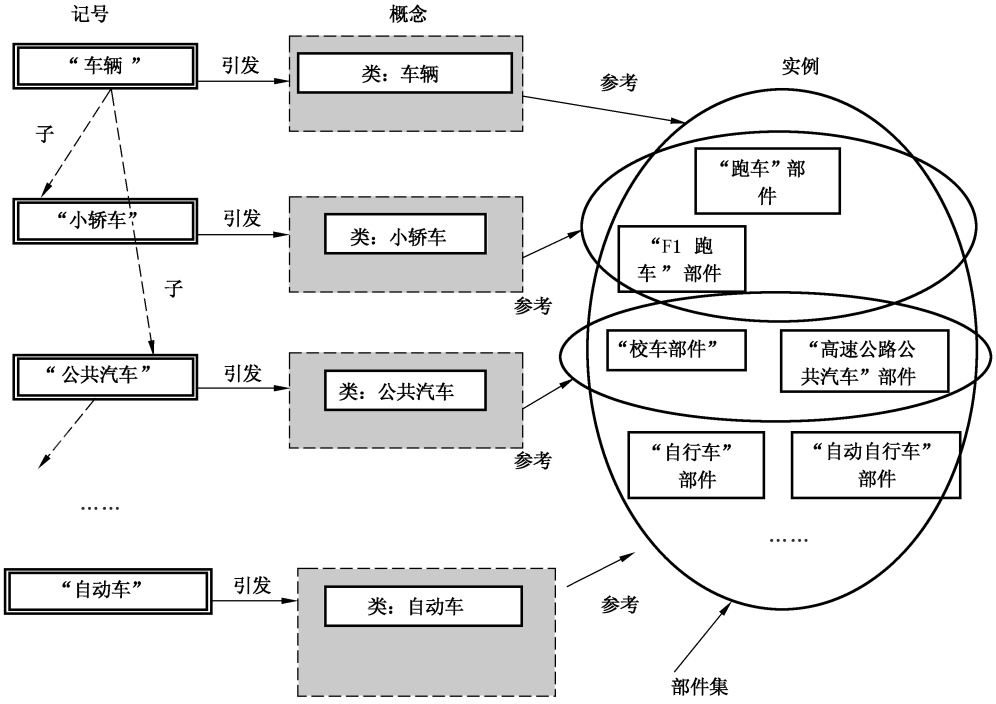

图 E.5　具有多层注册的分层目标对象

附 录 F
（资料性附录）
概 念 化

F.1 概述

“增强的含义三角形”能够对目标部件实施多视图、多层次和多范围的分类。部件和部件集可以注册为下层概念。总的来说，从不同的意图出发，每个部件可以呈现出不同的特征。因此，会有多种方法对目标部件实施分类。因而，需要使用概念化类型来定义部件之间的关系。

F.2 部件和分类

在 MFI 中，对象在注册到元数据注册系统时，没有区分元层次和模型类元素。换句话说，引入模型部件的目的是将每个对象看做是一个统一的对象。

通常，模型部件并不单独存在，而是包含或者关联了其他模型部件（比如模型类元素）的复杂结构。例如，某个模型部件的模型类元素“消息”的定义有可能涉及其他消息、术语、代码值和数据类型等部件。这样，组合而成的模型部件自身会有多种特征，因此需要从不同角度进行分类。

总的来说，在创建被注册的目标对象时，其相互之间具有依赖关系，由此形成的复杂结构也可以被作为注册对象。

这种复杂结构可以从下面 3 个角度去看待：

a） 多森林结构

不同的标准系列由不同的标准组织独立开发完成。在注册过程中，相关标准有可能相互之间产生联系，从而超越了标准开发组织内部的边界。这种注册对象必须形成由标准模型组成的森林结构。如果有必要，需要采取协调行动来开发相关的轮廓以实现这些标准之间的互操作。

b） 多层次结构

在各种标准构成的森林中，术语和概念应该由系统进行控制和维护，而其中的上下层关系也应该定义清晰。在注册时，上层可以注册为模型类元素，下层可以注册为模型部件。这种关系形成了目标对象的多层结构。

c） 多部件结构

模型部件可以包含作为外部参考对象的其他部件。这种关系构成了多部件结构。同样的，模型类元素也可以包含从其他模型部件集中选取的外部对象作为元素。

为了实现元模型、模型和可共享的制品之间的互操作，定义这样的概念是非常重要的。然而，这种概念很难作出准确的定义，在相关的社团之间取得共识也很困难。因此，需要使用标准和轮廓来指定基于共识的典型概念集。

F.3 概念和部件集之间的概念化类型

模型部件源自模型概念，同时它与模型概念也具有关联。核心模型定义了 4 种概念化类型。抽象句法-表达式关系是元模型和模型之间的一种关系。超-子关系的含义是，模型实例是定义为超类的模型概念的子类。抽象句法-表达式关系与超-子关系的区别在于：在超-子关系中，超类和子类在相同的元层上；但是，抽象句法-表达式关系会涉及不同的元层。基础-变体关系的含义是，变体模型源自基础模型，但依据特定的规则进行了修改，如参考本体和本地本体之间的关系。超-子关系可以看做是基础-变

