

中华人民共和国国家标准

GB/T 18975.2—2008/ISO 15926-2:2003

工业自动化系统与集成 流程工厂(包括 石油和天然气生产设施)生命周期 数据集成 第2部分:数据模型

Industrial automation systems and integration—Integration of life-cycle
data for process plants including oil and gas production facilities—
Part 2: Data model

(ISO 15926-2:2003, IDT)

2008-10-07 发布

2009-04-01 实施

中华人民共和国国家质量监督检验检疫总局 发布
中国国家标准化管理委员会

目 次

前言	III
1 范围	1
2 规范性引用文件	1
3 术语、定义和缩略语	2
3.1 术语和定义	2
3.2 缩略语	3
4 基本概念和假设	3
4.1 概念数据模型	3
4.2 数据模型设计	3
4.3 系统标识符	4
4.4 记录管理信息	4
4.5 文档约定	4
4.6 数据模型概念	6
4.7 可能个体	7
4.8 类	25
4.9 功能映射	68
4.10 其他用户定义关系	71
5 生命周期集成模式	75
5.1 概述	75
5.2 模式定义	76
附录 A(规范性附录) 信息对象注册	164
附录 B(资料性附录) 计算机可解释的列表	165
附录 C(资料性附录) GB/T 16656.11 EXPRESS 应用	166
附录 D(资料性附录) GB/T 18975 中集合理论的一些注	167
附录 E(资料性附录) 分析 association 的应用和含义	170
参考文献	176

前 言

GB/T 18975《工业自动化系统与集成 流程工厂(包括石油和天然气生产设施)生命周期数据集成》有下列部分:

——第1部分:综述与基本原理;

——第2部分:数据模型。

本部分是GB/T 18975的第2部分。本部分等同采用ISO 15926-2:2003《工业自动化系统与集成 流程工厂(包括石油和天然气生产设施)生命周期数据集成 第2部分:数据模型》。

本部分在技术内容和编写格式上与ISO 15926-2:2003保持一致,根据GB/T 1.1—2000,作了如下编辑性改动,主要是:

- a) 为了维护英文原义又便于了解名称代表的意义,对带下划线用EXPRESS语言描述的各黑体英文实体名,在本部分作为标题出现时以中文译名标出,在正文中以英文为主标出,在第一次出现或必要时,将中文译名括起来放在英文原名之后。
- b) 国际标准ISO 15926的各部分被修改采用为我国的国家标准时,其编号是GB/T 18975.×,对应的各部分在技术和使用上对等。但是考虑到与国家标准GB/T 18975相配套的EXPRESS描述,以及应用软件中各模式、实体、特性、属性、函数等的表达需要,为了使配套应用软件在实际应用时,不发生因标准转化所带来的种种问题,本部分的所有EXPRESS描述以及由STEP开发工具自动生成的文件和EXPRESS-G图中的国际标准编号仍保持不变。

本部分的附录A为规范性附录;附录B、附录C、附录D、附录E为资料性附录。

本部分由中国机械工业联合会提出。

本部分由全国工业自动化系统与集成标准化技术委员会(SAC/T 159)归口。

本部分起草单位:中国标准化研究院。

本部分主要起草人:李文武、刘守华、何涛、詹俊峰。

工业自动化系统与集成 流程工厂(包括 石油和天然气生产设施)生命周期 数据集成 第2部分:数据模型

1 范围

为了用计算机表达流程工厂的技术信息,GB/T 18975 的本部分规定了一种概念性数据模型。

本部分适用于:

- a) 规定生产、加工和运输加工原料的技术条件。
- b) 规定生产和加工所需原料的必要设施。包括以下部分:
 - 碳氢化合物加工与环境系统;
 - 已注入气和水的环境以及注射系统;
 - 石油与天然气产品输送系统;
 - 安全与控制系统;
 - 发电与供电系统;
 - 蒸汽产生与供给系统;
 - 结构;
 - 厂房和设施。
- c) 为提供必需的生产和加工功能,对原料和设备进行描述和选择,包括有关市场可用的原料和设备、工厂设备的安装和投料试运转信息。
- d) 生产和加工运转,包括生产条件及加工原料的消耗量、产量和质量。
- e) 设备的维修和更换。

本部分不适用于:

——厂房、生产设备和设施的建造。

2 规范性引用文件

下列文件中的条款通过 GB/T 18975 的本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

GB/T 7408 数据元和交换格式 信息交换 日期和时间表示法(GB/T 7408—2005,ISO 8601:2000,IDT)

GB/T 16262.1 信息技术 抽象语法记法一(ASN.1) 第1部分:基本记法规则(GB/T 16262.1—2006,ISO/IEC 8824-1:2002,IDT)

GB/T 16656.1 工业自动化系统与集成 产品数据表达与交换 第1部分:概述与基本原理(GB/T 16656.1—2008,ISO 10303-1:1994,MOD)

GB/T 16656.11 工业自动化系统与集成 产品数据的表达与交换 第11部分 描述方法:EXPRESS语言参考手册(eqv GB/T 16656.11—1996,ISO/DIS 10303-11:1993)

GB/T 18975.1 工业自动化系统与集成 流程工厂(包括石油和天然气生产设施)生命周期数据集成 第1部分:综述与基本原理(GB/T 18975.1—2003,ISO/DIS 15926-1:2001,MOD)

3 术语、定义和缩略语

3.1 术语和定义

为方便本部分的描述,下列术语和定义与所引用标准中的术语和定义重复。

注:下面列出的定义完全出自于括号中所列标准中的定义,例如[GB/T 16656.1]。引用标准中的定义是规范性的,这里的重复是资料性的,这里的术语如果与引用文件中的术语相矛盾则以引用文件为准。

3.1.1

类 class

为了包括和不包括,基于一个或多个准则的事物种类或划分。

注:类不必有任何成员(满足其成员准则的事物)。

[GB/T 18975.1]

3.1.2

概念性数据模型 conceptual data model

ISO/TR 9007^[2]定义的三层模式体系结构中的数据模型,其数据结构的表达形式与任何物理存储或外部表达格式表达无关。

注:根据 IDEF1X 规范^[6]改写。

[GB/T 18975.1]

3.1.3

数据 data

一种形式化的信息表达,它适合于人或计算机进行通信、解释或处理。

[GB/T 16656.1]

3.1.4

数据存储器 data store

为了将来的引用,允许存储数据的计算机系统。

[GB/T 18975.1]

3.1.5

数据仓库 data warehouse

为了提供集成的、无重复信息或冗余信息的数据集,把相关数据合并后保存在其中并支持许多不同应用视图的数据存储器。

[GB/T 18975.1]

3.1.6

个体 individual

在空间和时间上存在的事物。

注1:在这里,事物是在某些一致的逻辑范围内可以想象的,包括实际的、假设的、计划的、期望的或需要的个体。

示例:系列号 ABC123 的泵、Battersea 发电站、Joseph Whitworth 先生和恒星飞船“Enterprise”都是个体的例子。

注2:有关个体概念的详细讨论参考本部分的 4.7。

[GB/T 18975.1]

3.1.7

信息 information

事实、概念或指令。

[GB/T 16656.1]

3.1.8

流程工厂生命周期数据 process plant life-cycle data

用计算机可处理的形式表达一个或多个流程工厂整个生命周期中的信息数据。包括规划、设计、施工、运转、维护、停产和拆除。

[GB/T 18975.1]

3.1.9

参考数据 reference data

表达许多流程工厂所公用或许多用户所关心的类或个体信息的流程工厂生命周期数据。

[GB/T 18975.1]

3.2 缩略语

下列缩略语适用于本部分。

UTC 协调世界时,又称为世界标准时间(原子时钟):Coordinated Universal Time

4 基本概念和假设

4.1 概念数据模型

第5章规定的模型是一个概念数据模型,图1显示了它与内部模型和外部模型之间的关系(参见 GB/T 18975.1—2003 的 5.2)

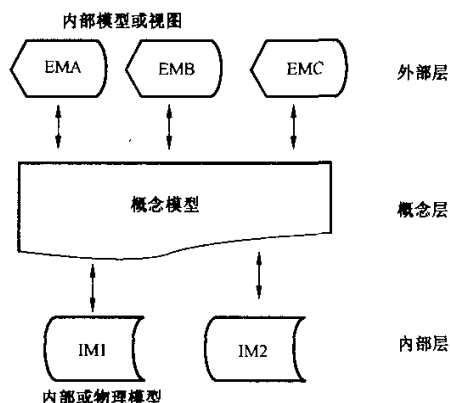


图1 三级结构

注1: 概念数据模型的术语定义在 GB/T 18975.1 中,并且是以 ISO/TR 9007^[2]中描述的三种模式结构为基础。

为了能使流程工厂生命周期中的信息集成起来,本模型不包含那些只适合其中某些特定应用的信息约束。

注2: 数据集成意味着要把来自各自不同数据源的信息组合起来,形成能表达已知含义的一致数据集。由于不同数据源的范围之间有重叠,组合数据时需要识别其共性,重复的信息要去掉,并且新的信息也要表示出来。为了成功的集成,数据模型必须有包含所有可能或需要数据的结构。

4.2 数据模型设计

本数据模型在设计时与 EPISTLE(欧洲流程工业 STEP 技术联络执委会)开发的数据建模原则保持一致。在定义概念模型时,这些原则控制了实体数据类型、属性和关系的使用。

这些原则有以下作用:

- 模型实体数据类型是实体数据类型的整个子类型/超类型层次结构的一部分。
- 实体数据类型是一般化的,表达了其成员稳定存在的性质,并由此命名。
- 属性信息通常是用实体数据类型的引用来表示。
- 表示数字、字符和二进制模式的属性都用 EXPRESS 简单类型定义的。
- 关系和时间是用实体数据类型来表示的。

4.3 系统标识符

数据模型中为每一个实体实例定义了一个人工的系统标识符——thing.id(事物标识)。每一个这样的标识符都是 EXPRESS 字符串数据类型的成员,并且在给定系统的标识符集合中被限定是唯一的。系统标识符是强制的。对给定的系统中的每个实体实例,其标识符在其整个生命周期中都应当是一致的。

示例:在一个管理海上平台设计和工程信息的数据库系统中,系统的标识符就应当在其所涉及的范围内是唯一的。在系统中的每个系统标识符,应当被解释成对其所标识的 thing(事物)的引用。

其他用于特定 thing 的系统外部标识符,用 class_of_identification(标识类)实体数据类型来记录。

4.4 记录管理信息

数据模型为计算机中表示流程工厂模型信息的记录提供了受限持有的信息。这些记录管理数据可以用为每个实体实例给出的属性值来描述。这些信息包括以下内容:

- 对产生自其他系统的记录,它被拷贝到当前系统时的日期和时间。
- 该记录在其原始系统中被初次创建的日期和时间。
- 该记录在其原始系统中被初次创建时的创建人、组织和系统。
- 该记录被逻辑删除的日期和时间。
- 对被逻辑删除的记录,该记录被删除的原因,逻辑删除是指当记录在系统中以历史记录的形式存在时,它就总会被视为一个非法语句。

5.2.1.2 中给出了记录管理属性的定义。

4.5 文档约定

4.5.1 实体和属性定义

4.6 给出了数据模型的概念,第 5 章中给出形式化的描述。4.6 描述了模型概念及其应用的例子。4.6 中的描述补充了第 5 章的定义和实例,以及模型的精确描述。

4.5.2 图表

本章中用到了大量的图表来描述基本概念和假设,使用了以下三类图:

- 时空图;
- 以 EXPRESS-G 惯例的子集为基础的模型图;
- 实例图。

4.5.2.1 时空图

在用时空外延对实际时空中存在的具体切实对象进行建模时,常用图 2 来进行说明。

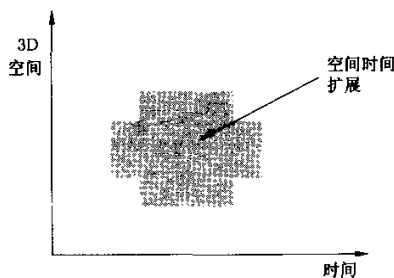


图 2 时空图

时空图中有两条垂直的轴线,纵轴表示三维空间,横轴表示时间。时空外延则是用图上带边界的或阴影的区域来表示。外延的左边界代表着外延的起始,右边界代表着外延的结束。上下边界之间的变化表示了空间外延的变化。

4.5.2.2 模型图

模型图是用来说明模型的各部分,使其能被更好地解释和理解。模型图只限于表示:EXPRESS 实体类型,EXPRESS 子类和 EXPRESS 关系,并使用了合适的 EXPRESS-G 符号。

示例：图 3 是一个模型图的例子

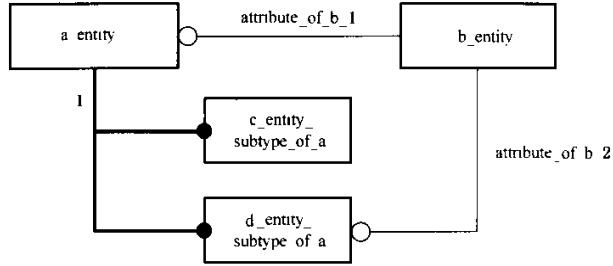


图 3 模型图

第 5 章中给出了这个模型的完整 EXPRESS-G 图集。

4.5.2.3 实例图

实例图是用来说明模型实体类型含义的。图 4 中描述了构成实例图时所用的一些符号。

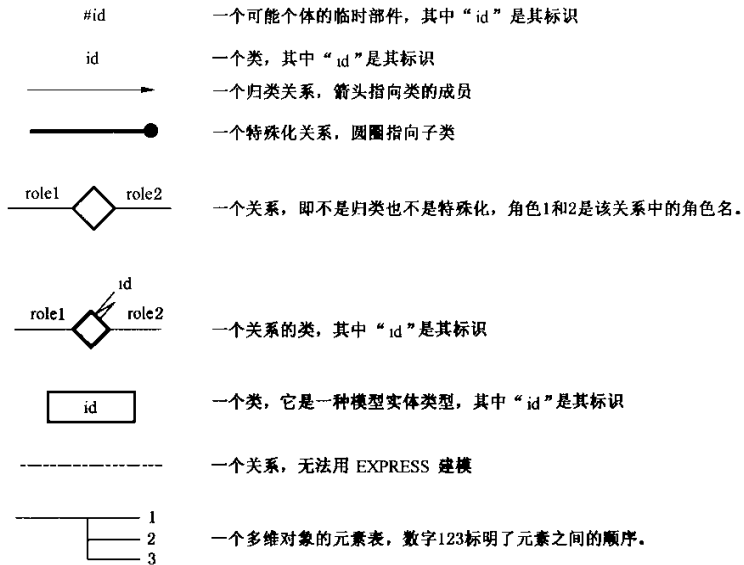


图 4 实例图注解

图 5 是一个实例图的例子。其中 #1234 所标识的对象是 thing 和泵的成员。“泵”是 class(类)的成员。thing 和 class 都是模型中的实体类型。

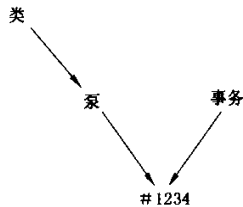


图 5 例子图注解

注 1：图 4 中的符号用于表示本部分中 thing, 实体数据类型, classification(分类), specialization(特殊化), relationship(关系)和 class_of_relationship(关系类)的例子。

注 2：在本部分中，第 5 章中定义的模型实体数据类型被认为是类。这些类在图表中说明时，都会带方框。

注 3：在本章中，如果实例图中的对象是实体数据类型或其直接、间接成员，则认为该实例图是完整的。

注 4：本章中的实例图不是为了标明模型在实际中将如何被实例化的。特别是大多数特定模型实体类型的成员关系没有表示出来，一些实体类型的成员关系也可能被忽略了。

4.6 数据模型概念

4.6.1 事物

数据模型由实体类型的子类型/超类型层次结构组成。

一个作为根的超类型 thing,它是代表任何事物的类。

thing 可划分成:

