



中华人民共和国国家标准

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

Information technology—Metamodel framework for interoperability(MFI)—
Part 4: Metamodel for model mapping

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	III
引言	IV
1 范围	1
2 规范性引用文件	1
3 术语、定义和缩略语	1
3.1 术语和定义	1
3.2 本部分使用的通用术语	2
3.3 缩略语	3
4 模型映射元模型的结构	4
4.1 模型映射元模型概述	4
4.2 模型映射元模型所采用的定义格式	4
4.3 模型映射元模型中的元类	6
4.4 交换模型的标准格式	11
5 符合性	12
5.1 符合性的程度	12
5.2 符合性的级别	13
5.3 承诺	13
5.4 实现符合性的声明	13
5.5 注册的角色和责任	13
附录 A (资料性附录) 变换语言	14
附录 B (资料性附录) MFI 注册系统和模型映射	15
附录 C (资料性附录) 模型变换示例	16
C.1 示例 1	16
C.2 示例 2	16
C.3 示例 3 和示例 4	16
C.4 示例 5 和示例 6	17
参考文献	18
图 1 互操作性元模型框架中的包	4
图 2 MFI 的模型映射包	5
图 3 变换合并(1)	9
图 4 变换合并(2)	10
图 B.1 MFI 注册系统和模型映射的环境	15
图 C.1 HL7 和 ebXML 中的变换	16
图 C.2 MDA 中的变换	17
图 C.3 CWM 和 XSLT 间的变换	17

表 1	定义类型的代码集合	6
表 2	变换规则类型	7
表 3	变换操作类型	7
表 4	变换类型	10
表 5	元模型-模型变换组合的类型	11
表 6	模型-值变换组合的类型	12
表 7	实现符合性的程度	12
表 8	符合性级别	13

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：过程模型注册元模型(正在制定中)；
- 第 6 部分：注册规程(正在制定中)；

本部分为 GB/T 32392 的第 4 部分。

本部分按照 GB/T 1.1—2009 给出的规则起草。

本部分参考国际标准草案 ISO/IEC CD 19763-4:2007 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 业内外人士主要关注的问题。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,如每个业务领域的的数据元素、实体轮廓和值域等。

企业越来越需要站在全球市场的角度设计和开发跨企业的电子商务和电子业务解决方案。为了尽可能减少满足这些互操作性需求的开销和时间,企业必须使用公共、标准的业务模型来设计自己的信息系统。

为了开发行业内的公共模型,需要行业内各种利益相关方共同确定行业中公共的核心数据和部件。而且,这些行业内的利益相关方应建立一种使用公共行业模型的方法,用于实现不同信息系统之间的互操作。

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

1 范围

GB/T 32392 的主要目标是规定一个支持互操作性的元模型框架。GB/T 32392 的本部分规定了一个用于注册模型之间以及 GB/T 32392.2—2015 注册对象之间映射规则的元模型。本部分为各种类型的变换规则提供了标识和分类模式,以便对其进行共享和重用。模型变换规则的描述和规范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 18391.3—2009 信息技术 元数据注册系统 第3部分:注册系统元模型和基本属性 (ISO/IEC 11179-3:2003, IDT)

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

3 术语、定义和缩略语

下列术语和定义适用于本文件。

3.1 术语和定义

3.1.1

映射 mapping

有向变换的可能实现。

3.1.2

模式 pattern

描述了被匹配对象的轮廓。

3.1.3

模式匹配 pattern matching

模型的某些部分通过某个模式进行匹配的过程。

3.1.4

查询 query

将某个模型作为输入,从该模型中选择出指定的元素。

3.1.5

关系 relation/relationship

多向变换规范和关系是不可执行的,即它们不能够创建或改变模型,但可用于检查两个模型是否一致。

3.1.6

变换 transformation

根据变换定义,从源模型自动生成目标模型的过程。

3.1.7

视图 view

从其他模型派生出的模型。

3.2 本部分使用的通用术语

3.2.1

管理项 administered item

用于记录管理库中管理信息的注册系统项。

见:GB/T 18391.3—2009。

3.2.2

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,说明在何种条件下需要这个属性。参见:可选(optional) (3.2.10)。

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。

3.2.3

元数据 metadata

定义或描述其他数据的数据。

见:GB/T 18391.3—2009。

3.2.4

元数据项 metadata item

元数据对象的实例。

注 1: 在互操作性元模型框架的所有部分中,这个术语只能够应用于 GB/T 32392.2—2015 中元模型所描述的元数据对象的实例,包括模型概念、模型领域概要和模型实例等元类的实例。

注 2: 元数据项拥有关联属性,适用于元数据项实例化的元数据对象。

见:GB/T 18391.3—2009。

3.2.5

元数据对象 metadata object

元模型定义的对象类型。

注: 在互操作性元模型框架的所有部分中,这个术语只能够应用于 GB/T 32392.2—2015 中元模型所描述的元数据对象的实例,包括模型概念、模型领域概要、模型实例等元类的实例。

3.2.6

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可以看作是一个元数据注册库。

见:GB/T 18391.3—2009。

3.2.7

元模型构件 metamodel construct

建模的记法单位。

见:GB/T 18391.3—2009。

3.2.8

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

软件制品的注册框架,这些软件制品是基于元模型和模型得到的。

3.2.9

(管理项的)名称 name(of administered item)