体关系在类上的应用。

本附录提供了更多已注册的目标对象，如“国家”“亚洲国家”“地域”“政治疆界模型”。

概念化类型的例子如下所示：

示例 1：概念化类型“Type-Instance(类型-实例)”(见图 F.1)。

示例 2：概念化类型“Type-Instance(类型-实例)”(见图 F.2)。

示例 3：概念化类型“Super-Sub(超-子)”(见图 F.3)。

示例 4：概念化类型“Base-Variant(基础-变体)”(见图 F.4，F.5)。

示例 5：概念化类型“Abstract Syntax-Expression(抽象句法-表达式)”(见图 F.6，F.7)。

另外，表 F.1 示出了在概念化类型“Base-Variant”上的操作。

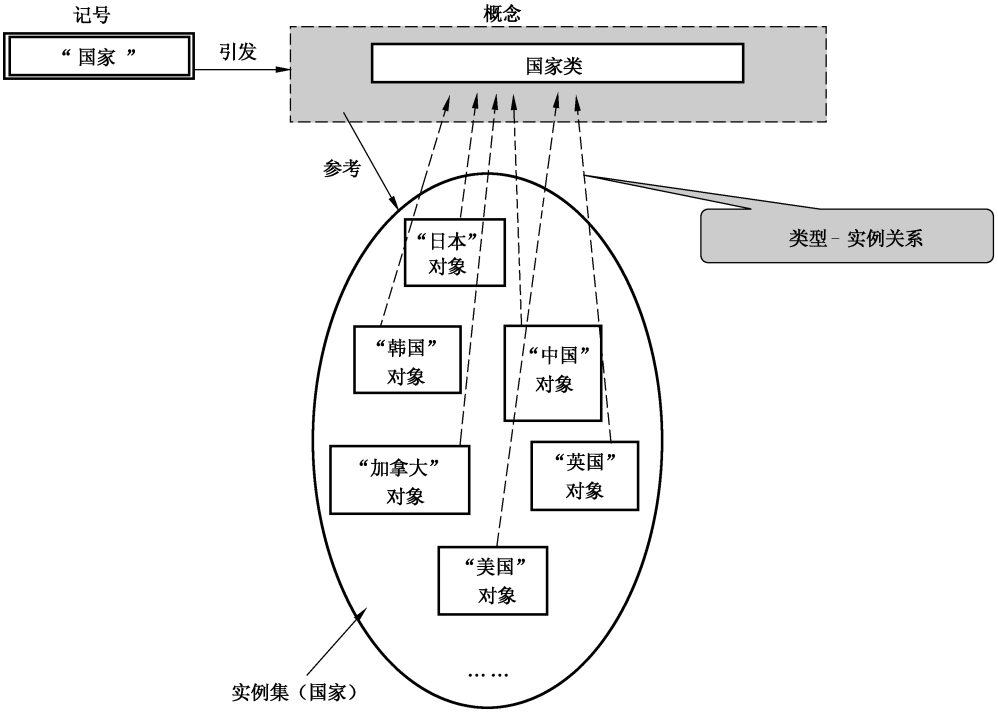

图 F.1　示例 1：概念化类型“类型-实例”(1)

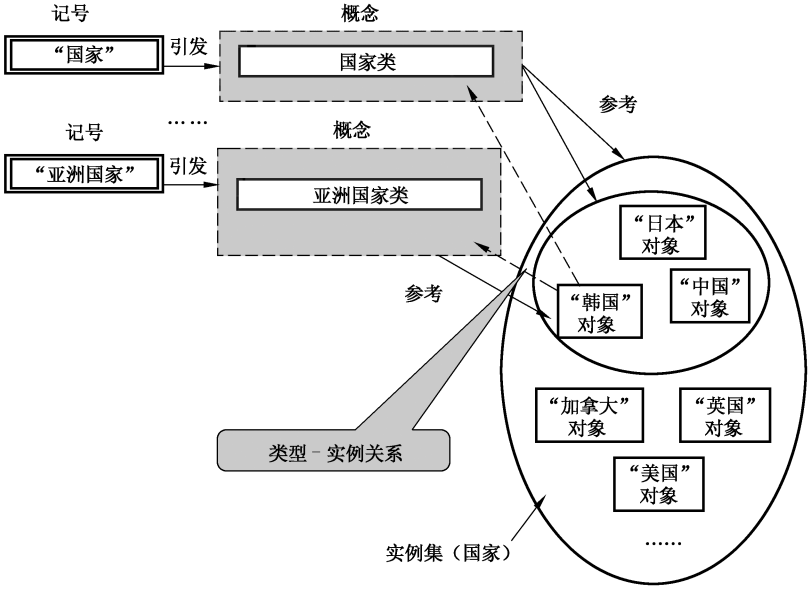

图 F.2　示例 2：概念化类型“类型-实例”(2)

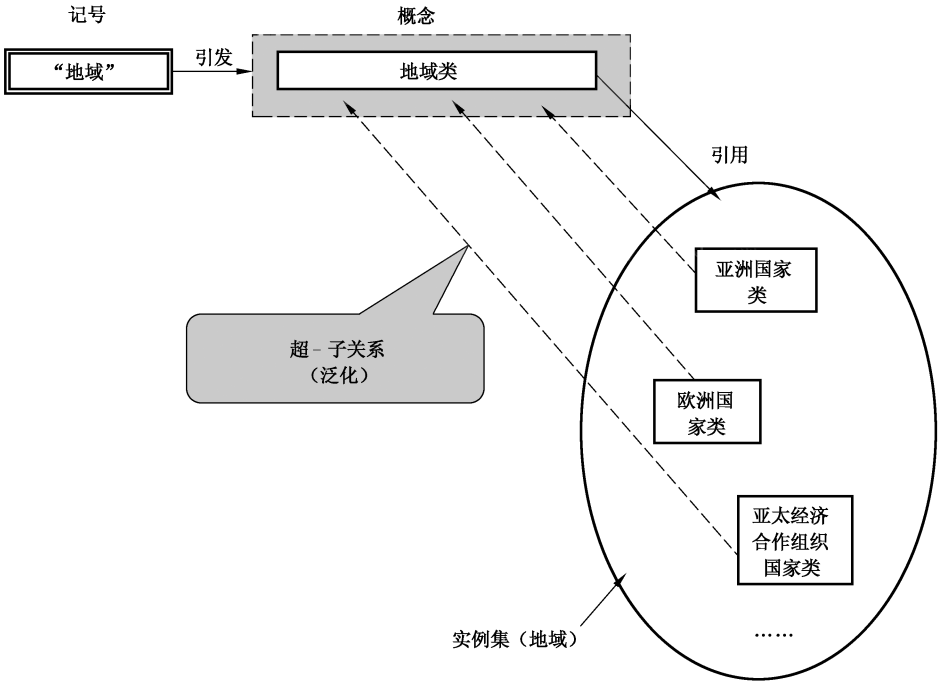

图 F.3　示例 3：概念化类型“超-子”