—— possible_individual(可能个体),或

—— abstract_objects(抽象对象)。

图 6 中显示了此层次结构最顶端的划分(见 5.2.1 和图 177 中)。

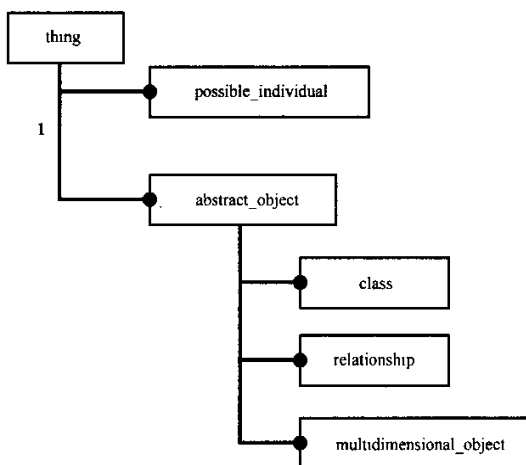


图 6 模型子类型/超类型层次结构的一部分

4.6.2 可能的个体

possible_individual 是时空中存在的 thing(见图 6 和 5.2.6)。一个 possible_individual 的特征是它的时空外延。任何两个个体的时空外延都不会完全相同。每天的现实事物,也就是人们常说的具体对象,都是 possible_individual。

示例: #1234 号泵就是一个 possible_individual。相反一个 abstract_objects 是时空中不存在的 thing。class、relationship 和 multidimensional_object(多维对象)都是 abstract_objects。

4.6.3 类

class 是对 thing 性质的理解,按照一个或多个条件(见 5.2.2 和图 178),把 thing 划分成属于 class 的 thing 和不属于 class 的 thing。class 的特征是它的成员关系。任何两个 class 的成员关系不会完全相同。

示例 1: 我们已知的泵的概念就是一个 class。

class 是全局的,没有时空外延。然而,class 可以包含时间、空间作为条件。

示例 2: “六月的销售”就是一个 class。

4.6.4 关系

两个 thing 之间的 relationship(见 5.2.11 和图 187)。在本部分中,relationship 被定义成有序对的分类。其上重复记录了另一个 relationship。任何两个 relationship 类的有序对都不会完全相同。可以按序给相关联的 thing 分配角色。

示例 1: “泵”和“#1234”之间的有序对表示了 classification 关系,其中“泵”是 class,“#1234”是成员。

本部分定义了 relationship 的显式子类型,覆盖了流程行业中的一些通常使用的关系。

示例 2: relationship 的显式子类型有 classification、specialization、lifecycle_stage(生命周期阶段)和 approval(批准)。

4.6.5 多维对象

一个 multidimensional_object 是一个有序的 thing 列表(见 5.2.4 和图 180)。列表中的 thing 可以

是 possible_individual, class, relationship 或其他 multidimensional_object。

示例：表[2.0, 4.0, 5.7]是一个 multidimensional_object。

multidimensional_object 元素的次序是用 EXPRESS LIST 聚集类型定义的。

4.7 可能个体

一个 possible_individual 是时空中存在的 thing,通常指一个具体的对象(见 5.2.6 和图 182)。

possible_individual 所包括的 thing 有：

- 实际存在的或过去已经存在的；
- 过去和将来可能存在的；
- 假设的,过去和将来都不存在。

在本部分中,一个 possible_individual 相对应于一个特定的时空外延。如两事物的时空外延完全相同,则认为它们是同一事物。

图 7 中的 possible_individual“#1234”的时空图说明了 possible_individual 的普遍性质。

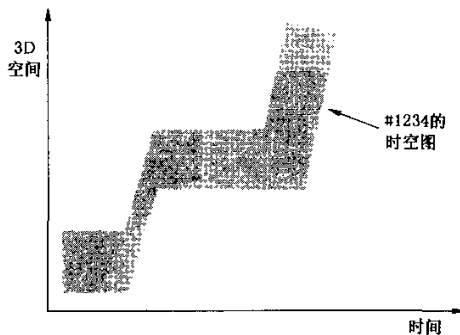


图 7 可能个体的时空范围

图 7 中时空外延的实例图见图 8。

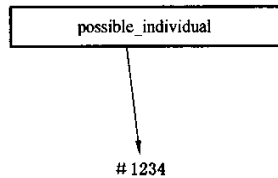


图 8 可能个体 #1234 的实例图

注：本部分中“可能个体”和“个体”都是用于时空外延的术语。

4.7.1 可能个体的组合

一个 possible_individual 可能是另一个 possible_individual 的一部分。组合或整体/部分的形态,与 possible_individual 是不同的 class。整体/部分用 composition_of_individual(个体成分)表示,它是 relationship 的一个子类型(见 5.2.6.5 和图 182),如图 9 所示。

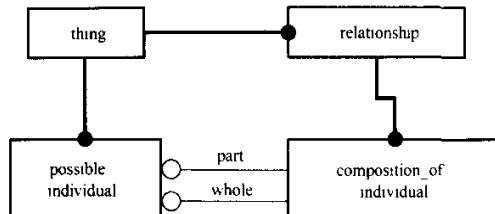


图 9 个体关系的组合

示例：设想一台离心泵的叶轮安装到泵上的时间是泵和叶轮时间外延的一部分,如图 10 所示。外延 #1234 和 #5678 分别代表叶轮和泵,其交集 #9012 既是叶轮的一部分也是泵的一部分。

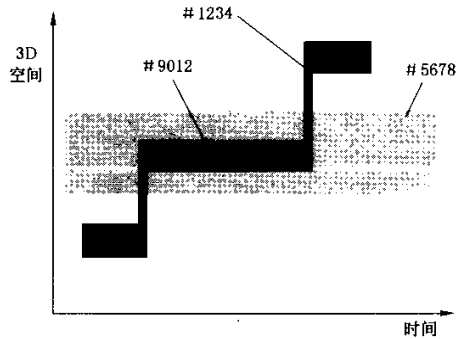


图 10 时空外延的交集

图 11 显示了使用这个模型表示三个时空外延。possible_individual # 9012 是 possible_individual # 1234 和 # 5678 的一部分。

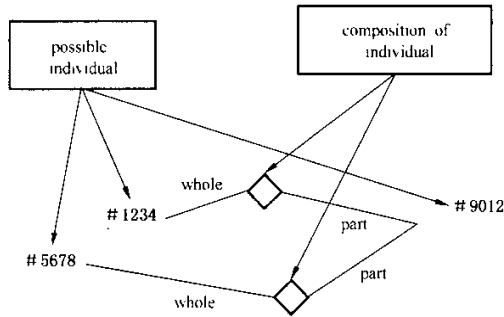


图 11 个体组合实例图

4.7.2 个体的临时部件

在本部分中,对应于 possible_individual 整个时空外延中的一段时间中的另一种 possible_individual 被称为临时部件。图 12 中显示了临时部件的性质。

注: 状态和子状态是通常用来等同于临时部件的术语。

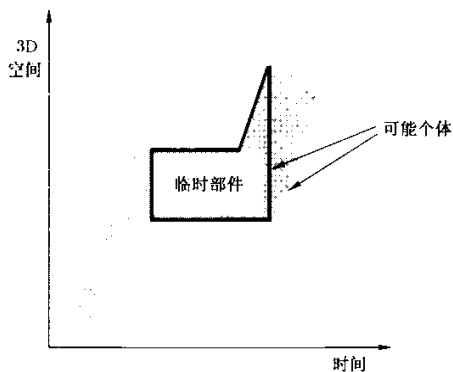


图 12 临时部件

表示临时部件的实体数据类型被定义为 composition_of_individual 的子类型,如图 13 所示。

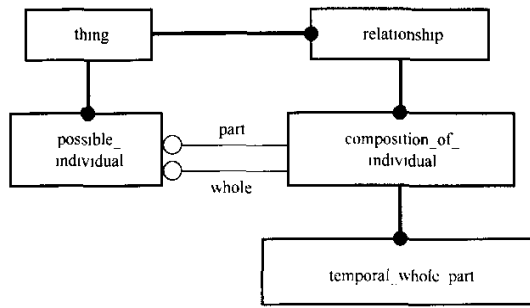


图 13 临时整体部件关系

示例：在图 14 中 #9012 是叶轮 #1234 的临时部件。其 composition_of_individual 关系可以被认为是 temporal_whole_part(临时整体部件)关系。

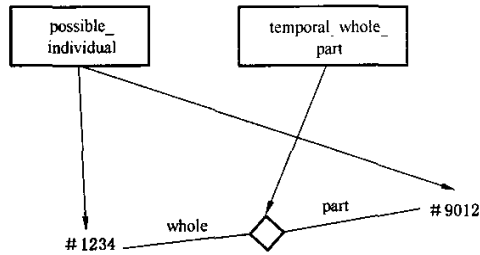


图 14 #1234 的临时部件 #9012

基于临时外延的特点，所有 possible_individual 的特征都可以应用到它的临时部件上，但整体性的概念除外。然而临时部件的特征不必要应用到整个的 possible_individual 上。

4.7.3 个体的连接

possible_individual 在其生命中可能被连接，这样才能与其他个体进行交互(见 5.2.21 和图 197)。这种连接可以是直接的，即有一个共同的空间边界。也可能是通过其他 possible_individual 间接发生。在这种情况下中间的个体就存在一个可能不被记录的有向连接链。

图 15 表示了一个 direct_connection(直接连接)的时空性质。两个个体在生命中的某一段联系起来。两个个体的临时部件“A”和“B”的连接在空间中有一个共同的边界。在这段时间内这个边界是稳定的，尽管在本例子中不是必须的。“A”和“B”也可以被看作另一个个体“W”的部分，图中没有表示出来。“A”和“B”连接起来并不表示它们是一个整体“W”的部分。这种关系必须用 composition_of_individual 关系单独表示。类似地，一个整体包含两个部分，并不意味着这两个部分有连接关系。

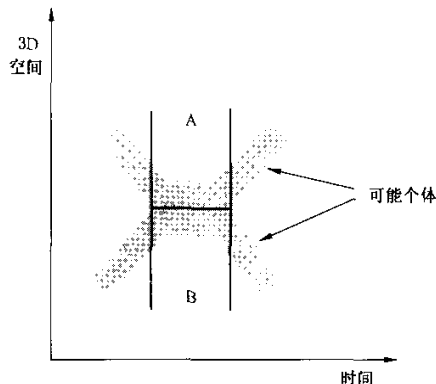


图 15 相连接的时空外延

图 16 所示连接模型中的元素。connection_of_individual(个体连接)是一种 relationship,有 direct_connection 和 indirect_connection(间接连接)的子类型。属性 side_1(边 1)和 side_2(边 2)没有隐含任何次序和方向,只是用来区分参与连接的不同个体。

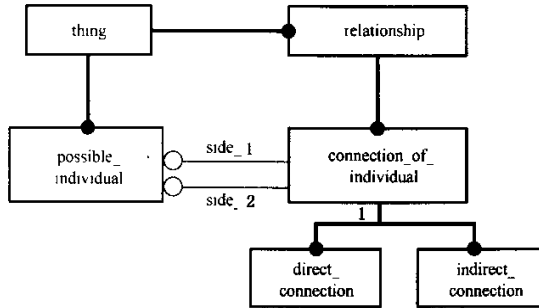


图 16 个体连接

示例 1: 图 17 显示了一台特定的发动机轴和封轴之间的连接。这种连接只涉及轴和轴封的临时部件。

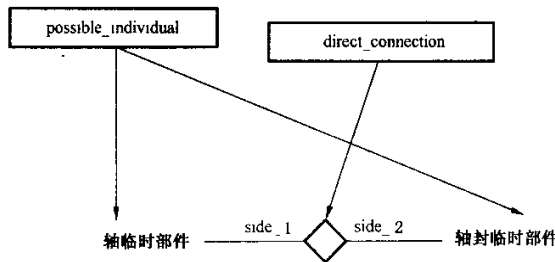


图 17 轴与封条的直接连接

一个 indirect_connection 是通过其他个体发生的。图 18 显示了 individual_used_in_connection(连接使用个体)的关系。这使得其他参与 indirect_connection 的个体被记录下来。

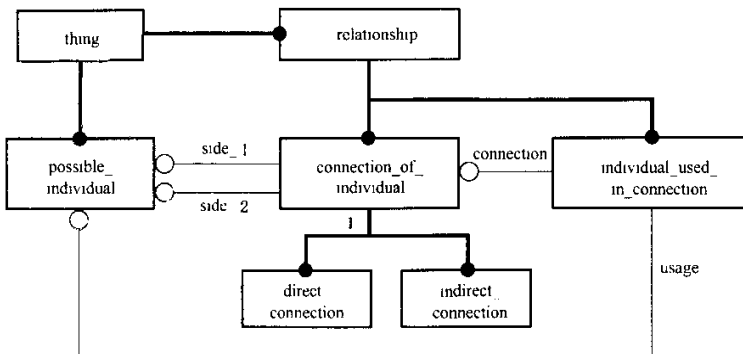


图 18 连接中使用的个体

示例 2: 图 19 显示了发动机轴和曲轴箱之间的非直接连接,这种连接是通过轴承和轴封连接起来。

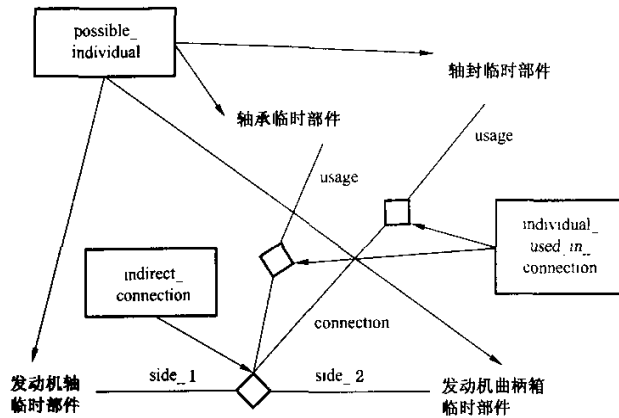


图 19 轴和曲柄轴箱的非直接关系

4.7.4 个体的临时顺序

如果用按时间对时空外延进行排序时,一个可以完全在另一个的后面。这在本部分中称为 temporal_sequence(临时顺序)(见 5.2.22.6 和图 198)。图 20 显示了 temporal_sequence 的时空性质,个体“ A”在时间轴上整体先于个体“ B”。

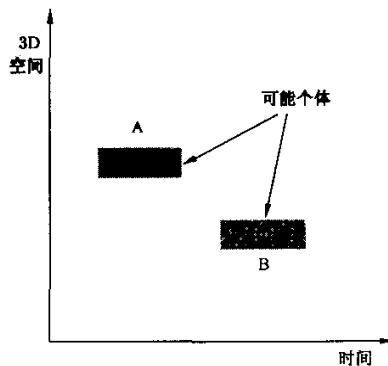


图 20 时空外延排序

图 21 显示了顺序模型的元素。temporal_sequence 是一种前驱和后续 possible_individual 之间的 relationship,它定义了一个时间顺序。

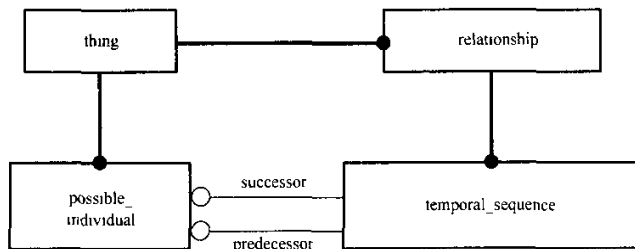


图 21 时间顺序

顺序只按实际情况进行记录。与顺序有关的规则,即成员中有先后关系的性质,是用 temporal_sequence 类来表示的(见 4.8.4.7 和 5.2.22.3)。