由特定语境指定给管理项的名称。

注:对应的元模型构件为“指定的属性”。

见:GB/T 18391.3—2009。

3.2.10

可选的 optional

允许的,但不是必须的。

注1:描述元数据项属性强制程度的三种状态之一,说明在何种条件下需要这个属性。

参见:强制的(mandatory)(3.2.2)。

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。

见:GB/T 18391.3—2009。

3.2.11

注册系统项 registry item

记录在元数据注册系统中的元数据项。

见:GB/T 18391.3—2009。

3.2.12

统一资源标识符 uniform resource identifier; URI

用于标识资源的格式化的字符串,主要用于互联网中。

注:其句法需要遵循《互联网资源定位符的功能性建议》[RFC1736]和《统一资源名称的功能性需求》[RFC1737]。

见:IETF RFC 2396。

3.3 缩略语

下列缩略语适用于本文件。

BIE	业务信息实体 (Business Information Entity)
CC	核心部件 (Core Component)
CCLib	核心部件库 (Core Component Library)
CWM	公共仓库元模型 (Common Warehouse Metamodel)
DMIM	领域消息信息模型 (Domain Message Information Model)
ebXML	电子业务用可扩展置标语言 (Electronics Business using eXtensible Markup Language)
ETL	提取、变换和装载 (Extract, Transform and Load)
HDF	HL7 开发框架 (HL7 Development Framework)
HL7	卫生七级 (Health Level Seven)
MDA	模型驱动体系结构 (Model Driven Architecture)
MOF	元对象设施 (Meta Object Facility)
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)
PIM	平台独立模型 (Platform Independent Model)

PSM	平台特定模型 (Platform Specific Model)
QVT	查询、视图和变换 (Query, View and Transformation)
RMIM	精确消息信息模型 (Refined Message Information Model)
UML	统一建模语言 (Unified Modeling Language)
UN/CEFACT	联合国贸易便捷化与电子业务标准委员会 (United Nations Centre for trade Facilitation and electronic business)
XMI	XML 元数据交换 (XML Metadata Interchange)
XPath	XML 路径语言 (XML Path Language)
XSL	可扩展样式表语言 (eXtensible Stylesheet Language)
XSLT	XSL 变换 (XSL Transformations)

4 模型映射元模型的结构

本部分是 MFI 的组成部分之一,它是基于 GB/T 32392.2—2015 制定的。

本部分为解决异构环境(由不同的软件平台和中间件组成)中可能出现的问题提供了方案。这种问题是普遍存在的,即使在包含多个相似平台的单一环境中,同一个建模产品也可能根据不同的格式或句法进行实现。在这种背景下,仍然需要建立模型映射以支持建模产品和具体数据的交换。

4.1 模型映射元模型概述

GB/T 32392.2—2015 和本部分均使用元模型来描述 MFI 元数据注册库的结构。MFI 的注册系统元模型被规定为一个概念化的抽象数据模型,即该元模型只描述了如何在现实世界中对相关信息进行组织和结构化。该元模型的任何实现都应当遵循 MFI 模型来建立元模型和相关模型的管理机制。

图 1 描述了 MFI 的模型映射包。MFI 核心模型包、遵循 MOF 的 QVT 包、遵循 MOF 的对象包、遵循 MOF 的 MDR 模型包和 MOF 包均由其他标准进行规范。

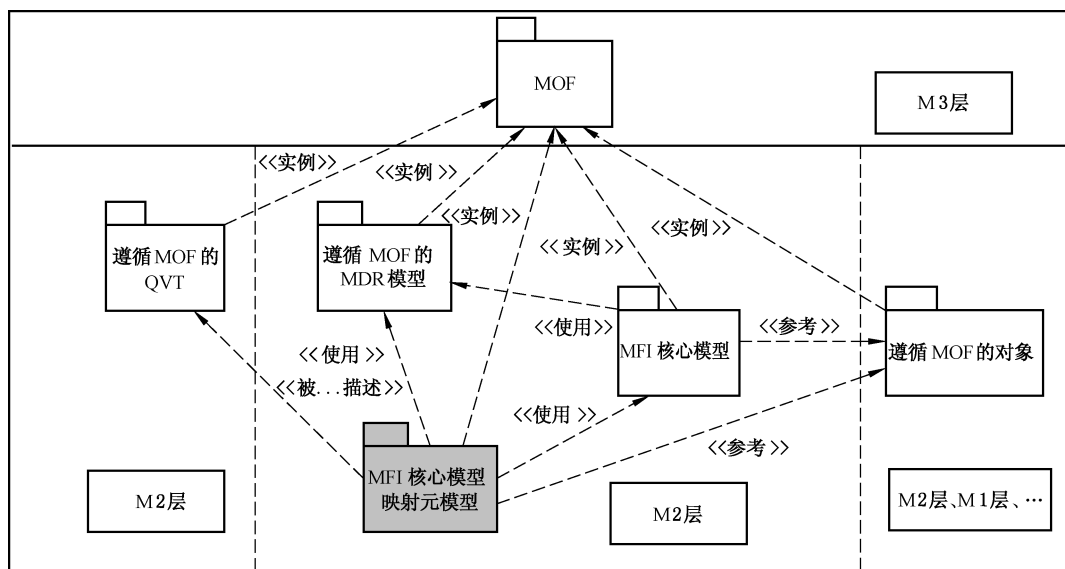


图 1 互操作性元模型框架中的包

4.2 模型映射元模型所采用的定义格式

MFI 模型映射元模型的注册系统模型使用了 MDR 中定义的管理项,并遵循 MOF 规范。它使用

术语“元模型构件”表示它所用到的模型构件,使用术语“元数据对象”表示它在 MOF 中规定的模型构件。所使用的元模型构件包括类、关系、关联类、属性和参考。这些术语的定义见 GB/T 32392.2—2015 中的 3.1,其模型描述参见 GB/T 32392.2—2015 的附录 B。本部分涉及的元数据对象将在本条中进行定义。