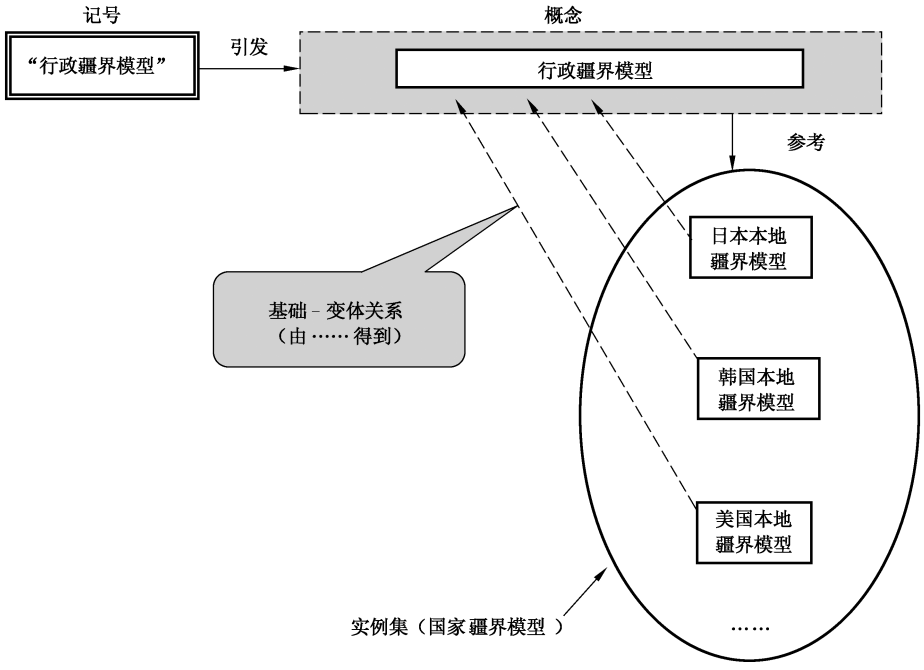

图 F.4　示例 4：概念化类型“基础-变体”(1)

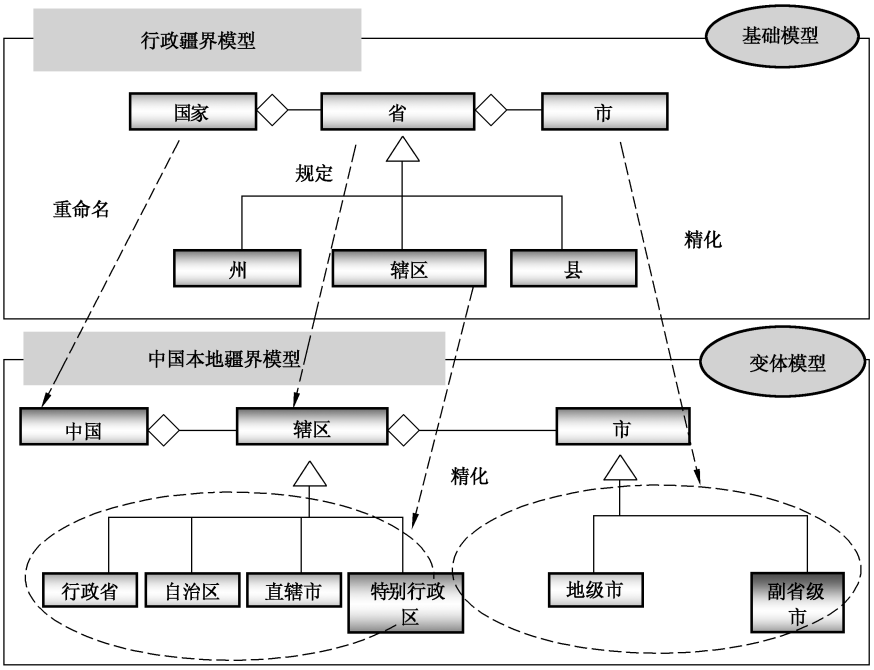

图 F.5　示例 4:概念化类型“基础-变体”(2)

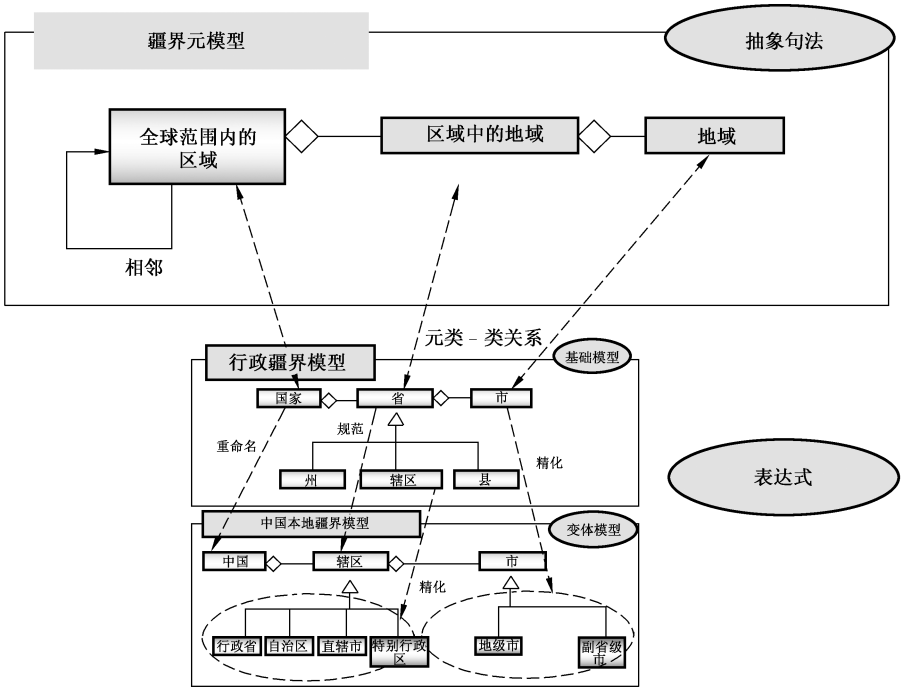

图 F.6　示例 5:概念化类型“抽象句法-表达式”(1)

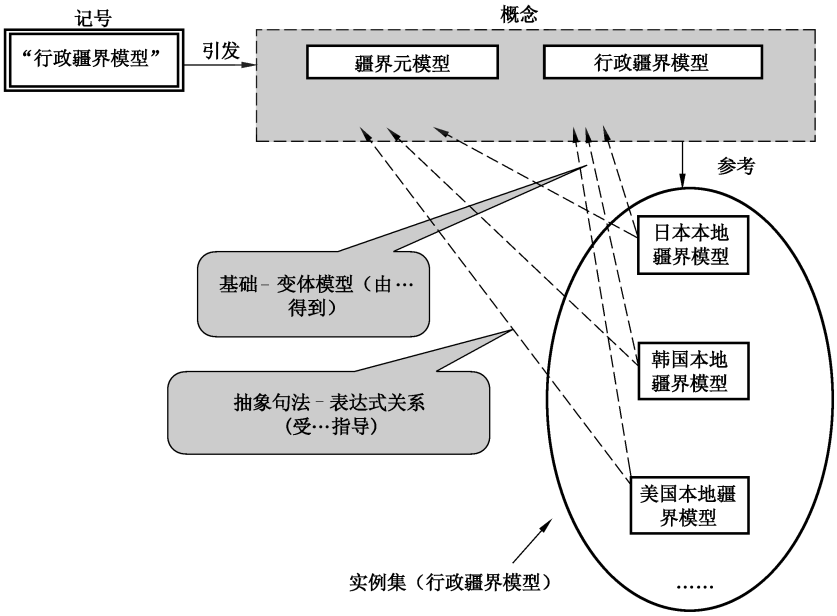

图 F.7　示例 5:概念化类型“抽象句法-表达式”(2)