示例:图 22 中显示了 possible_individual 詹姆斯瓦特出现在黑斯廷斯战役之后。

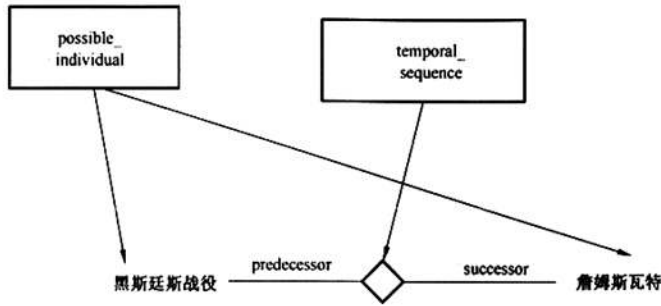


图 22 詹姆斯瓦特和黑斯廷斯战役

4.7.5 个体子类型

图 23 中显示了数据模型中 possible_individual 的直接子类型。

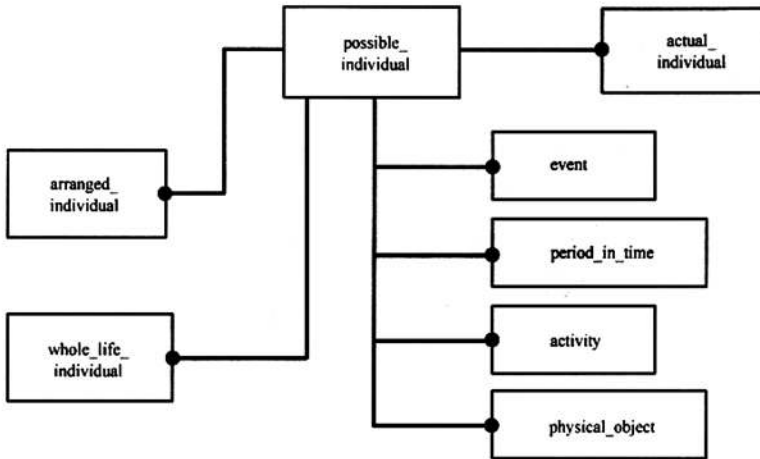


图 23 可能个体的子类型

4.7.6 实际个体

actual_individual(实际个体)是在现实世界中实际存在的 possible_individual。过去计划过或期待将来出现的 thing, 如果没有出现, 就只是 possible_individual。只有那些出现过的 possible_individual 才是 actual_individual。

示例: 欧米茄生产系统从 1998 年 3 月 21 日起需要的一台每分钟 25 加仑的泵是一个 possible_individual。1998 年 7 月制造的有每分钟 28 加仑容量的泵是一个 actual_individual。所需要的和实际的 28gpm 的泵的时间外延如图 24 所示。因为它们有不同的时空边界, 所以它们的时空外延不同。

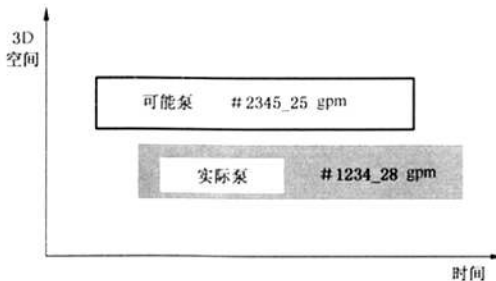


图 24 可能和实际的个体

图 25 显示了这些外延的实例图。两个 possible_individual 分别是 #2345 和 #1234, 对应了不同的外延。#1234 既是一个 actual_individual, 又是一个 possible_individual。

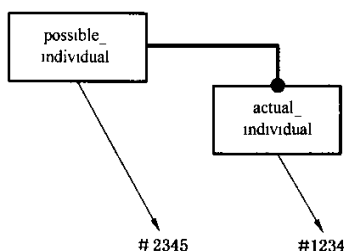


图 25 可能和实际实体的实例图

4.7.7 个体的生命周期阶段

组织和个人使用生命周期的概念,如提出、计划和需要等时,经常涉及 possible_individual。在本部分中 lifecycle_stage 被看成是两个 possible_individual(见 5.2.23.4、5.2.23.5 和图 199)之间的关系,如图 26 所示。

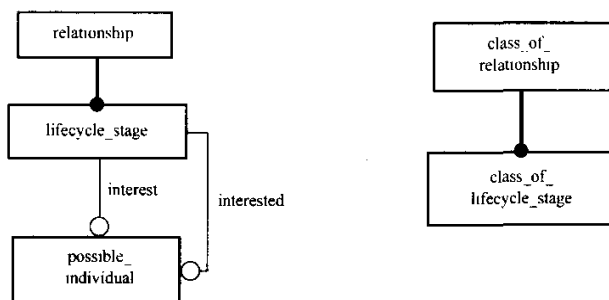


图 26 生命周期关系

class_of_lifecycle_stage(生命周期阶段类)的关系成员中没有约束。

示例: 图 27 显示了 XYZ 公司在 2002 年 1 月 6 日至 9 月 27 日需要 25gpm 泵 #2345。也可以认为还有另一家公司也需要和计划用这同一个 possible_individual。

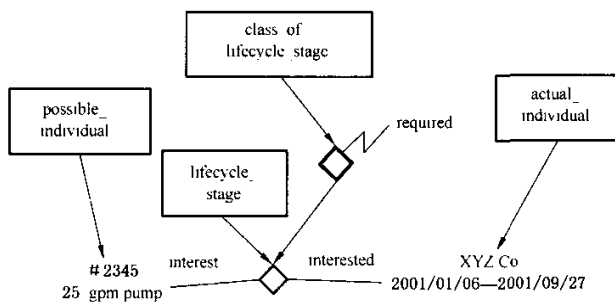


图 27 XYZ 公司需要的泵

4.7.8 完整生命个体

一个 whole_life_individual(完整生命个体)是指一个时空外延,并且它不是任何同类个体的临时部件(见 5.2.6.15 和图 182)。这些个体的特性相对于其他个体的特性是独立的。

示例 1: 编号为 #1234 的实际叶轮是一个 whole_life_individual。它不是任何一个其他叶轮的临时部件。

示例 2: 以一个塑料胚,从胚做成的杯子,再到把杯子压碎成塑料为例,这些都是塑料件个体的临时部件,它的分子结构与胚、杯子和压碎废品的分子结构是相同的。由于杯子不是塑料件的同类,所以它可能被看成是一个 whole_life_individual。类似地,塑料胚和压碎的塑料也是 whole_life_individual,如图 28 所示。

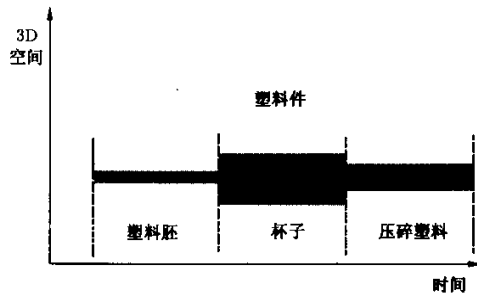


图 28 塑料件的时空图

示例 3：图 29 显示了把杯子 # X93 和塑料片 # 3A 表示成 whole_life_individual 的模型实例，在 temporal_whole_part 关系中杯子是部件，塑料件是整体。

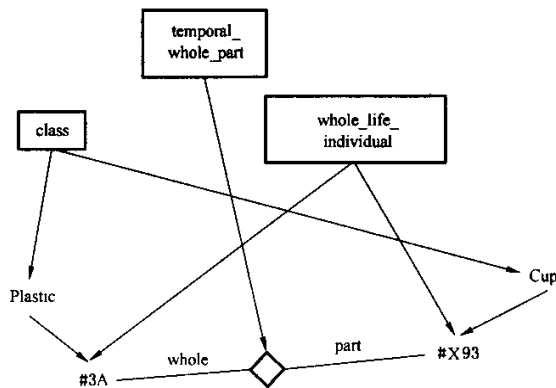


图 29 作为整体生命个体的塑料件和杯子

4.7.9 已排列个体

possible_individual 由其他 possible_individual 部件组成，如图 30 所示。有特定组织或排列的部件是 arranged_individual(已排列个体)(见 5.2.6.2 和图 182)。arranged_individual 与其所包含的个体部件在属性、特征和行为上不同。

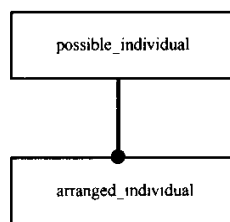


图 30 已排列个体

示例 1：一个带制造商序列号的特定的泵是一个 arranged_individual。泵的抽吸能力是它的个体部件所没有的行为。

示例 2：库存备用泵的叶轮不是一个 arranged_individual，不能提供整体机组的功能。

4.7.9.1 个体排列

arrangement_of_individual(个体的排列)是 composition_of_individual 的一个子类型，限指 whole(整体)中的 arranged_individual(见图 31)。一个 arrangement_of_individual 意味着相对于 whole 中其他部件(见 5.2.6.3 和图 182)，这个 part(部件)已经被排列了。

示例 1：按编队飞行的飞机是一个 arranged_individual。arrangement_of_individual 关系显示每个飞机的临时部件是编队的部件。当飞机在地面上时，它们不是编队的部件。因此编队由飞机的临时部件组成。

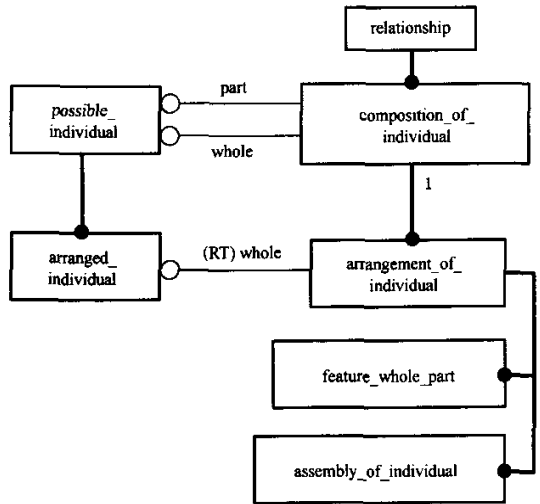


图 31 个体排列

定义 arrangement_of_individual 的两个子类型,以区分 arranged_individual 是部件组装还是特征。

assembly_of_individual (个体组装)关系表示整体中的部件是组件临时部件,并直接相连(见 5.2.6.4)。隐含连接的性质是指部件可以合理地连接起来或从整体分来,特别是使用一些机械的方法或焊接、粘贴和其他形式的粘附。这就允许在 arranged_individual 的生命周期中更换部件,并把它作为另一个 arranged_individual 的部件。

示例 2: 图 32 显示了叶轮 #127C 的两个临时部件 A 和 B 作为部件,与两个泵 #2345 和 #2346 之间的 assembly_of_individual 关系。泵 #2345 和 #2346 是另外两个完整体的临时部件。它们可能是泵类,但这里没有表示。

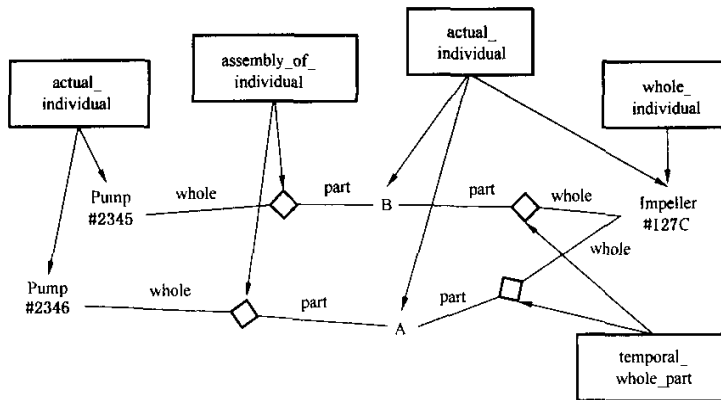


图 32 泵的组装

在本部分中 assembly_of_individual 仅限于中视镜尺寸的个体,因此不包括分子和原子大小的个体的组合。

feature_whole_part(整体部件特征)关系应用于部件与整体无法分开的 arranged_individual(见 5.2.6.6)。特征无法被当成它所指个体的部件来进行标识。在本部分的术语中,一个特征部件要么是一个 whole_life_individual,要么是一个 whole_life_individual 的部件,同时它是另外一个 arranged_individual 的特征。

示例 3: 图 33 显示了一个受腐蚀管道表面切片的数据。表面切片是 2 一个 whole_life_individual, 并是管道的一个特征。腐蚀切片中有反应腐蚀空间范围和严重程度的个体部件。这个临时部件继承了它是管道特征的条件。通过标识出一个合适的完整生命对象及这个 whole_life_individual 的临时部件,并使临时部件被不断地观察和度量,腐蚀发展过程就可以被全程跟踪。

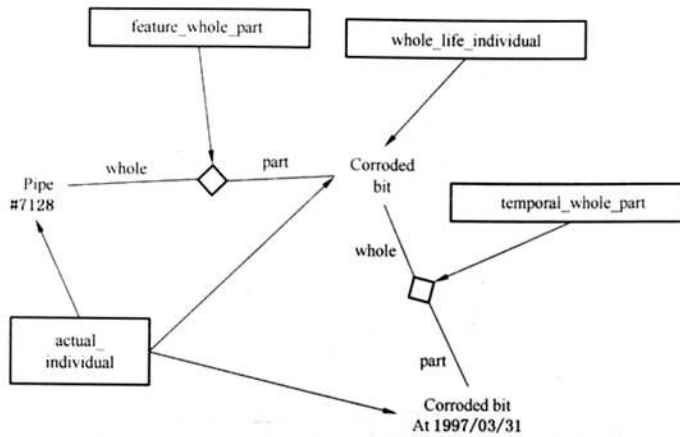


图 33 腐蚀特征

4.7.10 事件和时间点

在本部分中, event(事件)被定义成具有零时间外延的时空外延(见 5.2.9.5 和图 195)。时间可能只发生在某一个时间,也可能是在一个连续时间,或这两者的组合。图 34 显示了一次和连续 event 的时空图。两种类型都满足零时间外延的条件,因为每个部分都是零持续时间的。

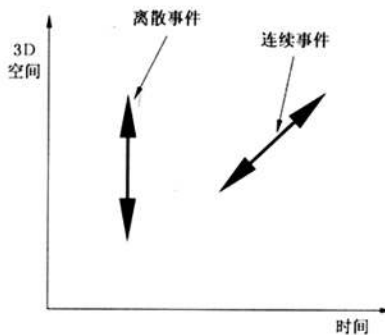


图 34 事件的时空外延

event 标记了 possible_individual 的时间边界。图 35 显示了一个对象的时空图,它在移动后休息了一下,然后继续移动。一次 event “A”和“B”描绘了这个对象固定不动的临时部件。连续 event “C”标记了对象移动时的上边界。event “D”有两个部件“C”和“B”是对象的移动临时部件的开始时间边界。

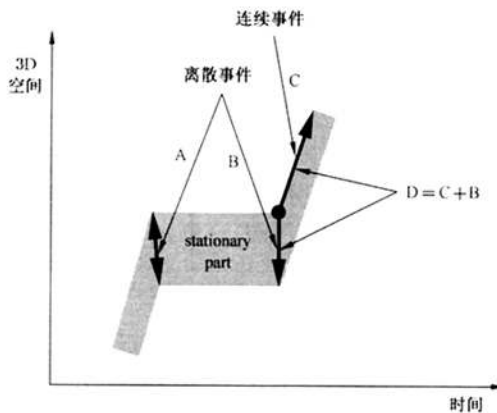


图 35 事件边界时空图

图 36 显示了 event 的模型。event 是 possible_individual 的子类型,也可以是一个 actual_individual。一个 event 是其他时空外延的 temporal_bounding(临时边界)。事件边界是 composition_of_individual 的子类,其中部件限指一个 event。其子类 beginning(开始)和 ending(结束)表示了有边界个体的 temporal_sequence。