MFI 模型映射元模型(见图 2)可以用一组 UML 类图进行描述,每个类将按照如下形式描述:

- a) 超类
直接继承的类。
- b) 属性
属性名称:对象类型和势,强制的或可选的
描述属性的内容和用途。
- c) 参考
参考名称:类名和势,强制的或可选的
描述参考的内容和用途。
- d) 约束
如果需要,可以使用自然语言来表达约束。

模型定义了属性和参考的最小势和最大势。最大势约束在所有时间内有效,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”或更高级别的注册状态表明所有的强制属性均已记录。

本部分将从以下 6 个方面对 MFI 模型映射元模型进行描述:

- a) 上/下层规则(见 4.3.1);
- b) 元模型-模型规则(见 4.3.2);
- c) 模型-值规则(见 4.3.3);

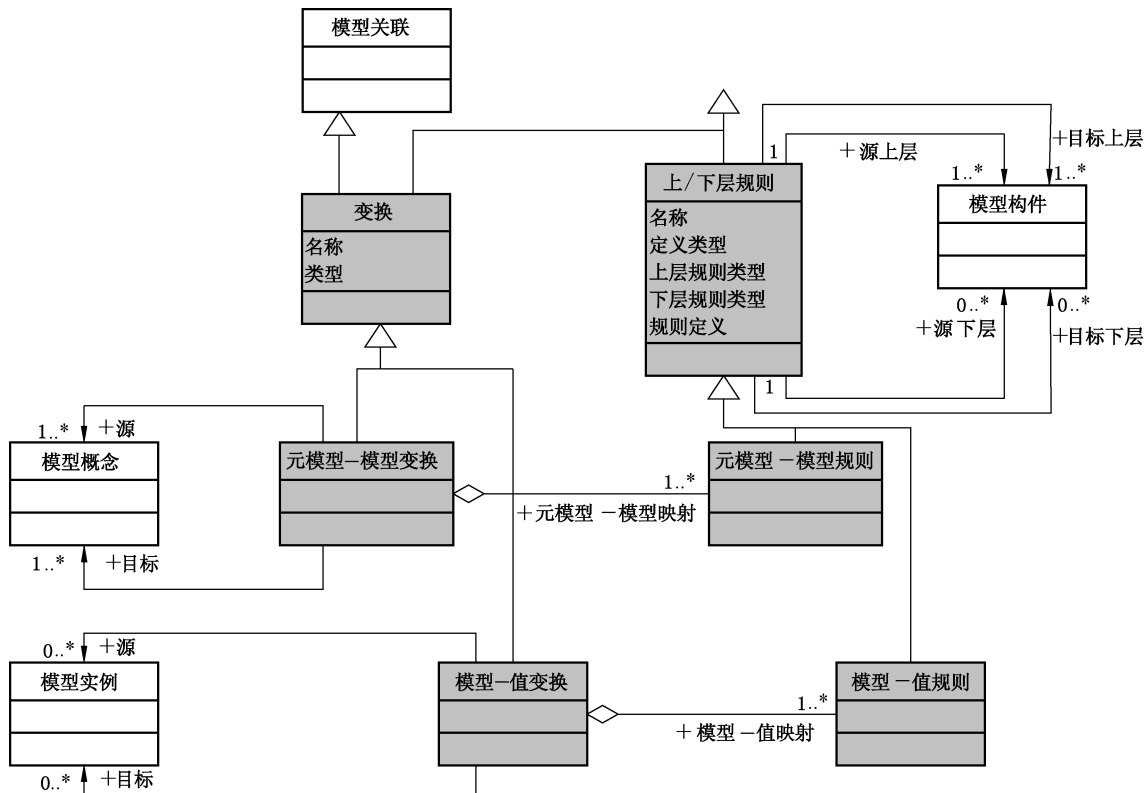


图 2 MFI 的模型映射包

- d) 变换(见 4.3.4);
- e) 元模型-模型变换(见 4.3.5);
- f) 模型-值变换(见 4.3.6)。

MF1 的模型映射示例参见附录 B,模型变换的示例参见附录 C。

4.3 模型映射元模型中的元类

4.3.1 上/下层规则

上/下层规则是一个抽象元类,指定了一组变换规则和被变换的模型部件。变换规则可用于 4 个模型部件集合,包括源上层、目标上层、源下层和目标下层。

变换是定义在上/下层源模型部件和上/下层目标模型部件之间的约束和派生。被变换的下层模型部件是基于 MOF 的对象集合,其类型只能够由已注册的包指定。变换规则是在这些模型部件的语境中定义的。

通常,变换可以有 0 个或多个方向。每个变换方向定义了一个变换的源和目标。变换执行后可以检查所有的约束是否满足,或推导出在一个选定的变换方向上的所有派生。

每个类按照如下方式进行描述:

a) 超类

管理项(源自 MF1 核心模型)

b) 属性

名称:串型[1..1],强制的

变换规则集合的标识符。

定义类型:类型代码[1..1],强制的

变换规则的语言类型可以用“QVT 关系”“QVT 核心”“QVT 可操作类型”等类型的代码来表示(见表 1 和附录 A)。

表 1 定义类型的代码集合

类型	代码	描述
QVT-关系	QVT-1	使用 MOF QVT 的声明语言进行定义
QVT-核心	QVT-2	使用 MOF QVT 的模式匹配语言进行定义
QVT-操作型的	QVT-3	使用 MOF QVT 操作映射的调用机制进行定义
QVT-黑盒	QVT-4	使用非标准的黑盒 MOF 操作实现进行定义
CWM	CWM	使用 CWM 的变换框架进行定义
XSLT	XSLT	使用 XSL 的变换语言 XSLT 进行定义