表 F.1　概念化类型“基础-变体”上的操作

操　作	描　述
重命名	将类和属性的名称更换为适合语境的名称
指定	在允许的范围内指定基础模型中的模型元素。如： ——固定关联终端的多重度或限制其范围； ——为属性选择一个代码集或指定允许的值； ——选择有效的属性,并删除未使用的属性； ——在类层次中选择并且固定子类； ——删除未使用的关联
精化	修改基础模型中的模型元素。如： ——增加一个子类,且该子类增加了一些属性和操作； ——增加一个应用于特定语境的约束
替换	替换基础模型中的某些模型元素。如： ——将通用部件替换为在特定地区使用的部件； ——将旧版本的部件替换为新版本部件
扩展	在基础模型中增加新的类或关联
融合	将基础模型组合或聚集成一个新模型

附 录 G
（资料性附录）
用法和部件类型

G.1 概述

本部分定义了元模型和模型之间的关系，以及注册、共享和重用规范性建模构件的框架。在元模型体系结构的 M2 层上，可以注册由全球性标准化组织开发的、与业务概念建模相关的标准。同时，需要注册这些标准的名称空间和定义在其中的概念。

此外，还需注册遵循全局标准的模型部件集（如具体的构造型或模式）。注册系统的用户（如建模者）将选择并使用合适的构造型和模式在本地化的标准层上创建他们自己的业务对象模型。本体化的标准层具有与全局标准层类似的结构，包括名称和名称空间（模型记号）、模型域（模型域轮廓）和模型类元素（模型概念）、模型部件（模型部件集）以及选定的模型元素（模型选择）。

G.2 典型的开发者、注册者和用户

表 G.1 示出了一个不同注册者和用户依据本部分完成注册的例子。假定标准组织和注册元模型的用户处于上层，注册制品或产品的用户处于下层。

表 G.1 典型的开发者、注册者和用户

设置层	元模型元素	开发者/注册者	用 户
全局标准	模型记号	标准化组织	模式和构造型的标准制定者和开发者
	模型概念与模型域轮廓（上层模型）	标准化组织	模式和构造型的标准制定者和开发者
	模型部件集（下层模型）	模式和构造型的标准制定者和开发者	行业标准制定者
	模型选择	行业标准模型的开发组织	遵守标准模型的建模制品或产品的开发者
本地化标准	模型记号	行业标准模型的开发组织	遵守标准模型的建模制品或产品的开发者
	模型概念与模型域轮廓（上层模型）	行业标准模型的开发组织 遵守标准模型开发产品的企业	遵守标准模型的建模产品的开发者 建模制品或产品的销售商
	模型部件集（下层模型）	遵守标准模型开发产品的企业	建模制品或产品的销售商
	模型选择	（非公共的，应该在各个企业内部进行管理）	（在企业内部进行管理）

G.3 部件类型

标准开发组织定义了许多不同的部件类型。表 G.2 列出了部分可供选择的部件类型。

表 G.2 部件类型的候选项

序号	构件类型名	代　　码	说　　明
1	UN/CEFACT	0001	电子商务模型
1.1	UN/CEFACT 数据类型	00010001	
1.2	UN/CEFACT 核心部件	00010002	
2	OASIS	0002	电子商务模型
2.1	OASIS.ebXML.业务过程	00020001	
2.2	OASIS.ebXML.消息	00020002	
2.3	OASIS.ebXML.交互协议概要	00020003	
2.4	OASIS.ebXML.交互协议约定	00020004	
2.5	OASIS.ebXML.注册库概要	00020005	
2.6	OASIS.ebXML.注册库网址	00020006	
3	HL7	0003	卫生保健域模型
3.1	HL7.V3.DMIM	00030001	
3.2	HL7.V3.RMIM	00030002	
3.3	HL7.V3.CMET	00030003	
3.4	HL7.V3.词汇表	00030004	
3.5	HL7.V3.CDA.第 1 版概要	00030005	
3.6	HL7 标准.V3.CDA.第 2 版概要	00030006	
4	OMG	0004	建模设施和轮廓
4.1	OMG UML.产品	00040001	
4.2	OMG.UML.轮廓	00040002	
4.3	OMG.CWM	00040003	
4.4	OMG.EDOC	00040004	
5	ISO/IEC JTC1/SC32/WG2	0005	元数据交换
5.1	MDR 注册库网址	00050001	
5.2	基于 MFI 的注册库网址	00050002	

G.4 注册过程

本章提供了一个“车辆”注册过程的示例。

G.4.1 元模型的例子

图 G.1 和图 G.2 是一个关于“车辆”注册过程元模型的示例。

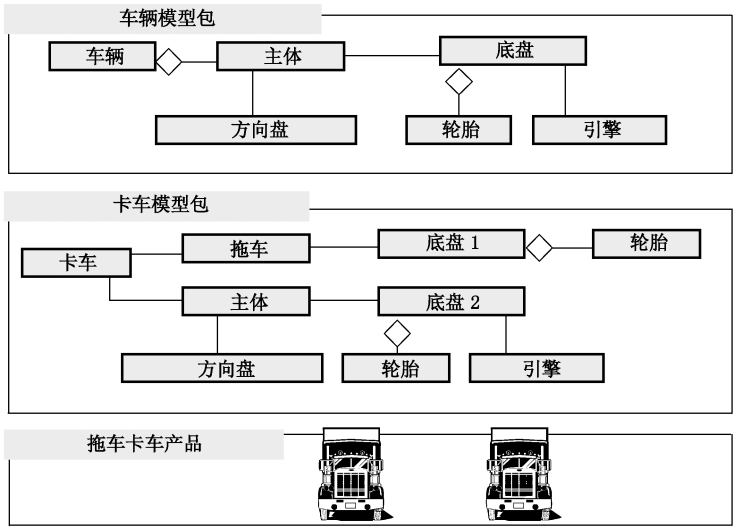

图 G.1　元模型示例(1)