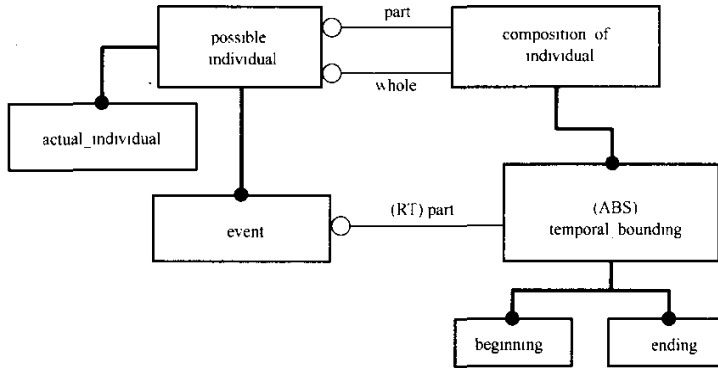


图 36 事件的建模图

示例 1: 一个清管器在已经放置了一段时间后要停下来,并在重新移开之前。在清管器稳定状态的最后是一个 event 和一个 actual_individual。图 37 显示了其中的数据。

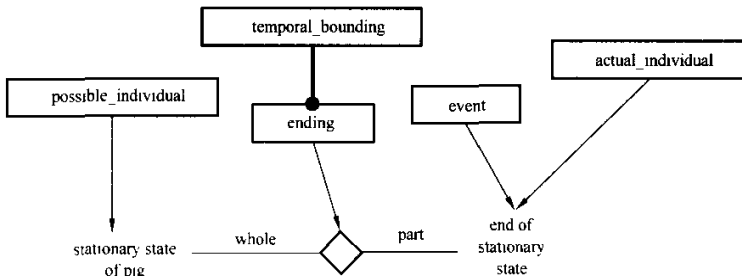


图 37 稳定状态结尾的实例图

示例 2: 图 38 显示了清管器在移动中前端的时空轨迹。移动状态的开始也是稳定状态的结束。这两个都是移动状态的 beginning event(开始事件)的部分。

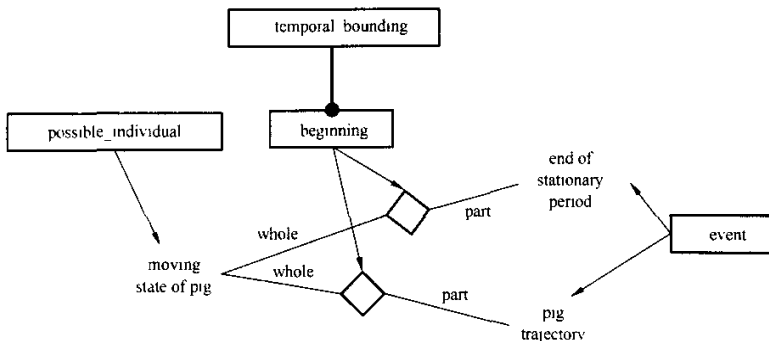


图 38 铸模时空轨迹的实例图

有些一次 event 包含了整个空间外延,则称为 point_in_time(时间点)(见 5.2.9.8 和图 185)。一次 event 总是一个 point_in_time 的部分,如图 39 所示。

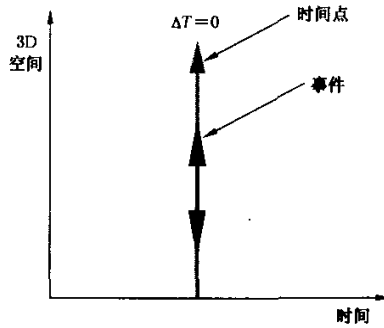


图 39 时间点的外延

图 40 显示了 event 和 point_in_time 的模型。由于 event 和 point_in_time 都是 possible_individual，所以它们也可能是 actual_individual。

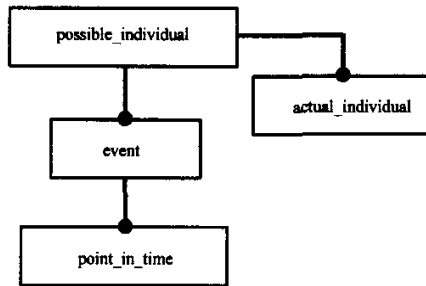


图 40 事件的模型图

示例 3：如我们知道的国际调整时间 2002 年 11 月 17 日上午 10 点是一个 point_in_time 和 actual_individual(见图 41)。

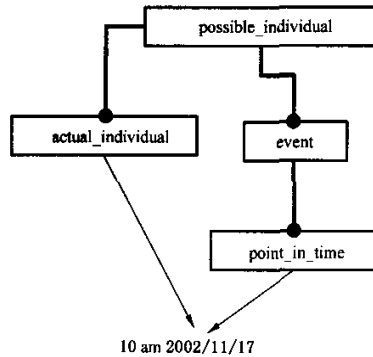


图 41 国际调整时间 2002 年 11 月 17 日上午 10 点作为实际时间点的实例图

不是时间点的 event 是定义该 event 的 point_in_time 的空间部件。这个关系用 composition_of_individual 来表示。

示例 4：图 42 显示了自国际调整时间 2002 年 11 月 17 日上午 10 点起，清管器开始呈现稳定的状态。

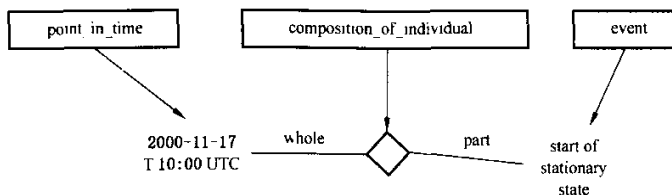


图 42 稳定状态时间的实例图

4.7.11 时间段

period_in_time(时间段)是一个 possible_individual。它是指一段时间的所有空间,即世界的临时部件。图 43 显示了 period_in_time 的时空性质。

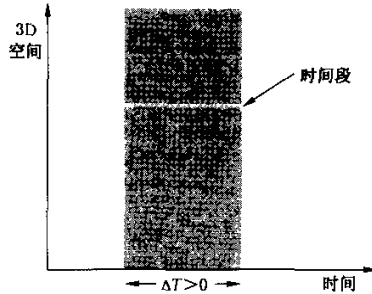


图 43 时间段的时空外延

图 44 显示了 period_in_time 的模型。

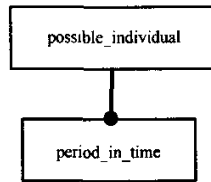


图 44 时间段的实体类型

图 45 显示了时间段的边界是时间的 beginning 和 ending 时间点。

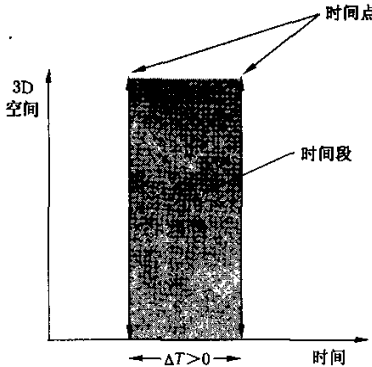


图 45 时间段和它边界时间点的时空图

示例: 图 46 显示了 #EH26 是一个 period_in_time, 其开始时间是 10:26, 结束时间是 11:09。

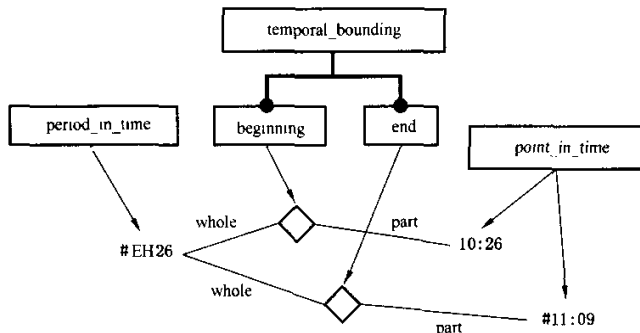


图 46 从 10:26 到 11:09 的时间段

4.7.12 物理对象

在本部分中, `physical_object` (物理对象) 是时空中的物质和/或能量的分布 (见 5.2.6.10 和图 182)。`physical_object` 总是空间的一部分, 尽管不必要但通常也具有非零的时间外延。

`physical_object` 和 `activity` (活动) 不是相斥的。一个 `possible_individual` 可以既是 `physical_object` 也可以是活动。

示例: “放射性材料”、“活的组织”和“火”就既是 `physical_object` 又是 `activity`。

基于 `physical_object` 的连贯性, 它可以分成四种, 即图 47 中所示的 `physical_object` 的子类型。

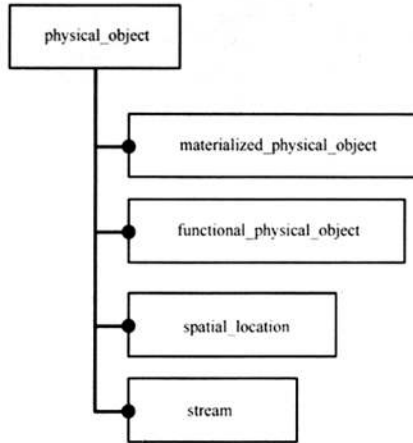


图 47 物理对象的类型

4.7.13 物化物理对象

`materialized_physical_object` (物化物理对象) 是一种 `physical_object`。在其存在期间它是由同样或缓慢变化的物质或能量组成的 (见 5.2.6.8 和图 182)。它包含了大多数经常描述的具体对象。

示例: 一个带有厂家系列号的机械泵就是一个 `materialized_physical_object`。在其生命中, 即使它的每一个零件都换掉了, 我们还会把它视为同一个事物。图 48 显示了一台泵的时空图。在开始时, 泵由两个部件 A 和 B 组成, 过了一段时间部件 A 被一个新的部件 C 替换了, 后来 B 又被 D 替换了。在所有事件过程中, 由于材料总是存在的, 所以材料的连续性就得到了保持。

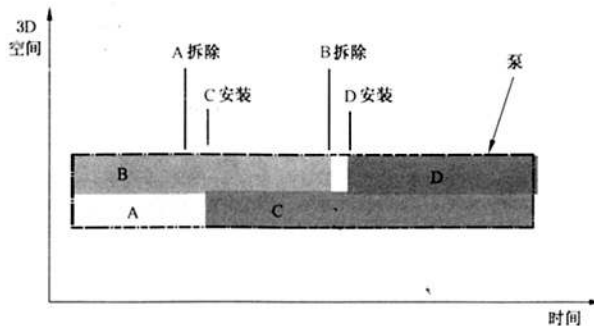


图 48 材料连续性时空图

泵由一些临时部件组成。每一个临时部件都对应于泵的一个零件,如图 49 所示。

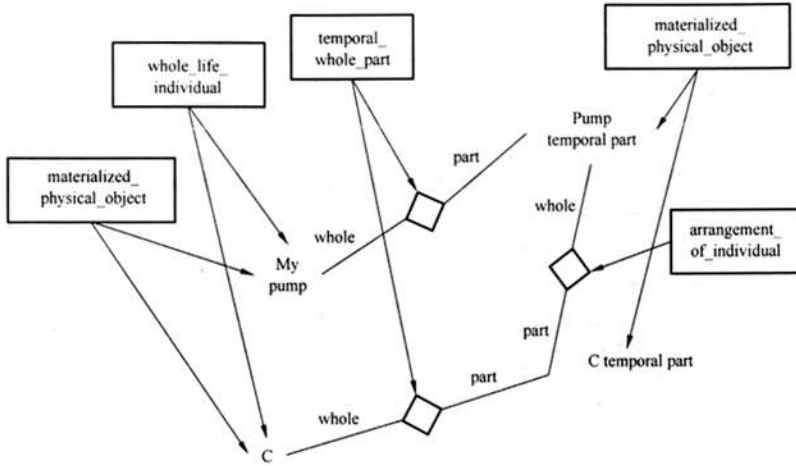


图 49 泵的临时部件

4.7.14 功能物理对象

functional_physical_object(功能物理对象)是基于一定功能连续性的 physical_object(见 5.2.6.7 和图 182)。

temporal_whole_part 关系指出了那些 materialized_physical_object 的临时部件是 functional_physical_object 的部件。

示例: 有一个位号为“P101”的泵,其实际的泵设备更换了多次,如图 50 所示。它就是一个 functional_physical_object。位号 P101 上安装了泵 1 后又移走了,随后泵 2 安装了又移走了。这里没有物质的连续性,但安装上的泵都起到了类似的作用。图 51 显示了泵 1 的实例图。

那个标签和泵的共同临时部件的 possible_individual,既是 materialized_physical_object 又是 functional_physical_object。

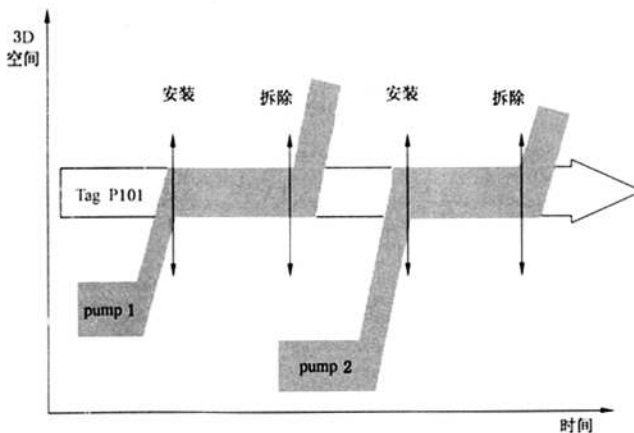


图 50 功能物理对象 P101 的时空图

4.7.15 空间区域

spatial_location(空间区域)是具有相对位置连续性的 physical_object(见 5.2.6.12 和图 182)。

示例: 一个海上许可区域就是一个 spatial_location。

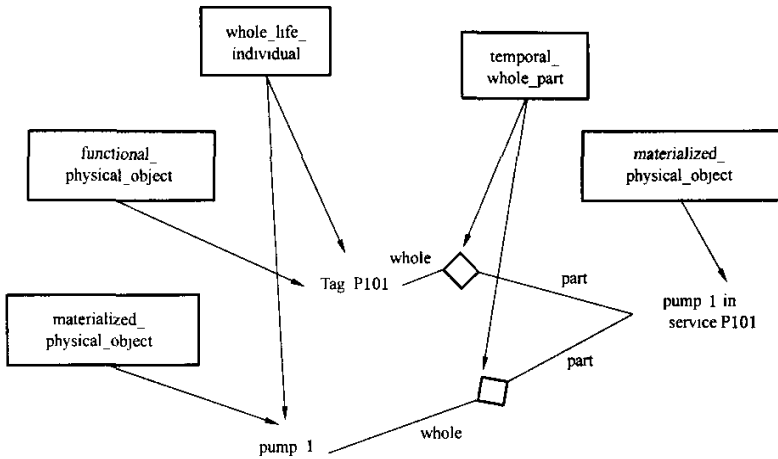


图 51 安装在位号 P101 的泵 1 实例图

4.7.16 流

stream(流)是一种 physical_object,其中流动通道的连续性是识别的基础(见 5.2.6.13 和图 182)。

示例:软管的流量是 stream 和 materialized_physical_object。

4.7.17 活动

一个 activity 是一些正在发生或改变的事物(见 5.2.9.1 和图 185)。在本部分中,活动是指其他个体和 event 参与的时空外延。participation(参与)是 composition_of_individual 的一种类型,如图 52 所示(见 5.2.9.7)。

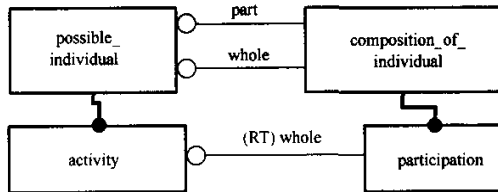


图 52 活动参与

示例 1:使用一个热压机把平整的塑料胚加工成塑料杯,就是一个活动。活动的外延包括外形被改变了的塑料和所使用机器的临时部件。图 53 显示了参加“杯子成形”活动的塑料的时空外延。成形活动使杯子诞生,并结束了塑料胚的生命。