上层规则类型:类型代码[1..1],强制的

上层规则类型规定了源上层模型部件和目标上层模型部件之间的关系类型,可以使用“关系”“映射”或“计算”的代码来规定(见表 2 和表 3)。

下层规则类型:类型代码[1..1],强制的

下层规则类型规定了源下层模型部件和目标下层模型部件之间的关系类型,可以使用“关系”“映射”或“计算”的代码来规定(见表 2 和表 3)。

表 2 变换规则类型

类型	代码	源模型		目标模型		功能
		元素	实例	元素	实例	
关系	REL	应该被规定	类/关联的模式	应该被规定	类/关联的模式	约束
映射	MAP	应该被规定	类/关联的模式和对象	从源生成	类/关联的模式和对象	约束/派生
计算	COM	在执行时规定	对象(值)	从源生成	对象(值)	派生

表 3 变换操作类型

类型	操作	描述
关系	更名	将源模型部件中名称元素的名称更换为目标模型部件的名称
	等价	检查源和目标模型部件的名称元素是否相等
	增强	为目标模型部件增加元素
映射	实例化	创建源模型部件元素的实例,并将其置于目标模型部件中
	生成	编译源模型部件,并为目标模型部件发布代码
	移动	将源模型部件的一部分嵌入到目标模型部件
	派生	抽取源模型部件的元素,并且以这些元素为基础创建目标模型部件
计算	聚合	为源模型部件的元素值实施算术和逻辑操作,并且将结果与目标模型部件的值合并
	转换	将源模型部件的数据类型值转换为合适的目标模型部件数据类型值

规则定义:URI[1..1],强制的

变换规则定义文件的统一资源标识符。

c) 参考

源上层:模型构件[1..*],强制的

被变换的源上层模型部件包。

目标上层:模型构件[1..*],强制的

被变换的目标上层模型部件包。

源下层:模型构件[0..*],强制的

被变换的源下层模型部件包。

目标下层:模型构件[0..*],强制的

被变换的目标下层模型部件包。

d) 约束

源上层可以处于元模型层或模型层,源下层可以处于模型层或值层。

目标上层可以处于元模型层或模型层,目标下层可以处于模型层或值层。

4.3.2 元模型-模型规则

元模型-模型规则是上/下层规则的子类,用于指定源模型部件和目标模型构件之间的元模型变换规则,提供了元模型层和模型层之间的形式化变换规则。

每个类按照如下方式进行描述:

a) 超类

上/下层规则。

b) 属性

定义类型:类型代码[1..1],强制的
(源自上/下层规则)

上层规则类型:类型代码[1..1],强制的
(源自上/下层规则)

上层规则类型代码应该是“关系”。

下层规则类型:类型代码[1..1],强制的

下层规则类型代码可以是“映射”或“关系”。

规则定义:URI[1..1],强制的
(源自上/下层规则)

c) 参考

源上层:模型部件[1..*],强制的
(源自上/下层规则)

目标上层:模型部件[1..*],强制的
(源自上/下层规则)

源下层:模型部件[0..*],强制的
(源自上/下层规则)

目标下层:模型部件[0..*],强制的
(源自上/下层规则)

d) 约束

源上层必须处于元模型层,源下层必须处于模型层。

目标上层必须处于元模型层,目标下层必须处于模型层。

4.3.3 模型-值规则

模型-值规则是上/下层规则的子类,指定了定义在源模型部件和目标模型构件之间的元模型变换规则,提供了模型层和值层之间的形式化变换规则。

每个类按照如下方式进行描述:

a) 超类

上/下层规则。

b) 属性

定义类型:类型代码[1..1],强制的
(源自上/下层规则)

上层规则类型:类型代码[1..1],强制的
(源自上/下层规则)

上层规则类型可以是“映射”或“关系”的代码。

下层规则类型:类型代码[1..1],强制的

下层规则类型应该是“计算”的代码。

规则定义:URI[1..1],强制的
(源自上/下层规则)

c) 参考

源上层:模型部件[1..*],强制的
(源自上/下层规则)

目标上层:模型部件[1..*],强制的
(源自上/下层规则)

源下层:模型部件[0..*],强制的
(源自上/下层规则)

目标下层:模型部件[0..*],强制的
(源自上/下层规则)