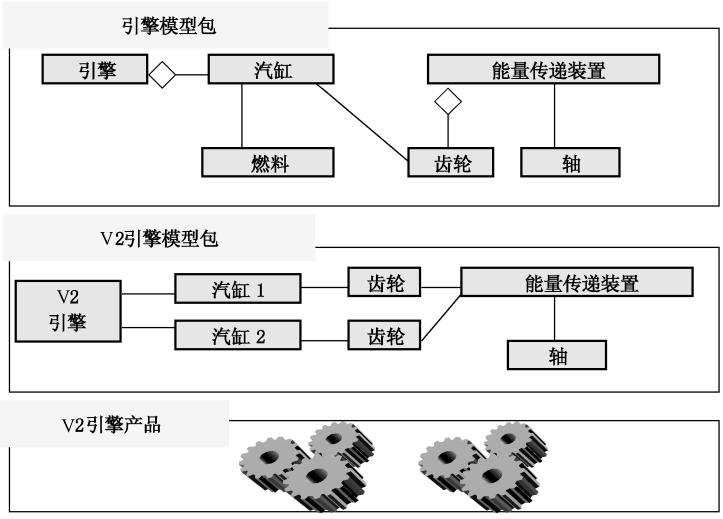

图 G.2　元模型示例(2)

G.4.2　模型部件

图 G.3 描述了模型部件的注册。

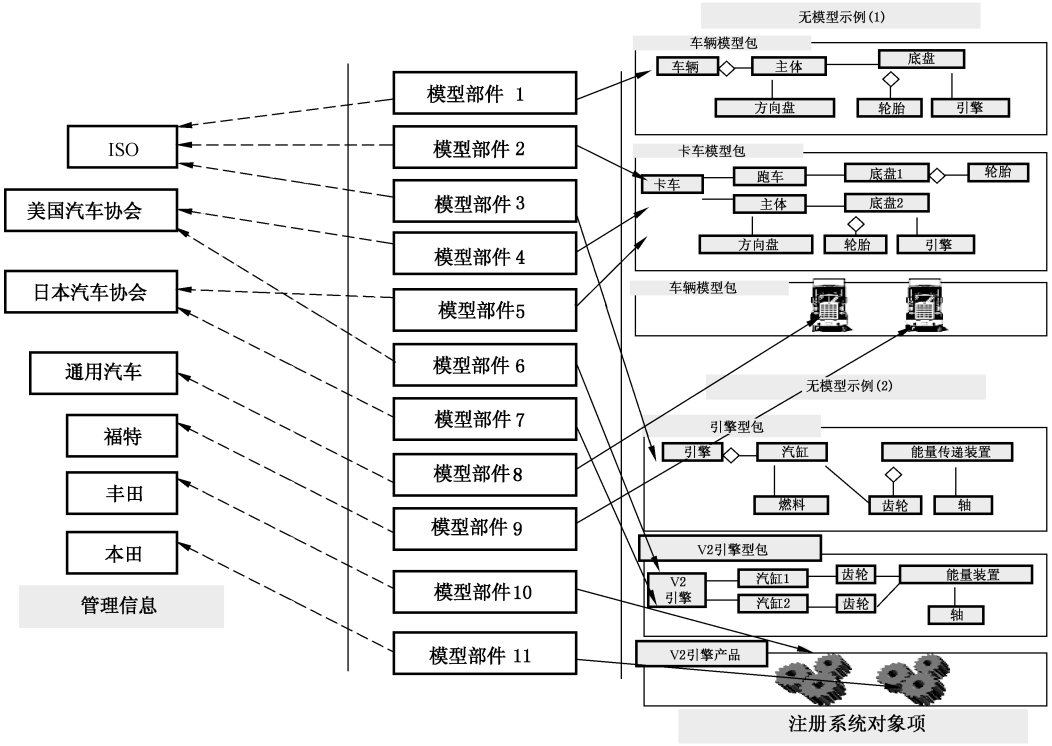

图 G.3　模型部件的注册

表 G.3 列出了模型部件。

表 G.3　模型部件

序号	实例	名称(构件/注册机构标识符/版本)	备注
1	模型部件 1	车辆模型包/ISO/1.0 版	
2	模型部件 2	卡车模型包 1/ISO/2.1 版	
3	模型部件 3	引擎模型包/ISO/1.0 版	
4	模型部件 4	卡车模型包 2/美国汽车协会/1.0 版	
5	模型部件 5	卡车模型包 3/日本汽车协会/2.0 版	
6	模型部件 6	V2 引擎模型包 1/美国汽车协会/1.0 版	
7	模型部件 7	V2 引擎模型包 2/日本汽车协会/2.0 版	
8	模型部件 8	拖车产品 1/通用汽车/3.0 版	
9	模型部件 9	拖车产品 2/福特/2.0 版	
10	模型部件 10	V2 引擎产品/丰田/3.0 版	
11	模型部件 11	V2 引擎产品/本田/2.0 版	