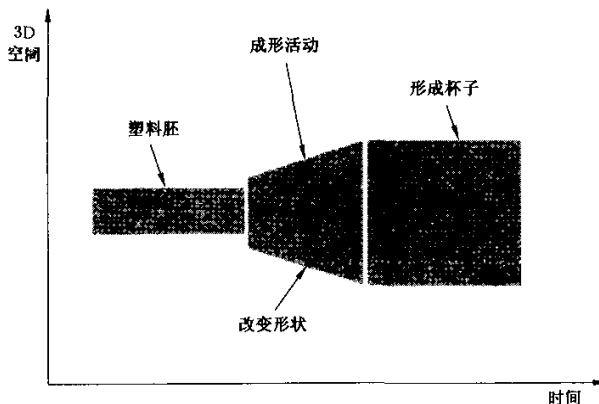


图 53 杯子成形活动

图 54 显示了热模机、塑料和杯子参与时的模型实例。

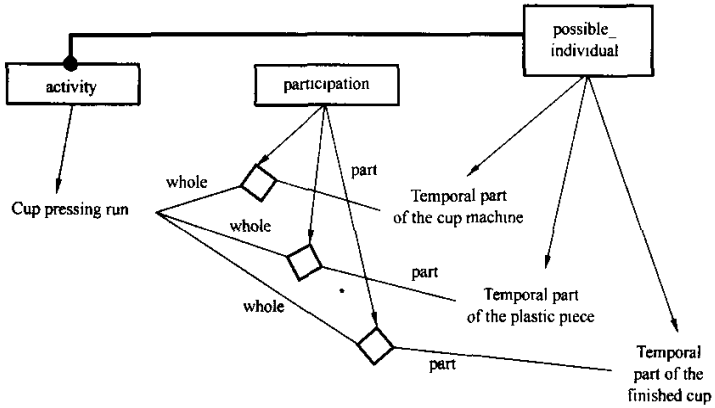


图 54 杯子成形活动的实例图

带来变化的活动可以用 event 来标记(见 5.2.9.5)。在图 53 中显示了两个明显的变化:杯子开始出现,塑料胚生命截止。图 55 显示了这个原因模型,cause_of_event(事件原因)是 relationship 的一个子类(见 5.2.9.3)。

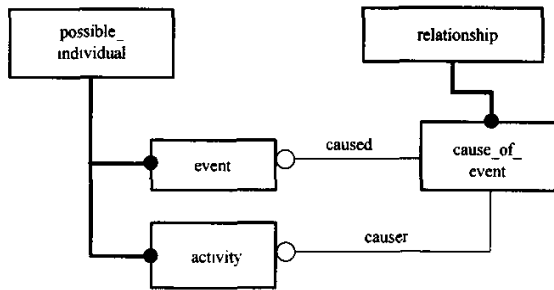


图 55 事件原因模型

图 56 显示了一个特定杯子压制 activity 造成了杯子开始出现和塑料胚结束的 event。

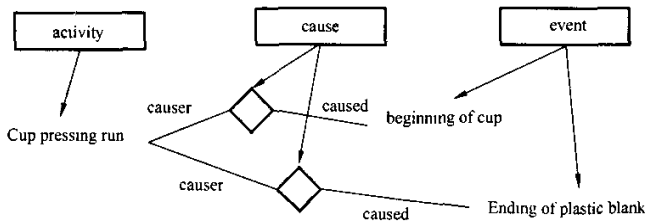


图 56 杯子压制中导致的杯子开始出现

模型也允许与 activity 有关的过去或将来的 abstract_objects 和 possible_individual 直接地包含进来,如图 57 所示(见 5.2.9.6)。

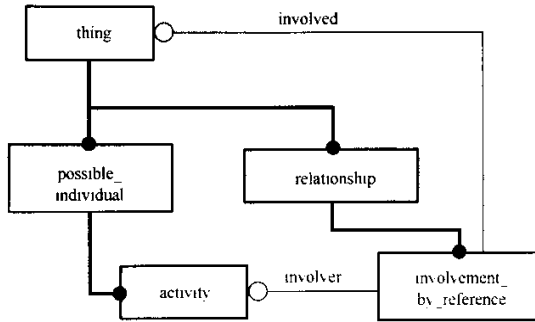


图 57 通过引用包含

示例 3：一个生产式 activity 可能包含用于个体 (activity 产生的) 的规范。规范是一种 class_of_individual，它与此活动具有 involvement_by_reference 关系。

活动可能会导致识别出抽象条件 (见 5.2.9.9)。recognition (识别) 是 activity 与 thing 之间的一种关系，其中 thing 是此 activity 的结果。图 58 显示了 recognition 的模型。

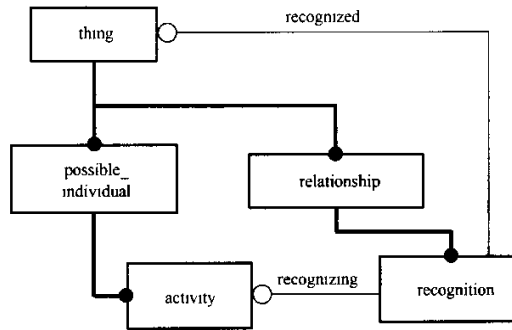


图 58 通过活动识别

示例 4：图 59 显示了对一个船舶的检查 activity，把该船舶识别成“A 级”类别。

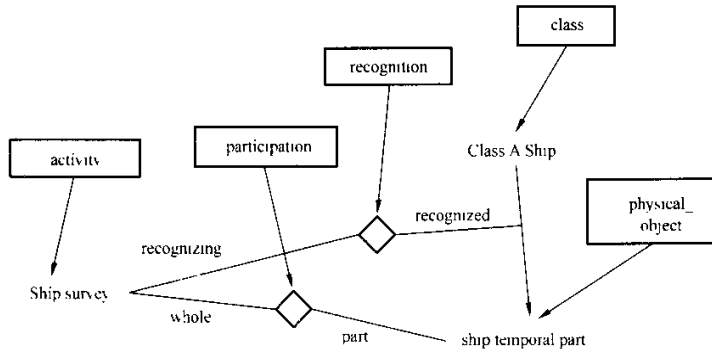


图 59 船舶分类活动

4.7.18 批准

approval (批准) 是一个 relationship，它把同意 approval 的 possible_individual，如人、组织或机器等，与被批准的事物联系起来 (见 5.2.23.1 和图 199)。在本部分中只有 relationship 才能被批准，这样就给出了 approval 的含义。一个 possible_individual 或一个 class 已经是它本身了，批准它自己是没有意义的。它批准自己被包含到某事物中才是有意义的。

图 60 显示了 approval 模型的元素。

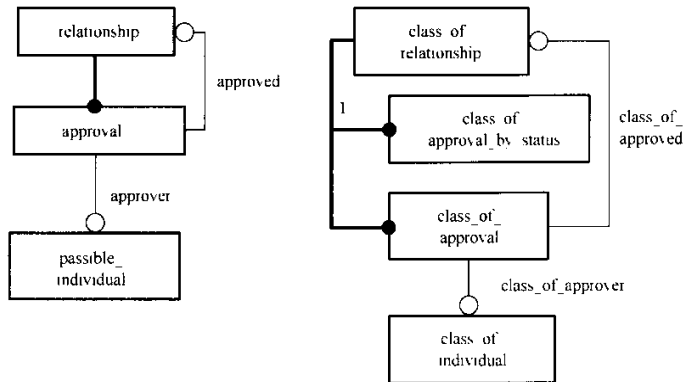


图 60 批准

approval 类型是由一个 class_of_approval_by_status(批准状态类)(见 5.2.23.3)确定的 approval 关系给出的。class_of_approval(批准类)可以对关系类建立规则,并由个体类批准(见 5.2.23.2)。

示例:图 61 显示了塑料原料在杯子成形 activity 中的 participation 行为被“产品监督员”批准。“批准”是一个 class_of_approval_by_status,“未批准”也是一个 class_of_approval_by_status。

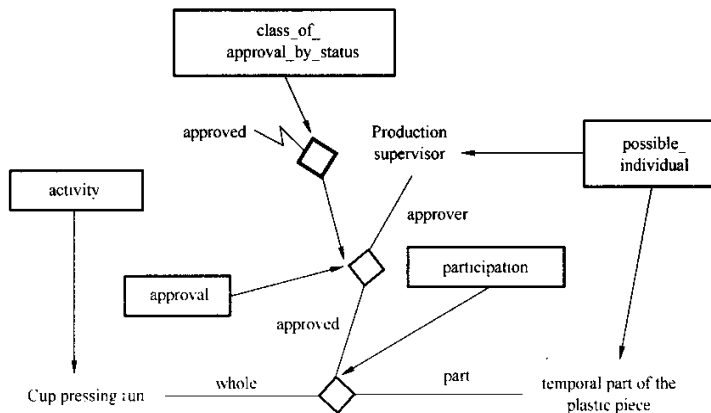


图 61 杯子原料的批准

4.8 类

class 是 thing 的一个种类、分类或分支,它具有一定的共同特性(见 5.2.2 和图 178)。

class 有一个定义了包含和不包含的基础。

在本部分中:

- class 所包含的事物可作为类的成员。
- class 对应于一个非良好构件的集合。其属性将在附录 D 中描述。
- class 所包含和不包含的事物可由一个原文定义和/或其关系给出。

4.8.1 分类

classification 是一个 relationship,标明了 class 的成员关系(见 5.2.2.3 和图 178)。类的成员关系意味着其成员满足了包含和非包含的条件。如图 62 所示,classification 被视为 relationship 的一个显式子类型。possible_individual、class、relationship 和 multidimensional_object 是不同的。归类是非及物的。一个 class 的成员不一定是该 class 的任何 class 成员。

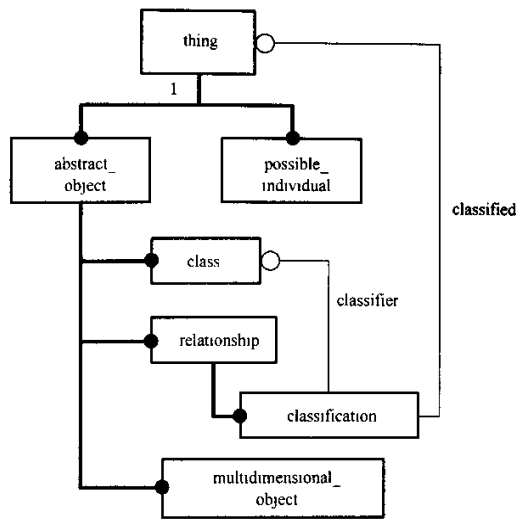


图 62 归类关系的模型

示例 1: 图 63 显示了一类 thing“泵”作为一个 class。“泵”的成员是有泵活动的 arranged_individual。arranged_individual #1234 是一个“泵”。标明 #1234 是“泵”class 成员的关系是一个 classification。在“泵”class 的例子中,其成员属性不受 arranged_individual 的影响,无论 arranged_individual 是一个 whole_life_individual 还是 whole_life_individual 的临时部件。

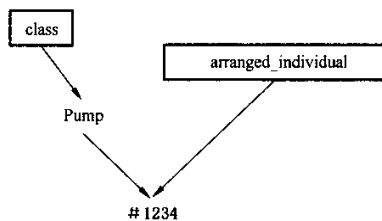


图 63 泵的归类

归类时允许事物被定义成多个类的成员。

示例 2: 如图 64 所示, #1234 作为一个 arranged_individual 是一个“泵”同时也是“正运行的”。

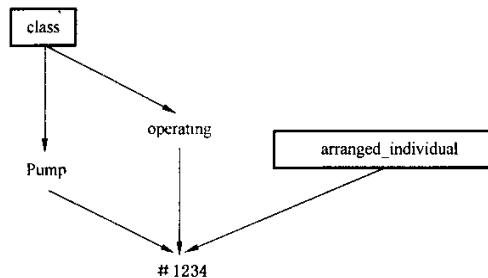


图 64 运行中泵的归类

许多归类不仅适应于一个 whole_life_individual,也适应于 whole_life_individual 的临时部件。

示例 3: 如图 65 所示, #1234 是一个完整生命泵 #PA01 的临时部件。“泵”的归类正属于 whole_life_individual,并继承自其任何临时部件。然而,如果 whole_life_individual 不在数据库中表示,与临时部件直接对应的归类会被记录下来。

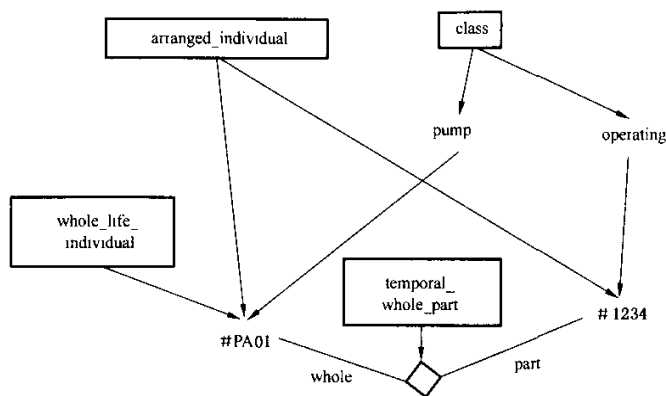


图 65 泵操作中的临时部件

4.8.2 特殊化

如图 66 所示,specification 是两个 class 之间的 relationship,其中子类的成员一定是超类的成员。specification 关系用于表示一个 class 是另一个 class 的子分支。

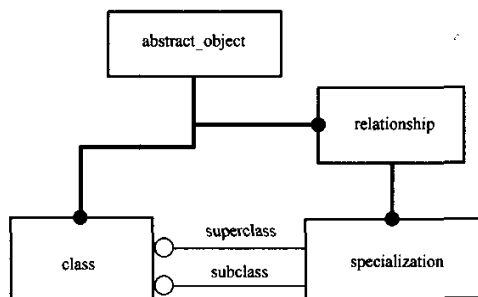


图 66 特殊化关系

因为子类的成员一定是超类的成员,子类成员关系必须符合超类成员关系的所有规则。这样子类就被称为继承了超类的规则。

示例 1: 图 67 显示了一个制造商的模型 106 是一个泵 class 的特殊化 class。“模型 106”的成员都是“泵”的成员。

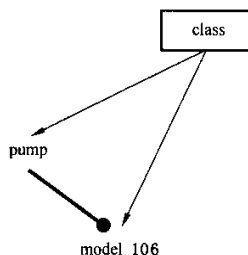


图 67 泵特殊化

特殊化关系是及物的。子类的子类成员是更广义超类的成员。

示例 2: 在图 68 中,一个制造商为“模型 106”的泵提供了两个选择,类型“A”和“B”。类型“A”和“B”都是“模型 106”的特殊化。类型“A”的成员是“模型 106”和“泵”的成员。

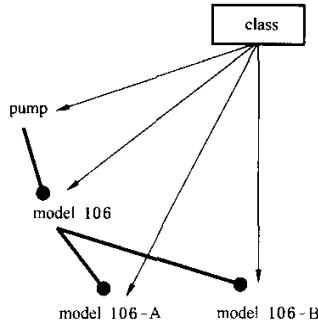


图 68 特殊化的传递性

归类 and 特殊化是本质上完全不同的。当 classification 的成员引用 class 时,成员 class 中的成员不一定是分类 class 的成员。