d) 约束

源上层必须位于模型层,源下层必须位于值层。

目标上层必须位于模型层,目标下层必须位于值层。

4.3.4 变换

变换是管理项和模型关联的子类,指定了定义在源模型概念(或模型实例)和目标模型概念(或模型实例)之间的变换。

映射是变换的一部分,可以应用于 0 或多个模型概念(或模型实例)。映射的每一个模型概念(或模型实例)均处于变换的某个方向中(见图 3 和图 4)。

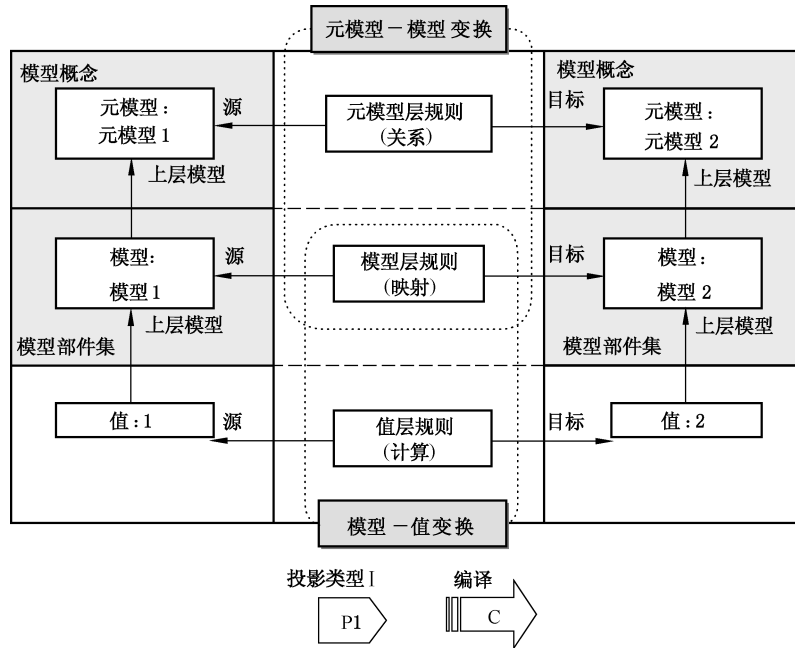


图 3 变换合并(1)

当用户执行变换得到模型时,该变换中的所有映射都会执行,从而得到选定变换方向中的模型概念(或模型实例)。

每个类按照如下方式进行描述:

a) 超类

管理项,模型关联。

b) 属性

名称:串型[1..1],强制的

模型概念(或模型实例)之间变换的标识符。

类型:类型代码[1..1],强制的

变换的类型,可以用“编译类型”“投影类型 I”“投影类型 II”或“变换类型”等代码表示(见表 4)。

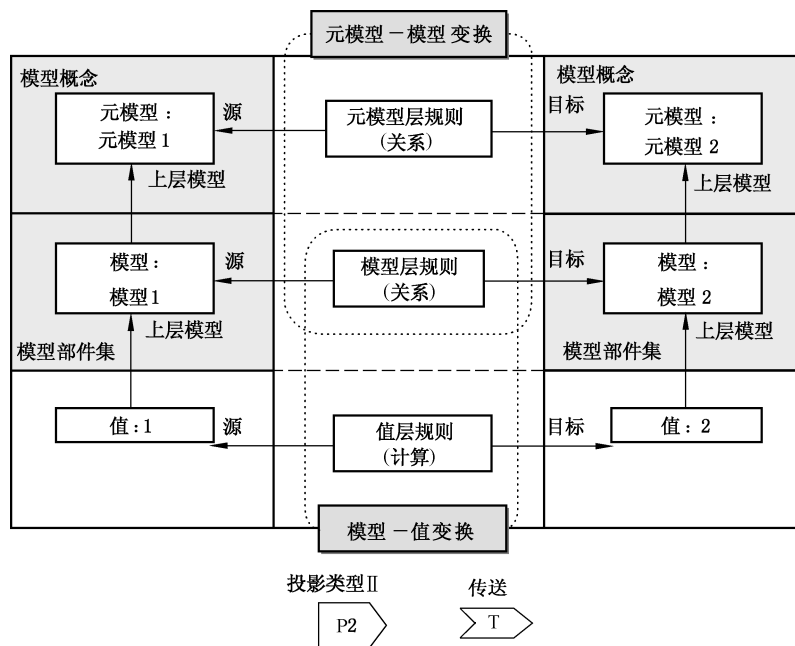


图 4 变换合并(2)

表 4 变换类型

种类	代码	描述
编译类型	C	通过生成,把源编译为目标
投影类型 I	P1	通过生成,把源的子集投影为目标
投影类型 II	P2	通过匹配,把源的子集投影为目标
变换类型	T	通过匹配,将修改的源变换为目标

c) 参考

无。

d) 约束

无。

4.3.5 元模型-模型变换

元模型-模型变换是变换的子类,指定了定义在源模型部件和目标模型部件之间的模型变换。每个类按照如下方式进行描述:

a) 超类

变换。

b) 属性

名称:串型[1..1],强制的

(源自变换)

类型:类型代码[1..1],强制的

(源自变换)

元模型-模型映射:元模型-模型规则[1..*],强制的

在元模型层和模型层中,源模型部件和目标模型部件之间的映射规则。

c) 参考

源:模型概念[1..*],强制的

源模型部件的模型概念,位于元模型层。

目标:模型概念[1..*],强制的

目标模型部件的模型概念,位于元模型层。

d) 约束

元模型-模型映射的上层规则类型(元模型层)和下层规则类型(模型层)应该使用表 5 中提供的代码。

表 5 元模型-模型变换组合的类型

种类	代码	元模型层	模型层	值层
编译类型	C	关系	映射	无
投影类型 I	P1	关系	映射	无
投影类型 II	P2	关系	关系	无
变换类型	T	关系	关系	无

4.3.6 模型-值变换

模型-值变换是变换的子类,指定了定义在源值和目标值之间的变换。

每个类按照如下方式进行描述:

a) 超类

变换。

b) 属性

名称:串型 [1..1],强制的

(源自 变换)

类型:类型代码[1..1],强制的

(源自 变换)

模型-值映射:模型-值规则[1..*],强制的

在模型层和值层中,源模型部件和目标模型部件之间的映射规则。

c) 参考

源:模型实例[0..*],强制的

每个源模型部件的模型实例,位于模型层。

目标:模型实例[0..*],强制的

每个目标模型部件的模型实例,位于模型层。

d) 约束

模型-值映射的上层规则类型(模型层)和下层规则类型(值层)应该使用表 6 中提供的代码。

4.4 交换模型的标准格式

本部分没有规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。

其他元数据格式也是允许的。

表 6 模型-值变换组合的类型

种类	代码	元模型层	模型层	值层
编译类型	C	无	映射	计算
投影类型 I	P1	无	映射	计算
投影类型 II	P2	无	关系	计算
变换类型	T	无	关系	计算