示例：

模型部件 1：车辆模型包/ISO/1.0 版

“模型部件 1”是一个名称为“车辆模型包”的模型部件。

实例由注册机构“ISO”负责注册，版本号为 1.0。

它包含多个模型类元素，包括“车辆”“主体”“底盘”“方向盘”“轮胎”“引擎”。

G.4.3 模型域轮廓

图 G.4 描述了模型域轮廓的注册。

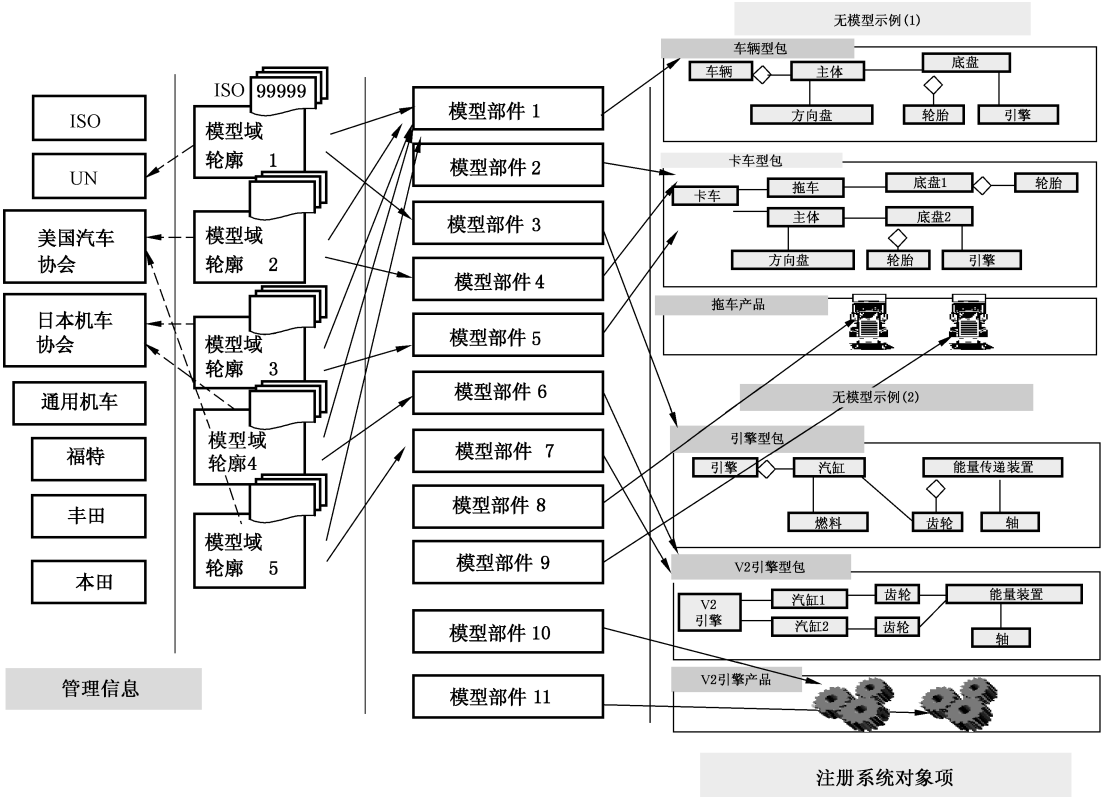

图 G.4 模型域轮廓的注册

表 G.4 列出了模型域轮廓。

表 G.4 模型域轮廓

序号	实例	名称(部件/注册机构标识符/版本)	部件
1	模型域轮廓 1	车辆轮廓/联合国/1.0 版	模型部件 1,模型部件 3
2	模型域轮廓 2	卡车轮廓 1/美国汽车协会/2.1 版	模型部件 1,模型部件 4
3	模型域轮廓 3	卡车轮廓 2/美国汽车协会/1.0 版	模型部件 1,模型部件 5
4	模型域轮廓 4	引擎轮廓 1/日本汽车协会/2.0 版	模型部件 1,模型部件 6
5	模型域轮廓 5	引擎轮廓 2/日本汽车协会/1.0 版	模型部件 1,模型部件 7

示例:

模型域轮廓 1:车辆轮廓/联合国/1.0 版

“模型域轮廓 1”是一个名为“车辆轮廓”的模型域轮廓。

实例由注册机构“联合国”负责注册,版本号为 1.0。

它包含模型部件:“车辆模型包(模型部件 1)”“引擎模型包(模型部件 3)”。

它由模型规约“ISO99999”描述。

它符合图 G.1 遵循 MOF 的模型,该模型使用的模型类元素包括:“车辆”“主体”“底盘”“方向盘”“轮胎”“引擎”。

G.4.4 模型概念

图 G.5 描述了模型概念的注册。

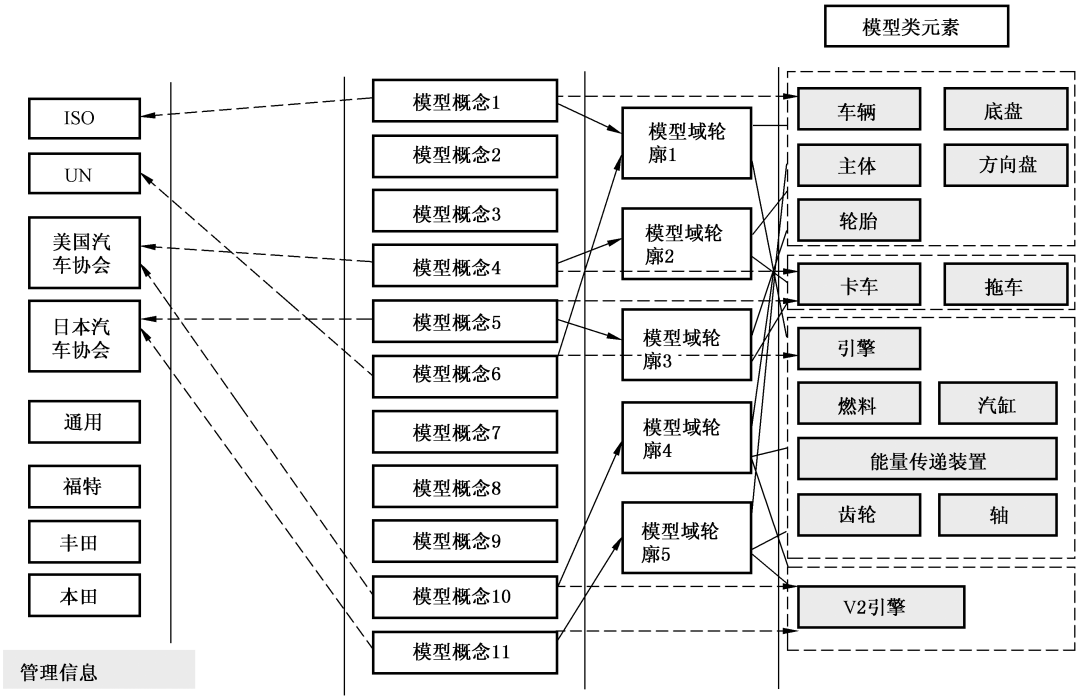

图 G.5 模型概念的注册

表 G.5 列出了模型概念。

表 G.5 模型概念

序号	实例	名称（概念/域/注册机构标识符/版本）	类元素
1	模型概念 1	车辆/车辆轮廓/ISO/1.0 版	车辆
2	模型概念 2		
3	模型概念 3		
4	模型概念 4	卡车/卡车轮廓 1/美国汽车协会/1.0 版	卡车
5	模型概念 5	卡车/卡车轮廓 2/日本汽车协会/2.0 版	卡车
6	模型概念 6	引擎/引擎轮廓 1/联合国/1.0 版	引擎
7	模型概念 7		
8	模型概念 8		
9	模型概念 9		
10	模型概念 10	V2 引擎/引擎轮廓 2/美国汽车协会 /3.0 版	V2 引擎
11	模型概念 11	V2 引擎/引擎轮廓 3/日本汽车协会 /2.0 版	V2 引擎