示例: 图 68 中“模型 106-A”的成员不是一个 class。

4.8.3 类的类型

图 69 显示了本部分中 class 的直接子类型。

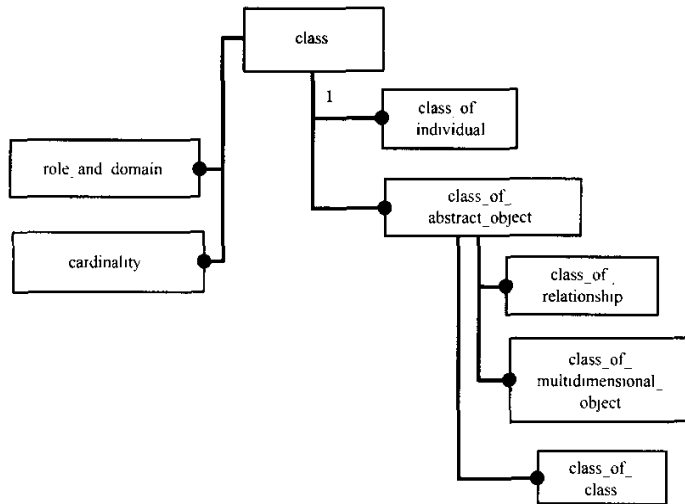


图 69 类的子类型

4.8.3.1 个体类

一个 class_of_individual(个体类)是一个 class,其成员是时空外延,即 possible_individual(见 5.2.7 和图 183)。

示例 1: “泵”类是一个 class_of_individual。

示例 2: “红色”类是一个 class_of_individual,仅时空外延才能是“红色”。

4.8.3.2 类的类

class_of_class(类的类)是一个 class,其成员都是 class(见 5.2.3 和图 179)。class_of_class 是对类成员细分的一种方法。也就是说它是用来标识细分类型的。

示例: 在图 70 中“颜色”类是一个 class_of_class。“红色”类和“蓝色”类是“颜色”class_of_class 的成员。

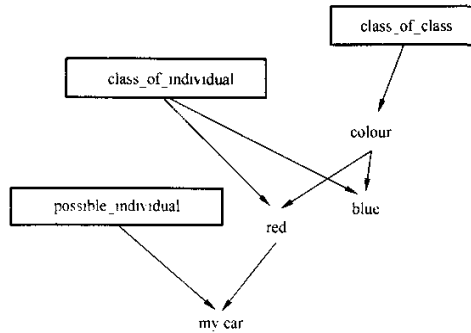


图 70 类的类“颜色”

应当说明的是本部分中定义的 EXPRESS class 的子类型是 class_of_class 的实例, 尽管 EXPRESS 语言规范中不允许这样提。

4.8.3.3 关系类

class_of_relationship(关系类)使关系类型可以被识别, 通常它用包含在成员 relationship 中的事物的类型术语定义了约束(见 5.2.12 和图 188)。

本部分中定义了 class_of_relationship 的一些显式子类型, 其关系中的角色被 EXPRESS 属性显式地定义出来。其他非显示定义的 class_of_relationship, 可以用 class_of_relationship_with_signature(带签名关系类)来处理, 这将在 4.10.2 中更完整地描述出来。

5.2.12.2 中描述了显式子类的完整清单。图 71 显示了 class_of_connection_of_individual(个体连接类)的一个显式子类型的模型。

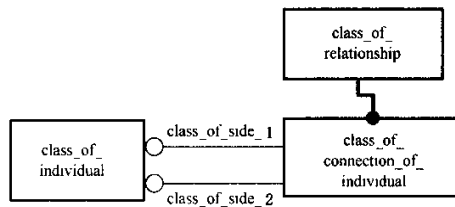


图 71 个体连接的类

示例: 在图 72 中, class_of_connection_of_individual “A 类轴承与封条的连接”说明 class_of_individual “A 类汽车轴承”的成员与“封条”的成员有连接。connection_of_individual 关系(带密封# 1234 连接轴承# 5678)是 class_of_connection_of_individual “A 类轴承密封”的成员。

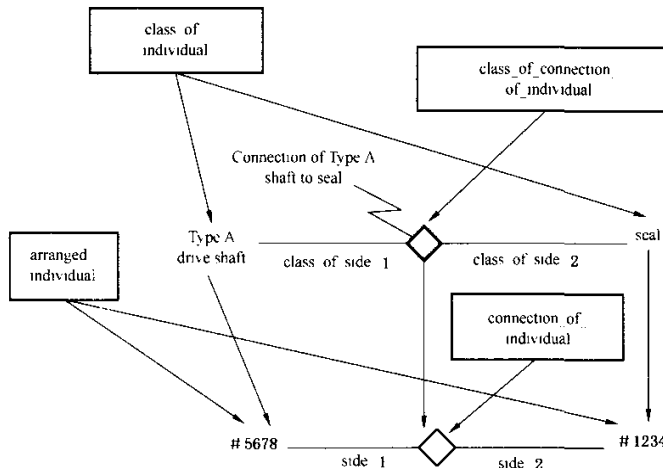


图 72 连接到 A 型驱动轴的密封

4.8.3.3.1 基数约束

一个 class_of_relationship 的基数约束可以定义成 cardinality(基数)类的属性(见 5.2.13.5, 5.2.13.1 和图 73), 其模型如图 73 所示。基数属性的作用方式与实体关系图中常见的基数约束相同。class_of_relationship 中从首端成员到末端成员所连接的成员关系数量有最大值、最小值的约束。如果没有设定最大基数, 则假定没有限制。如果没有最小基数, 则假定零是最小限定值。

end_1_cardinality(端 1 基数)和 end_2_cardinality(端 2 基数)属性可以应用于角色属性, 并符合 class_of_relationship 的 EXPRESS 定义中的顺序。

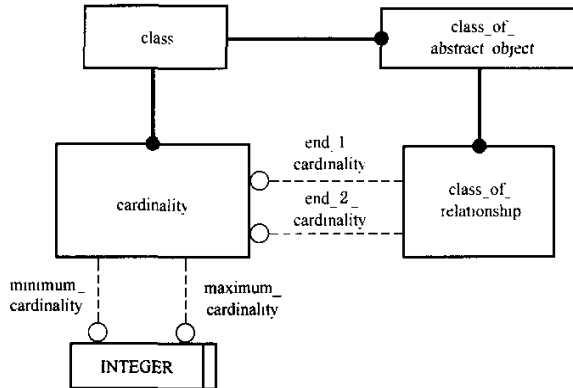


图 73 关系类的基数约束

基数约束可以应用于 class_of_relationship 的显式子类型的成员。

示例：一个 A 型汽车轴需要两个封条定位。图 74 显示了这个例子的基数。一个轴可以连接到零个、一个或两个封条, 一个封条可以连接到零个或一个轴上。轴上封条的零和一基数标识出在任何情况下为了维护目的可以移走一个或多个封条。

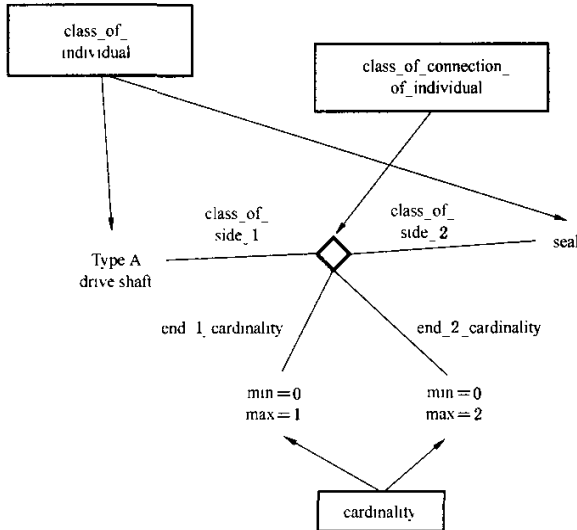


图 74 至多连接两个轴密封的 A 类驱动轴

4.10.3 中描述了 class_of_relationship_with_signature 的使用基数。

4.8.3.3.2 关系对称类

大多数 class_of_relationship 的显式子类型通过引用 class 在其 relationship 成员上定义了约束。在 class_of_connection_of_individual 的例子中, 角色 class_of_side_1(边 1 类)和 class_of_side_2(边 2 类)表示成员 relationship 都涉及类成员的参与。这些例子可以用对称 class_of_relationship 描述。

并非所有的 class_of_relationship 都是对称的。图 75 中显示的 class_of_representation_of_thing (事物表示类),其成员关系中被表示的角色指向 thing,还有一个是 class,它是指 class 而不是其成员。

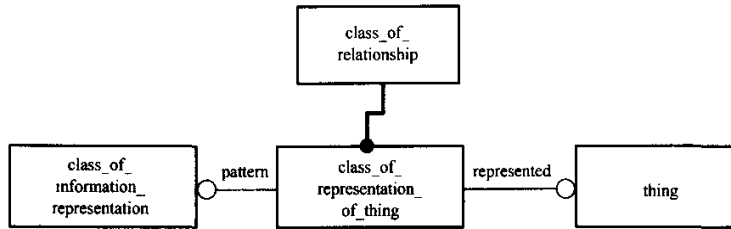


图 75 一个非对称的关系类

一个对称 class_of_relationship 可以进一步被约束成非对称 class_of_relationship,如图 76 所示的模型。一个 class_of_relationship 可以是一个 class_of_connection_of_individual,也可以是一个与 class_of_relationship_with_related_end_2(端 2 相关类)。相关的属性可以进一步把 class_of_side_2 约束成一个 thing。实体类型 class_of_relationship_with_related_end_1(与端 1 相关类)和 class_of_relationship_with_related_end_2 可以按照模型 EXPRESS 定义中的顺序,来约束 class_of_relationship 的角色。

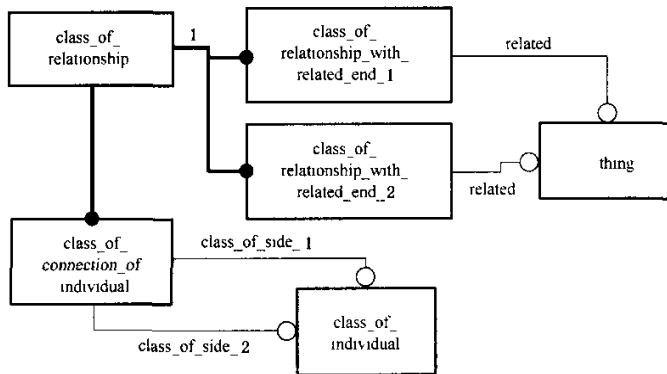


图 76 约束一个对称关系类

示例：图 77 中显示了一个类,其成员都是 connection_of_individual 关系,把封条和特殊轴 #5678 连接起来。这个类的已知成员将提供与特定的轴相连的封条的历史情况。图 77 中显示的 class_of_connection_of_individual 是图 72 中“A 类轴和封条” class_of_connection_of_individual 的特殊化。

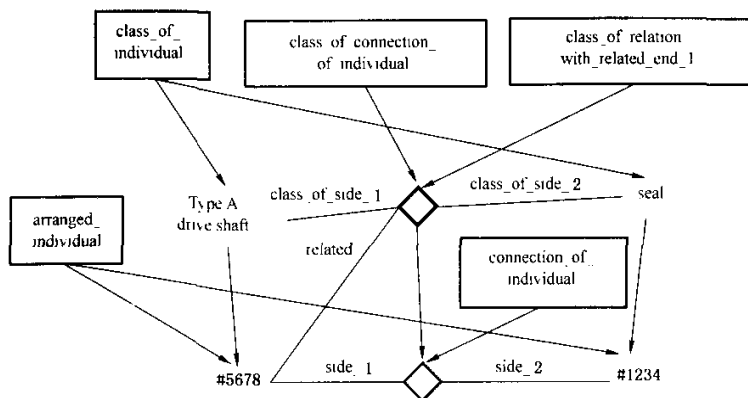


图 77 连接到特定轴的密封

在此例中相关的属性是 class_of_side_1 属性的特殊化,但 EXPRESS 语言的规则不允许这样显示。

4.8.4 个体类

class_of_individual 是一个类,其成员都是时空外延即 possible_individual(见 5.2.7 和图 183)。图 78 显示了 class_of_individual 的显式定义子类型。

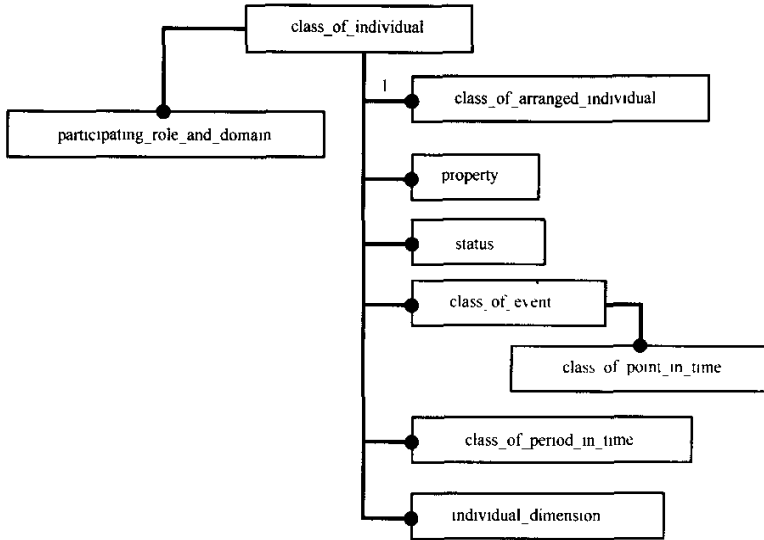


图 78 个体类的子类型

class_of_individual 成员的组合规则可以用 class_of_composition_of_individual(个体组合类)来描述(见 5.2.7.5),其模型如图 79 所示。定义的显式子类型有 class_of_temporal_whole_part(临时整体部分类)、class_of_participation(参与类)、class_of_arrangement_of_individual(个体排列类)和 class_of_assembly_of_individual(个体组装类)(见 5.2.7.12,5.2.10.5, 5.2.7.1 和 5.2.7.2)。

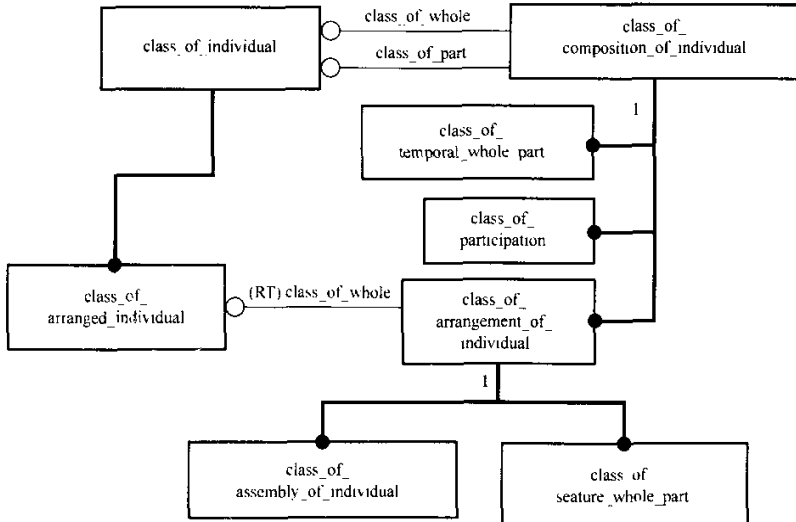


图 79 个体组合类及其子类型

class_of_composition_of_individual(个体组合类)可以用来在整个类的成员上定义条件,使它们包含部件类的成员。

示例:图 80 中的离心泵是一个 class_of_arranged_individual(已排列个体类)。class_of_assembly_of_individual “泵叶轮”记录了“离心泵”有“叶轮”这样一个部件。