5 符合性

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,应保证元模型和实现之间的双向映射是无二义性的。

符合的实现应该:

- a) 满足 4.3 的要求;
- b) 标识符合性程度(5.1);
- c) 标识符合性级别(5.2)。

5.1 符合性的程度

MF1 模型映射元模型的符合性从 3 个方面进行规定:第一是值的要求;第二是元数据交换的互操作性;第三是注册内容(如上层模型和下层模型)之间的符合。

- a) MF1 模型映射元模型的符合在表 7 中规定;
- b) 交换格式的符合在表 7 中规定;
- c) 注册内容的符合在本部分中未规定。

“级别 1”“级别 2”实现之间的区别对于同时解决互操作性和扩展性的需要是必要的。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 7。

表 7 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	不应该支持使用、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	不应该承认、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。

5.2 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 8 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

5.3 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。

5.4 实现符合性的声明

声明符合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(5.1);
- b) 符合性级别 1 或级别 2(5.2),或二者皆是;
- c) 支持或使用哪些扩展。

5.5 注册的角色和责任

符合需要在注册机构的角色和责任中考虑,包含在 GB/T 18391.6—2009 中。系统的扩展符合性需要规程的形式化,各部分之间的角色和职责达成一致,处理软件产品的使用和从其他系统转换的指南。这些方面的形式化应和上述条款中符合性要求一致,也应与 GB/T 18391.6—2009 中注册机构的角色陈述一致。

附录 A (资料性附录) 变换语言

使用模型模式进行模型变换是实现模型之间互操作的有用技术。模型变换可以利用模型模式实现参数化。变换定义同样应该采用形式化的变换定义语言进行描述。模型编译器是一个应用程序,它读取一种用形式化语言书写的变换定义,并将其作为变换的源模型,然后将其变换为另一个等价的元模型,即目标模型。

基于 MOF 的工具可以用于规定和实现变换规则。这类支持工具需要使用特定的变换描述语言。变换定义工具为变换定义的创建和修改提供了支持。一般来说,需要使用一种形式化的变换描述语言来定义复杂的变换规则,并在不同工具间实现变换的共享。

例如,OMG 定义的 MOF QVT 是一种形式化的变换语言,它可以通过精确有效的方式规定变换规则。变换引擎(如模型编译器)能够支持从模型生成简单的代码,并为模型驱动体系结构提供支持。

变换语言的种类:

a) QVT-关系

MOF 模型之间关系的声明式规范。“关系”语言支持复杂的对象模式匹配,并且隐式地创建了跟踪类及其实例,用于记录在变换执行过程中所发生的事情。

b) QVT-核心

一种规模较小的模型语言,仅能够通过评估变元在模型中的条件来支持变元集合的模式匹配。它对称性地处理源模型、目标模型和跟踪模型中的所有模型元素。另外,跟踪模型应被显式地定义,且这些跟踪不能够像“关系”那样从变换定义中推导出来。核心模型可以被直接实现;或者作为“关系”语义的参考,其中“关系”需要使用变换语言映射到核心模型中。

c) QVT-可操作类型,QVT-黑盒

QVT-关系和 QVT-核心语言在两个不同的抽象级别体现了相同的语义。除此之外,还有两种机制可以从 QVT-关系和 QVT-核心中调用变换的命令式实现,即“操作映射”和非标准的黑盒 MOF 操作。每个关系定义了一个类,可以实例化这个类来跟踪被变换的模型元素;同时,它可以一对一映射到 QVT-可操作映射或 QVT-黑盒的操作标识。

d) CWM

基于 OMG CWM 定义的变换框架需要利用 XSLT 实现变换。CWM 变换框架提供了源和目标元素的连接机制,但是目标元素的推导必须使用某个具体语言来实现,而这个语言并未在 CWM 中规定。也就是说,CWM 只提供了一个通用的模型,没有提供实际的机制来实现模型变换。

e) XSLT

XSLT 是一种模型变换语言。由于 MOF 模型可以通过 XML 元数据交换(XMI)序列化为 XML 文件,进而可以利用 XSLT 实现模型变换。

附录 B
(资料性附录)

MFI 注册系统和模型映射

特定的图示语言(如 UML)已经广泛应用于信息技术的各个领域。源模型和目标模型都同样是多样化的,一个目标模型可以是其他的模型或编程语言,或介于微处理器和超级计算机之间的机器语言。

MFI 注册系统和模型映射的用例很多。

图 B.1 描述了一个医疗保健应用的示例。MFI 注册系统与存储库(MFI R&R)中存放了已注册的模型部件和模型变换,包括表示模型映射的最小元模型集合。如图 B.1 所示,需要实现不同系统之间的模型映射。