示例：

模型概念 1：车辆/车辆轮廓/ISO/1.0 版

“模型概念 1”是一个名为“车辆”的模型概念，它符合模型域轮廓“车辆轮廓（模型域轮廓 1）”。

实例由注册机构“ISO”负责注册，版本号为 1.0。

它由模型类元素“车辆”进行分类。

G.4.5 模型记号

图 G.6 表示模型记号的注册。

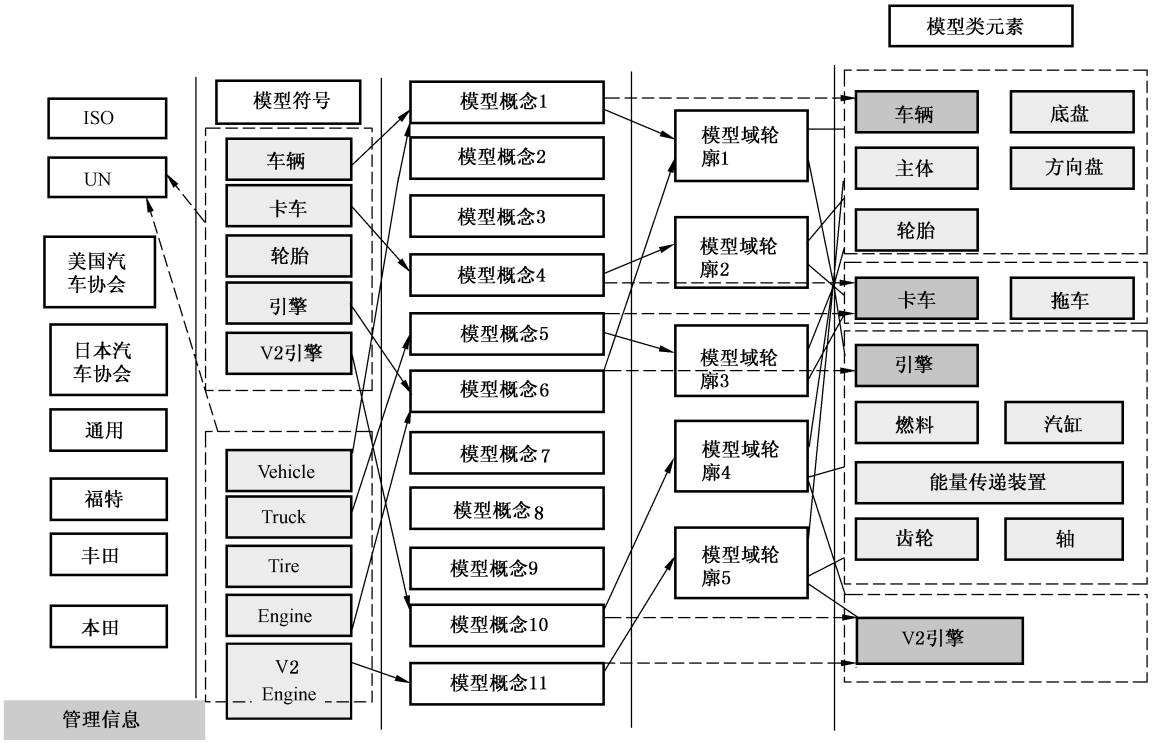

图 G.6 模型记号的注册

表 G.6 表示模型记号的信息。

表 G.6 模型记号

序号	实例	名称(记号/概念/域/注册机构标识符/版本)	类元素
1	模型记号 1	车辆/车辆/车辆轮廓/UN/1.0 版	车辆
2	模型记号 2	卡车/卡车/卡车轮廓 1/联合国/1.0 版	卡车
3	模型记号 3		
4	模型记号 4	引擎/引擎/引擎 1/联合国/1.0 版	引擎
5	模型记号 5	V2 引擎/V2 引擎/引擎轮廓 1/联合国/1.0 版	V2 引擎
6	模型记号 6		
7	模型记号 7	Vehicle/车辆/车辆轮廓/联合国/1.0 版	车辆
8	模型记号 8	Truck/卡车/卡车轮廓 2/联合国/1.0 版	卡车
9	模型记号 9		
10	模型记号 10	Engine/引擎/引擎轮廓 2/联合国/1.0 版	引擎
11	模型记号 11	V2 Engine/引擎/引擎轮廓 2/联合国/1.0 版	V2 引擎

示例：

模型记号 1：车辆/车辆/车辆轮廓/UN/1.0 版

“模型记号 1”是一个名为“车辆”的模型记号，它通过模型域轮廓“车辆轮廓(模型域轮廓 1)”来指派模型概念“车辆(模型概念 1)”。

实例由注册机构“联合国”负责注册，版本号为 1.0。

它由模型类元素“车辆”进行分类。

G.4.6 模型部件集和模型选择(1)