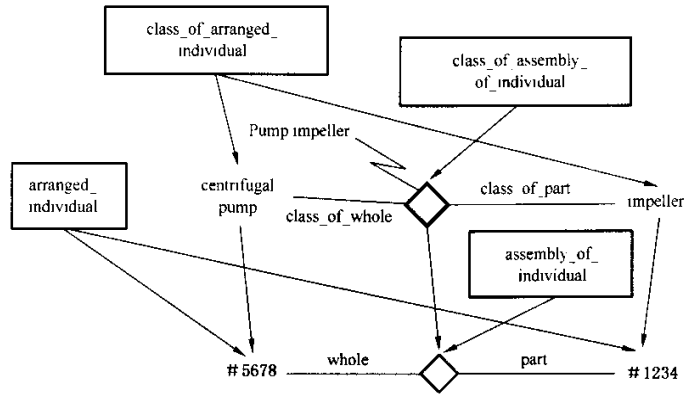


图 80 离心泵的组合

4.8.4.1 已排列个体类

arranged_individual 的部件有特定的整体排列或角色(见 5.2.6.2)。class_of_arranged_individual 是 arranged_individual 的细分,其显式子类型涉及:

- 基于物质组织的排列。
- 基于信息表达和表示的排列。
- ……与日常对象的实用相描述一致的复杂排列。

4.8.4.1.1 物质组织类

class_of_arranged_individual 的各种物质组织显式子类型包括(见 5.2.8 和图 184):

- class_of_sub_atomic_particle(亚原子颗粒类);
- class_of_atom(原子类);
- class_of_molecule(分子类);
- class_of_compound(化合物类);
- crystalline_structure(晶体结构);
- phase(物质的态);
- class_of_particulate_material(微粒材料类);
- class_of_composite_material(复合材料类);
- class_of_functional_object(功能对象类);
- ……class_of_biological_matter(生物物质类)。

图 81 显式了物质组织类的模型。

物质组织类型反映了组合排列的递增层次,从亚原子颗粒到最终的功能对象和生物物质。

示例 1: 氢原子是一个 class_of_atom。“氢原子”的成员是原子个体,它是“中子”、“质子”和“电子”的排列。而“中子”、“质子”和“电子”是 class_of_sub_atomic_particle 的成员。

在某些情况下,每一级 class 的成员是下一级 class 成员的聚集。在本部分中,大规模分子或原子的聚集被定义成化合物。

示例 2: 一个氢分子的聚集也许组成了一个气体,它是 class_of_compound “氢”的成员。

示例 3: “水”是一个 class_of_compound,它是 H₂O 分子的排列。图 82、图 83 显示水按 H₂O 分子排列次序的实体图解。

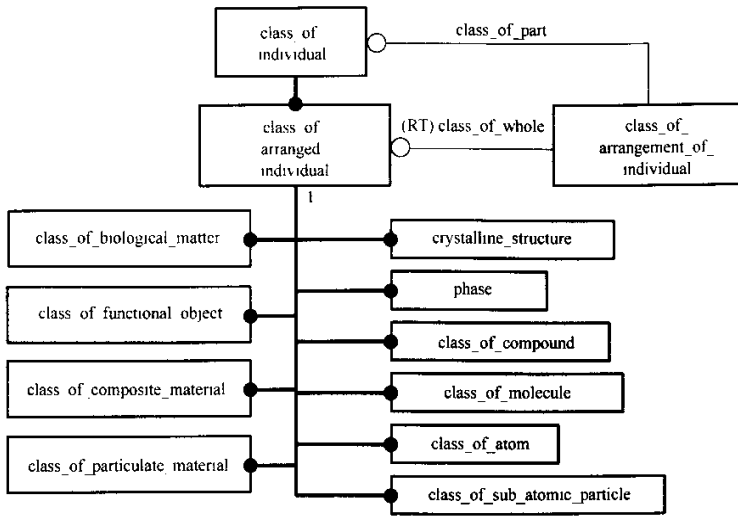


图 81 物质结构的已排列个体类

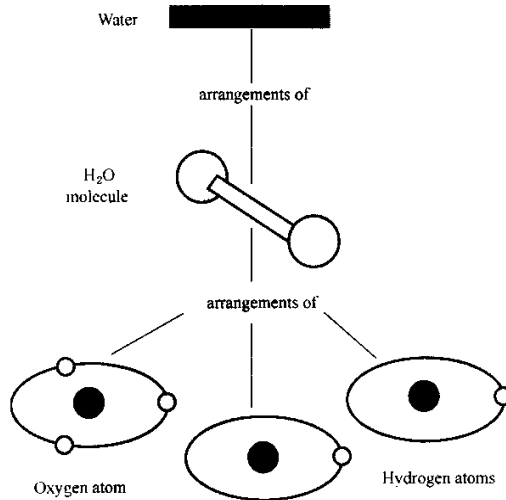


图 82 水的排列层次

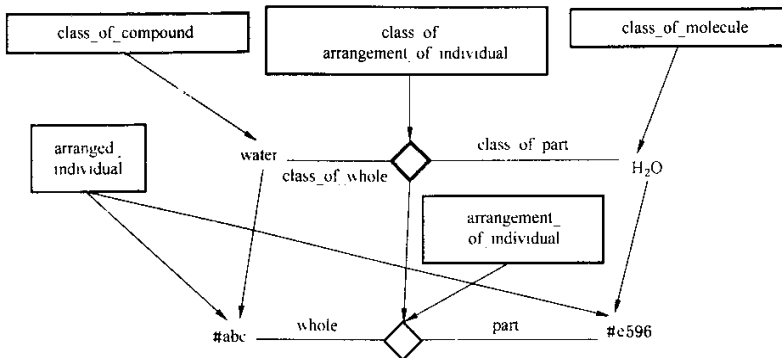


图 83 H₂O 分子的排列

为 crystalline_structure 和 phase 分别定义了两个 class_of_arranged_individual。它们是由化合物层上的物质聚集而成的。

示例 4：一团氢气是 class_of_compound “氢气”的成员，也是 phase 类“气体”的成员。

示例 5：钻石是带有特殊 crystalline_structure 的碳原子的固体聚集。

微粒和组合物物质类是更高级的排列，其部件都是化合物。

示例 6：“沙子”是一个 class_of_particulate_material，其部件都是“硅土化合物”类的成员。

示例 7：“玻璃纤维”是一个 class_of_composite_material，其部件都是“玻璃纤维材料合成物”类和“树脂化合物”类的成员。

功能对象是组织的最高层。这里，arranged_individual 是适合于某些功能和目的。其他使功能得以实现的因素如重量、尺寸、外形和材质不考虑在内。目的和形式的组合由 4.8.4.1.2 描述的复杂排列类来识别。

示例 8：“杯子”和“泵”是 class_of_functional_object 的例子。

示例 9：“离心泵”不是一个 class_of_functional_object，离心是泵功能的形式或设计。

4.8.4.1.2 复杂排列

复杂排列类使更详细的个体类型能被识别(见图 184)。它们通常使许多其他类的交集，把材料、外形和特性等方面组合起来。图 84 显示了这些复杂排列的显式类型。

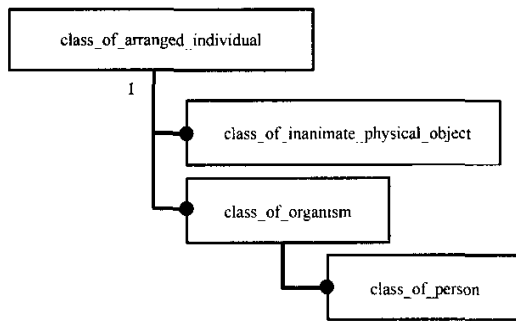


图 84 已安排个体的复杂类

示例：“塑料杯”是一个 class_of_inanimate_physical_object(无生命物理对象类)，是 class_of_functional_object “杯子”和 class_of_compound“塑料”的交集，如图 85 所示。

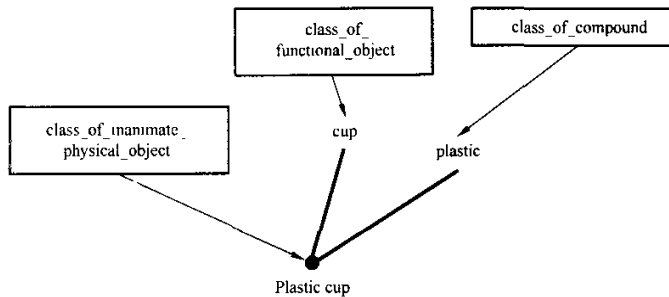


图 85 无生命物理对象类

4.8.4.1.3 信息类

使用符号来表示信息，依赖于要使用一致的和可识别的模式。模式是类。纸或图像屏幕上的一个特定的笔记或效果图可以为我们感知到，这个可能个体就是模式类的成员。

在本部分中，class_of_information_representation(信息表达类)确定了表示信息的一种模式(见 5.2.17.4 和图 193)。

渲染模式通常有许多表象上的变量如颜色、字体、尺寸和重量。class_of_information_representa-

tion 描述了这些变量(见 5.2.8.10 和图 184)。

class_of_information_representation 的成员是由可识别的模式和它们的表达风格组合而成的(见 5.2.8.9),如图 86 中所示的模型元素。

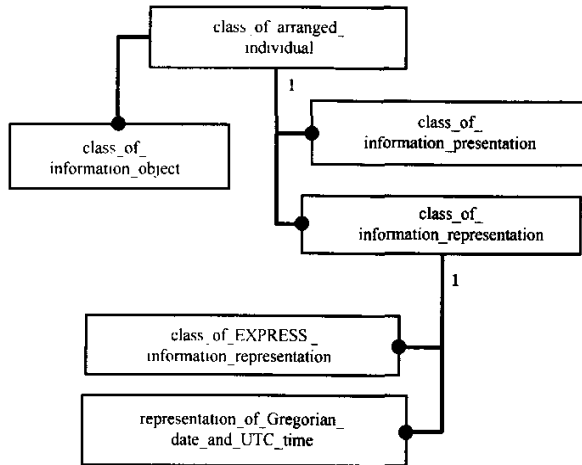


图 86 已安排个体的信息类

示例:图 87 显示了一个 arranged_individual# smith,它是 class_of_inanimate_physical_object “标签”的成员,也是 class_of_information_object “Times New Roman 粗体 24 点大的 smith”的成员。“Times New Roman 粗体 24 点大的 smith”类是 class_of_information_representation“smith”和表达类“Times new roman”、“粗体”和“24 点”类的交集。标签底层的物理属性没有显示。

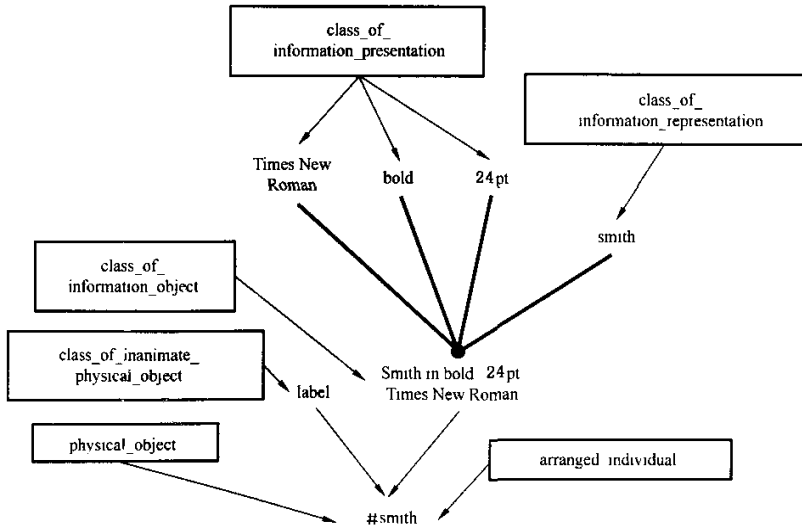


图 87 信息对象类

ISO 10303 EXPRESS 所定义的字符串、实数、整数、二进制、逻辑和布尔的文字字符模式,以及 GB/T 7408 的事件表达,都是 class_of_information_representation 的显式子类型。

4.8.4.2 表达

4.8.4.2.1 符号和模式

表示是把符号和模式作为信息来使用。一个符号是一个 possible_individual 角色,即时空扩展。符号可以是任何个体,可以表示任何 thing。图 88 显示了符号的表示模型。

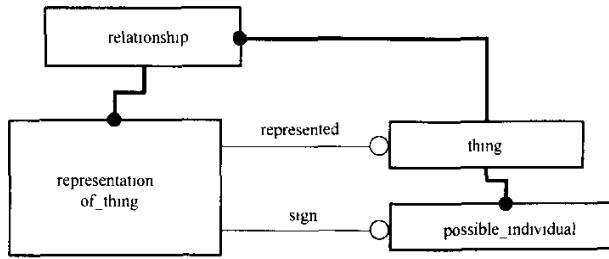


图 88 事物表示

representation_of_thing 是一个 relationship,它标明了—个 possible_individual 是其他事物的符号 (见 5.2.16.4 和图 192)。

模式是符号的类型或类,是其所标记成员的可重复性质。同样模式的成员的符号经常被用类表示同样的事物。所以模式“Jow Smith”无论何时、何地 and 如何着色都是指同一个人。图 89 显示了模式表现的模型(见 5.2.17.5 和图 193)。

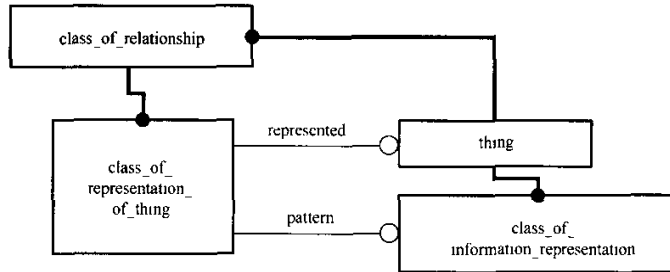


图 89 事物表现类

示例: 图 90 显示 Smith 本人作为 actual_individual #3578,通过 representation_of_thing 关系与符号 #smith 联系起来。符号 #smith 是一个非生命的 physical_object,即“Smith”模式的成员。representation_of_thing 关系是 class_of_representation_of_thing 的成员,把模式“Smith”与 #3578 联系起来。

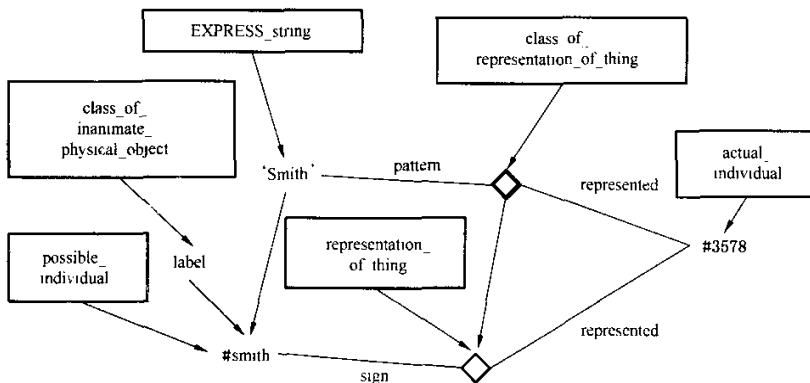


图 90 #3578 的表示

4.8.4.2.2 标识、描述和定义

identification(标识)、description(描述)和 definition(定义)是应用于文字数字、图像和声音符号的所有表示类型(见 5.2.16.1、5.2.16.2 和 5.2.16.3)。因为个体不能被定义,它们是其自身,如图 91 所示 definition 被限于 class。

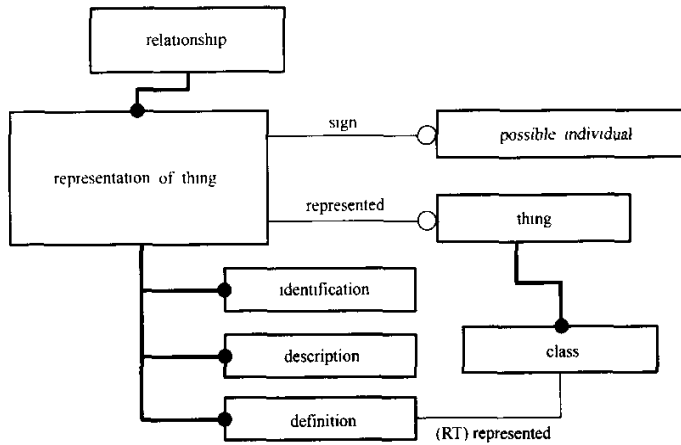


图 91 标识、描述和定义

在流程工厂相关的活动中, identification、description 和 definition 被更经常地在模式层发布, 并应用于模式的所有符号, 如图 92 所示(见 5.2.17.1、5.2.17.2 和 5.2.17.3)。