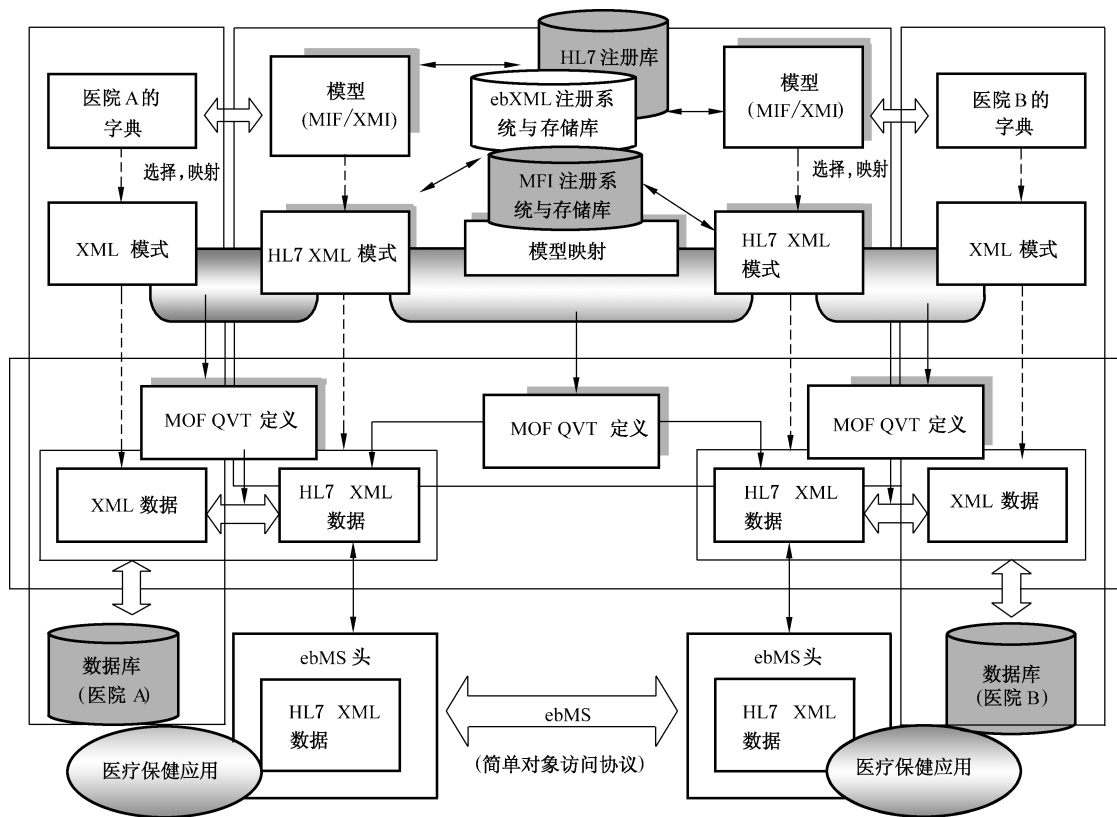


图 B.1 MFI 注册系统和模型映射的环境

附录 C
(资料性附录)
模型变换示例

本附录列举的模型变换包括以下几个示例：

- a) 示例 1:HL7 中的变换(见图 C.1)；
- b) 示例 2:ebXML 中的变换(见图 C.1)；
- c) 示例 3 和示例 4:MDA 中的变换(见图 C.2)；
- d) 示例 5 和示例 6:CWM 和 XSLT 之间的变换(见图 C.3)。

C.1 示例 1

HL7 开发框架提供了从 DMIM 到 RMIM 的模型变换。

C.2 示例 2

构建消息模型的另一种方法是使用 ebXML,它是由 OASIS 和 UN/CEFACT 开发的。比如,从 CC 到基于 CCLib 元模型的 BIE 之间的映射是一个典型的模型映射示例。

C.3 示例 3 和示例 4

示例 3 和示例 4 是基于 MDA 的应用场景。MDA 提供了从高层抽象的上层模型到多个实现模型的变换功能。在 MDA 中,从 PIM 到 PSM 的变换是一个典型的变换示例。

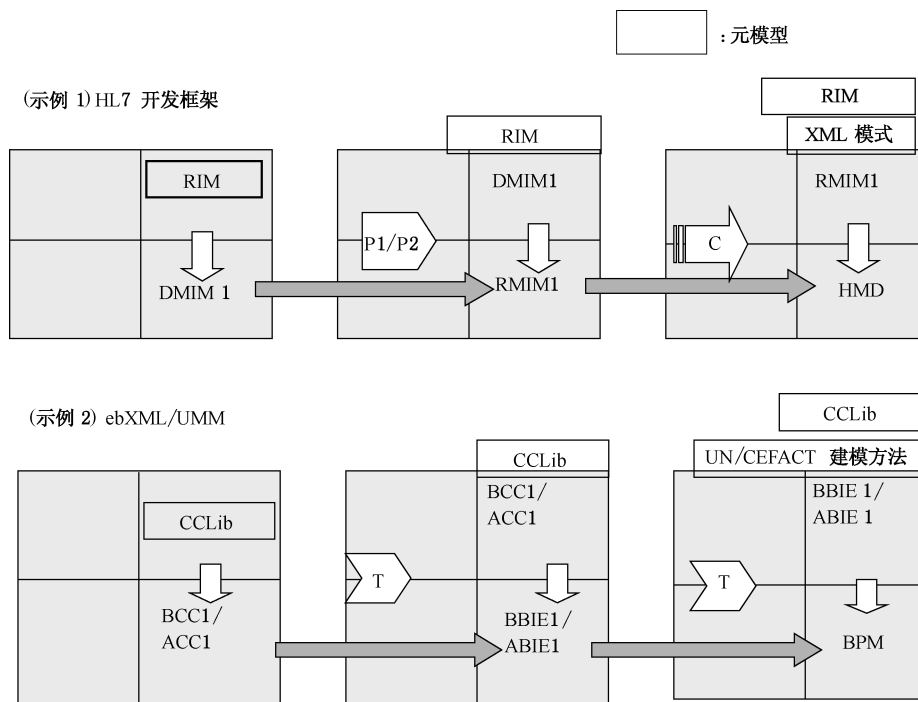
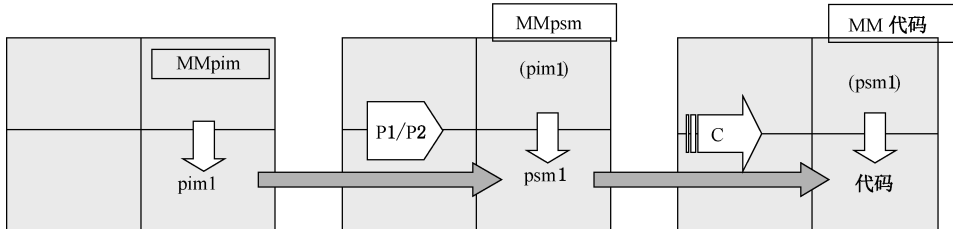