图 G.7 表示模型部件集和模型选择的注册。

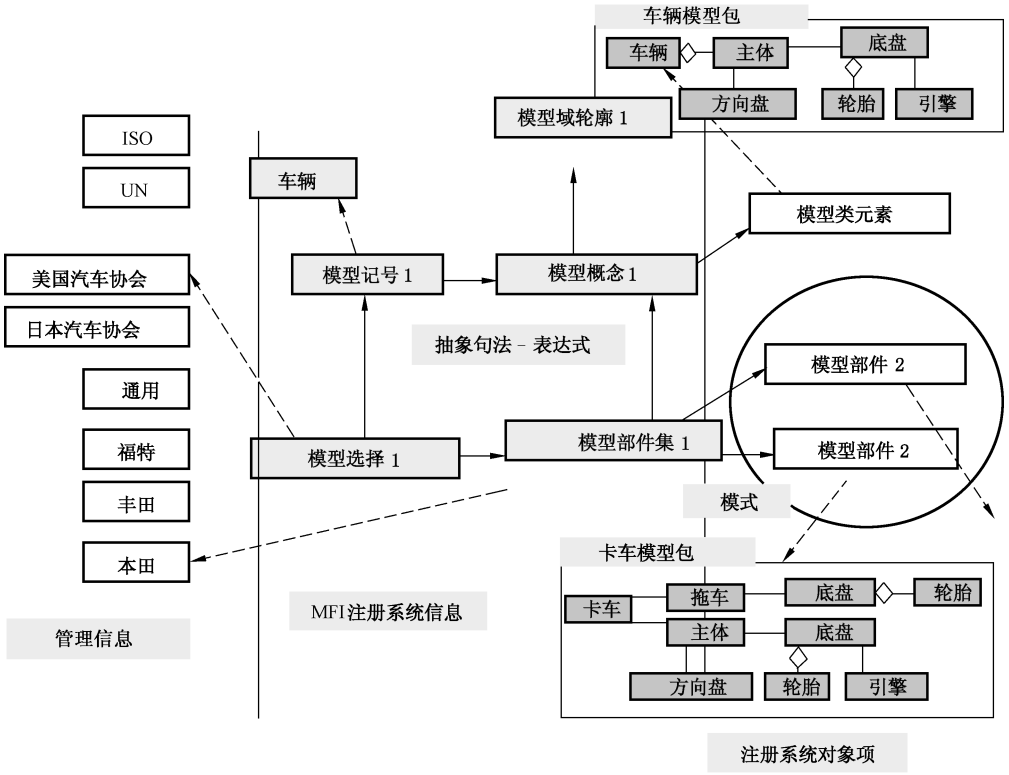

图 G.7 模型部件集和模型选择的注册

表 G.7 表示模型部件集和模型选择。

表 G.7 模型部件集和模型选择

序号	实例	名称(部件集/概念/注册机构标识符/版本)	类型
1	模型部件集 1	车辆集 1/车辆/福特/2.0 版	抽象句法-表达式关系，模式
2	模型部件集 2	名称(选择/记号/部件集/注册机构标识符/版本)	
3	模型选择 1	美国允许的车辆/车辆/车辆集 1/美国汽车协会/1.0 版	
4	模型选择 2		

示例 1：

模型部件集 1：车辆集 1/车辆/福特/2.0 版

“模型部件集 1”是一个名为“车辆集 1”的模型部件集，通过“抽象句法-表达式关系”可以抽象为模型概念“车辆(模型概念 1)”。

实例由注册机构“福特”负责注册，版本号为 2.0。

它包含模型部件“卡车模型包 2(模型部件 4)”和“大篷车模型包(模型部件 12)”，选定的模型部件可以构成特定的模式。

示例 2：

模型选择 1：美国允许的车辆/车辆/车辆集 1/美国汽车协会/1.0 版

“模型选择 1”是一个名为“美国允许的车辆”的模型选择，它通过模型记号“车辆(模型记号 1)”来表达，选择了模型部件集“车辆集 1(模型部件集 1)”。

实例由注册机构“美国汽车协会”负责注册，版本号为 1.0。

G.4.7 模型部件集和模型选择(2)

图 G.8 表示模型部件和模型选择的注册。

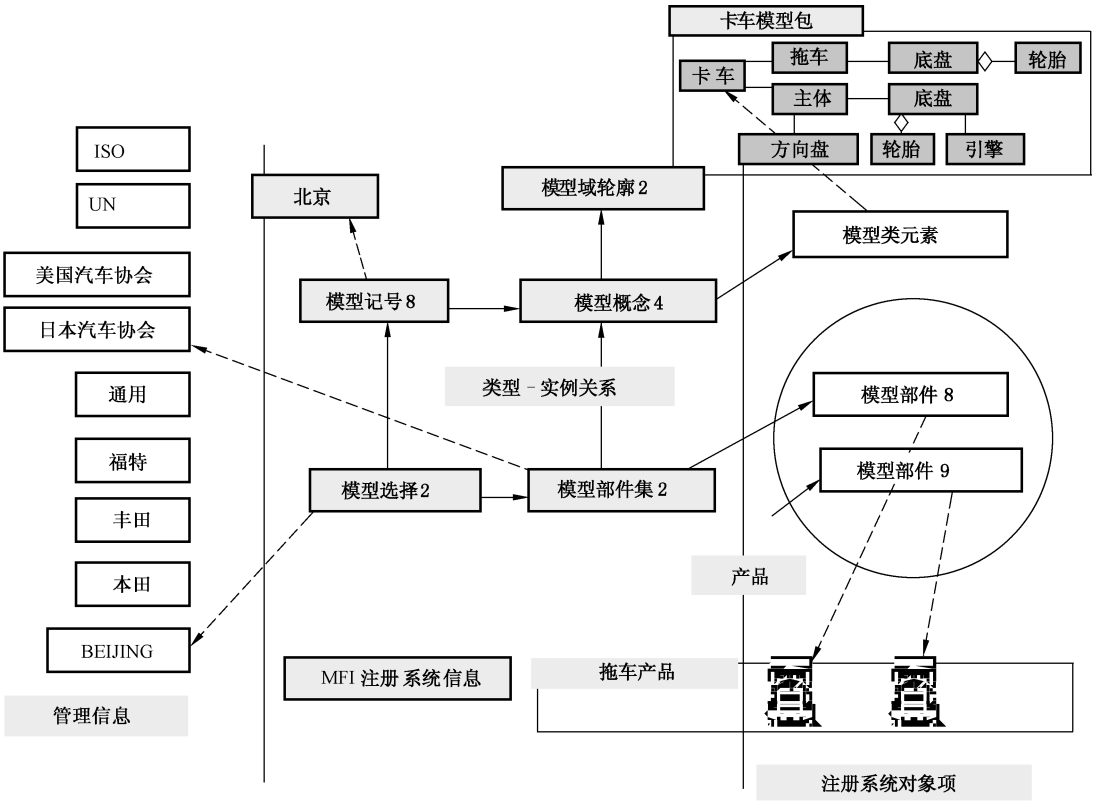

图 G.8 模型部件和模型选择的注册

表 G.8 表示模型部件和模型选择。

表 G.8 模型部件和模型选择

序号	实例	名称(部件集/概念/注册机构标识符/版本)	类型
1	模型部件集 1		
2	模型部件集 2	卡车集 1/卡车/日本汽车协会/2.0 版	类型-实例关系，产品
		名称(选择/记号/部件集/注册机构标识符/版本)	
3	模型选择 1		
4	模型选择 2	北京允许的卡车/Truck/卡车集 1/北京市政府/1.0 版	

示例 1：

模型部件集 2：卡车集 1/卡车/日本汽车协会/2.0 版

“模型部件集 2”是一个名为“卡车集 1”的模型部件集，它通过“类型-实例关系”概念化为模型概念“卡车（模型概念 1）”。

实例由注册机构“日本汽车协会”负责注册，版本号为 2.0。

它包含模型部件“拖车卡车产品 1（模型部件 8）”和“拖车卡车产品 2（模型部件 9）”，选中的模型部件可以作为产品。

示例 2：

模型选择 2：北京允许的卡车/Truck/卡车集 1/北京市政府/1.0 版

“模型选择 2”是一个名为“北京允许的卡车”的模型选择，它的模型记号是“Truck（模型记号 9）”，并且选择了模型部件集“卡车集 1（模型部件集 2）”。

实例由注册机构“北京市政府”负责注册，版本号为 1.0。