图 C.1 HL7 和 ebXML 中的变换

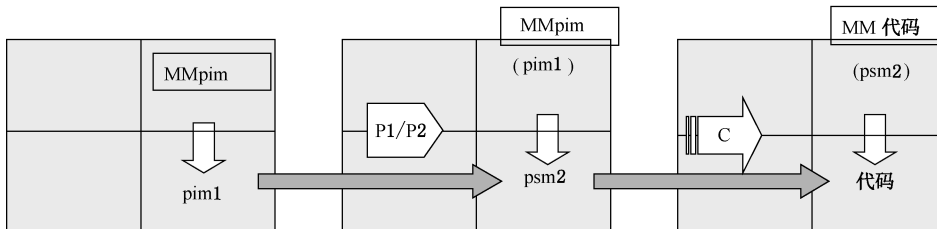
C.4 示例 5 和示例 6

示例 5 和示例 6 描述了 CWM 中不同元模型之间的模型变换。例如,ETL 提供了基于结构化查询语言(SQL)的表之间的模型变换。同样,与 XSL、XPath、XSLT 相关的变换也是这类变换的典型应用。

(示例 3) 模型驱动体系结构(从 PIM 到 PSM)



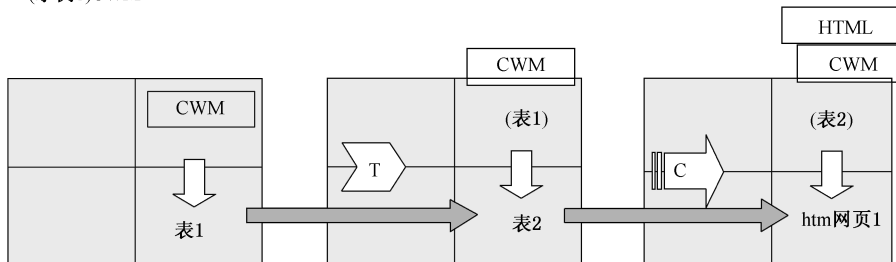
(示例 4) 模型驱动体系结构(从 PIM 到 PIM)



注: 本图中大写英文表示对应规范的名称,小写英文表示遵循这些规范得到的实例。

图 C.2 MDA 中的变换

(示例5)CWM



(示例6)XSL, XPATH, XSLT

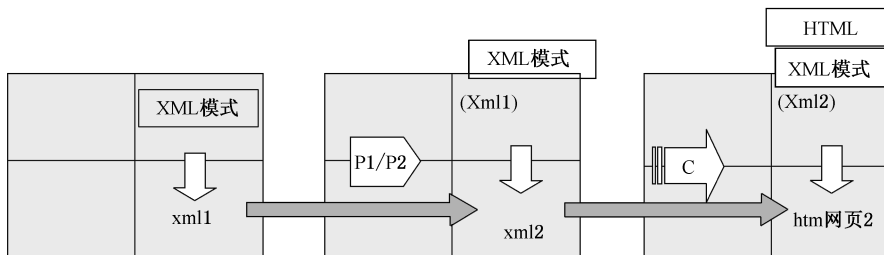


图 C.3 CWM 和 XSLT 间的变换

参 考 文 献

[1] ISO/IEC TR 9007:1987, Information processing systems—Concepts and terminology for the conceptual schema and the information base.

TR 9007 provides information on conceptual modelling.

[2] ISO/IEC 10027:1990, Information technology—Information Resource Dictionary System (IRDS) Framework.

ISO/IEC 10027 describes the concept of levels of modelling.

[3] ISO/IEC 20944, Information Technology—Metadata Registry Interoperability and Bindings (MDRIB)

[4] Object Management Group, MOF 2.0 Query / Views / Transformations RFP, OMG Document: ad/2002-04-10, revised on April 24, 2002.

[5] Adaptive, Alcatel, UMTF, Compuware, DSTC, IBM, Interactive Objects, Kennedy Carter, Softeam, Sun Microsystems, Tata Consultancy Services, Thales, TNIVALIOSYS, MOF 2.0 Query/View/Transformation RFP (ad/2002-04-10) QVT-Merge Group version 2.1 ad/2005-07-01.

[6] QVT-Partners. MOF Query/Views/Transformations, Revised Submission. OMG Document: ad/2003-08-08.

[7] CBOP, DSTC, and IBM, MOF Query/Views/Transformations, Revised Submission. OMG Document: ad/03-08-03.

[8] OMG, The Common Warehouse Model 1.1., OMG Document: formal/2003-02-03.

[9] W3C, XSL Transformations (XSLT) Version 1.0, November 1999, <http://www.w3.org/TR/xslt>.