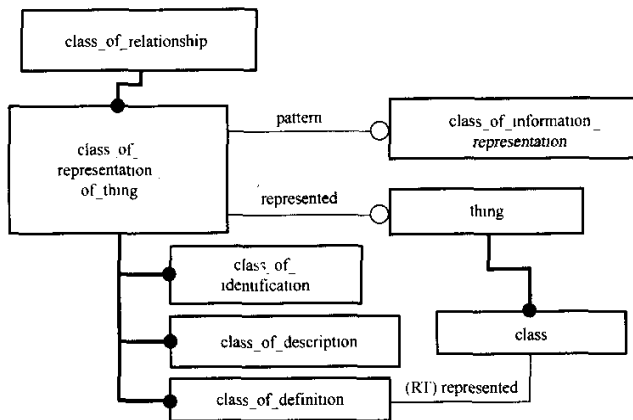


图 92 标识、描述和定义类

示例: *arranged_individual* “我的泵”被给定了一个 *identification* 模式“AC-1234”。如图 93 所示, *identification* 符号“AC-1234”出现在泵的“铭牌”上。这个组合关系所表明的“铭牌”是泵的部件没有显示。

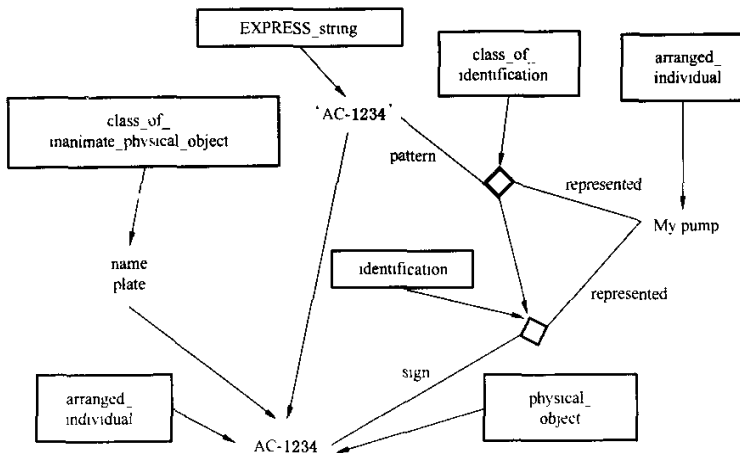


图 93 泵标识

4.8.4.2.3 表达的使用和责任

使用一定的符号和模式表现特殊的事物是自由决定的,并且可能受限于一特定的人和组织。同样的模式可以被不同的人和组织用于表示不同的事物,并且集中表示模式可以被不同的组织指派和使用于某个特定的事物。

本部分分析了广泛使用所带来的责任。使用责任是为了指明相关的人和组织,它们就使用符号和模式表示事物进行决策(见 5.2.16.5 和图 192)。使用意味一个组织或人在它们的活动中使用该表示(见 5.2.16.6 和图 192)。

图 94 中显示了表示的使用和责任的模型。

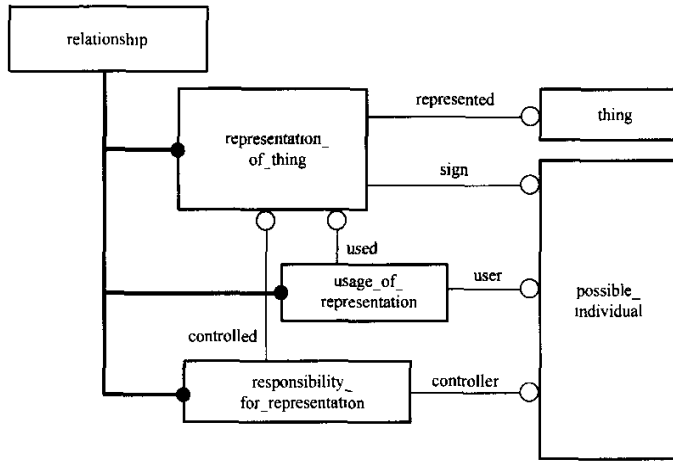


图 94 表示的使用和控制

在流程工厂相关的活动中,表示的使用和责任被更经常地在模式层发布,并应用于模式的所有符号,如图 95 所示(见 5.2.17.7,5.2.17.8 和图 193)。

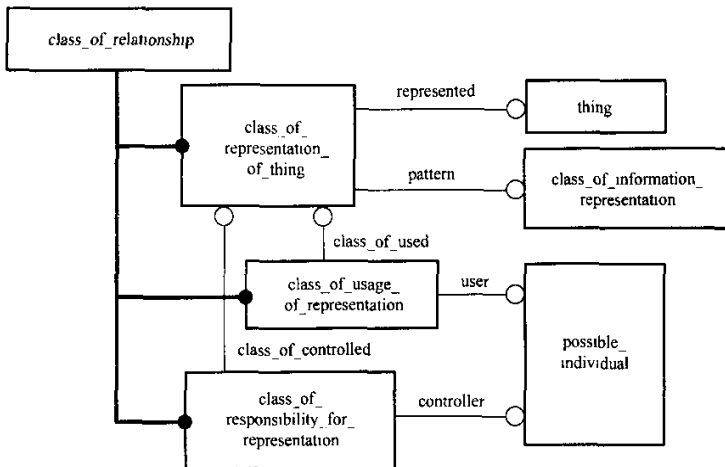


图 95 表示类的使用和控制

示例：图 96 显示了 identification 模式“AC-1234”用于“XYZ 公司”发布给“ABC 石油公司使用的“我的泵”。

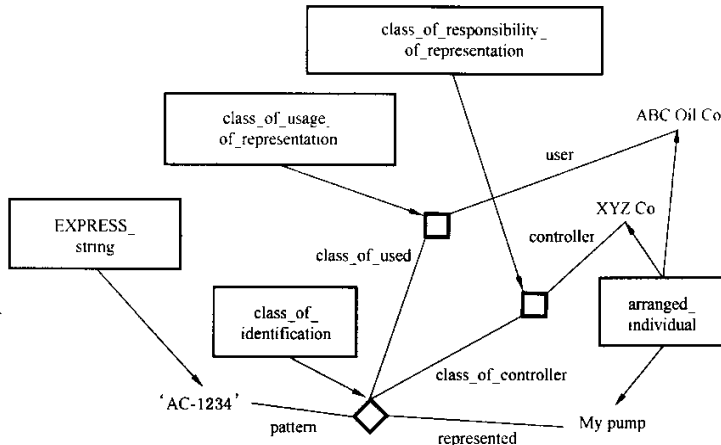


图 96 XYZ 公司产品标识

4.8.4.2.4 模式类

模式可以被进一步地抽象化,以此来定义用成员模式表示事物类型的规则和约束成员模式相组合的规则(见 5.2.19 和图 195)。模式类用 class_of_class_of_information_representation 定义,如图 97 所示,并给出了 language(语言)、representation_form(表示形式)和 document_definition(文档定义)的显示子类型(见 5.2.19.9,5.2.19.10 和 5.2.19.11)。

示例 1: 用于工程数据表的模板是一个 document_definition。

示例 2: “十六进制”是一个 representation_form,它是“文字”的特殊化,也是一个 representation_form。

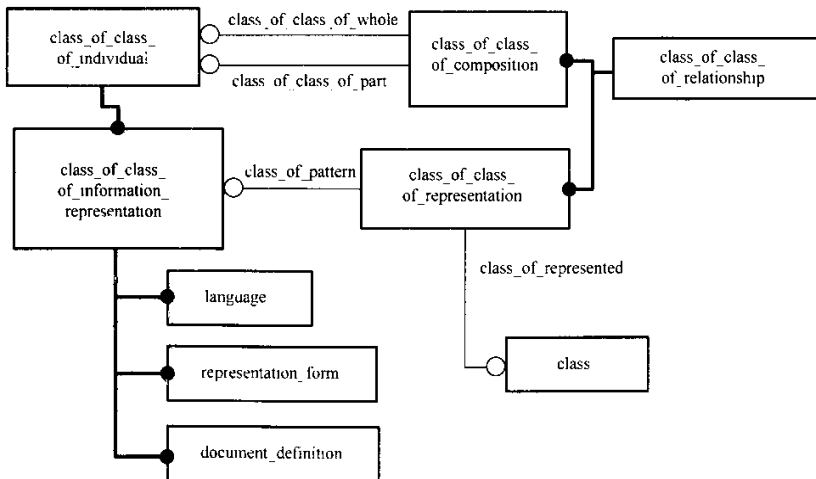


图 97 信息表示类的类

示例 3: 图 98 表示了一个 GB/T 16656.21 文件包含了一个表示 functional_physical_object P101 的记录。“P21”文件是一个 representation_form,由“P21 记录”部件。

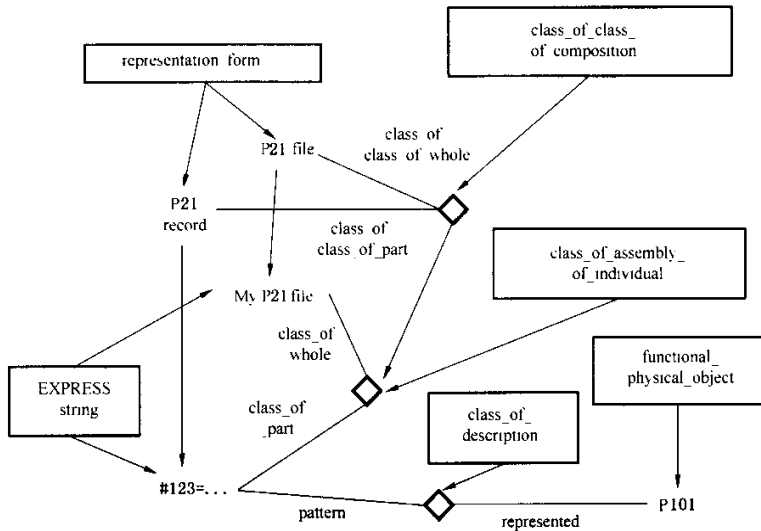


图 98 P21 文件表示

人员和组织可以使用和(或)控制某些模式类来表示一定的事物类型,其模型如图 99 所示。为此目的定义了 class_of_class_of_relationship 的两个子类型(见 5.2.19.7 和 5.2.19.8),class_of_class_of_usage_of_representation 和 class_of_class_of_responsibility_for_representation。

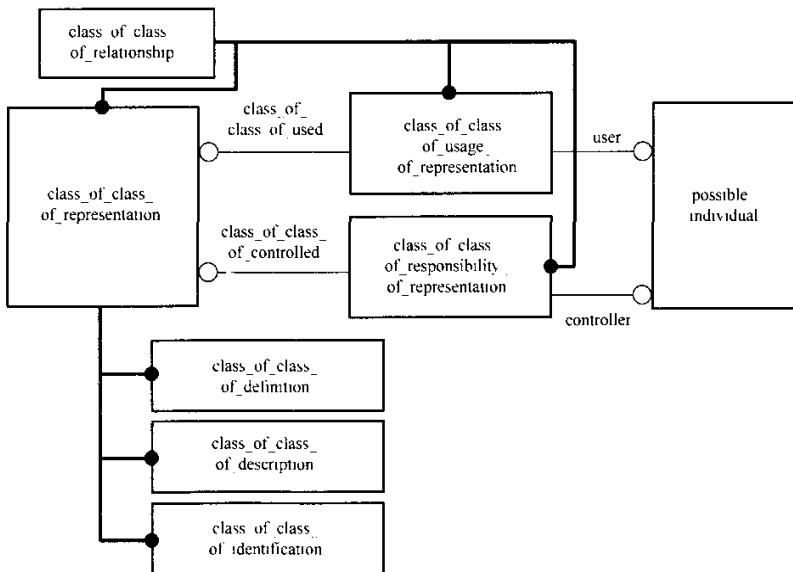


图 99 表示使用和责任类的类

示例 1:“XYZ 公司”设计和维护的泵数据表格式是一个描述“XYZ 公司”设计的各种类型泵的 document_definition。“24 类型的泵”在“24 类型泵数据表”中描述,它是一个 class_of_information_representation。注意:“24 类型的泵数据表”描述了泵的类型,而不是任何特定的泵成员。图 100 中例子没有描述数据表的类型。

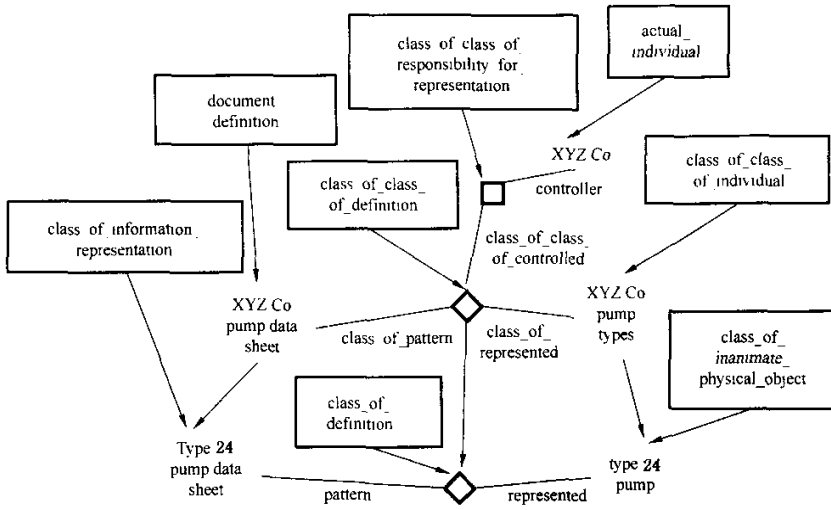


图 100 XYZ 公司泵数据表

4.8.4.2.5 名字空间模式

一些命名规范规定了模式的使用,模式像标识符一样有不变的后缀和前缀(见 5.2.20 和图 196)。这样的模式类可以用 class_of_namespace 定义(见 5.2.20.2)。class_of_namespace 把一个模式类约束到有一个不变的部件,或者是最左部件和最右部件(见 5.2.20.1 和 5.2.20.3),其模型如图 101 所示。

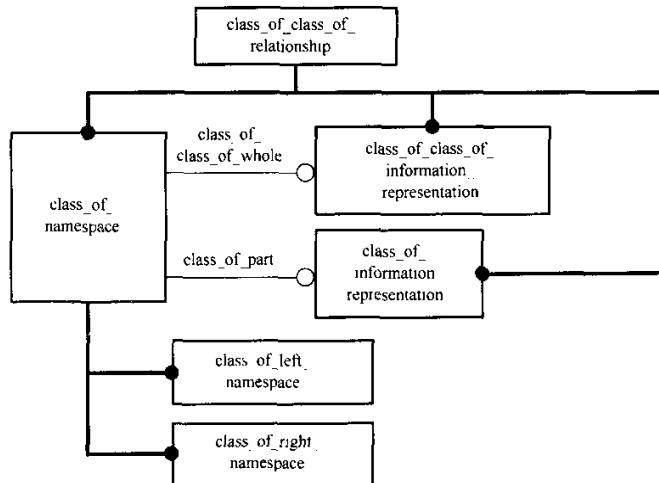


图 101 名字空间类

示例 1: 如图 102 所示容器“V1”的管嘴标识符集合是“V1;”后面加一个管嘴标识符。“V1”的标识符“V1;*”是 class_of_class_of_information_representation。这个 class_of_left_namespace 指明“V1;*”的所有成员都是“V1;”起头后面带一个管嘴的标识符,如“N1”、“N2”、“N3”和“N4”等。

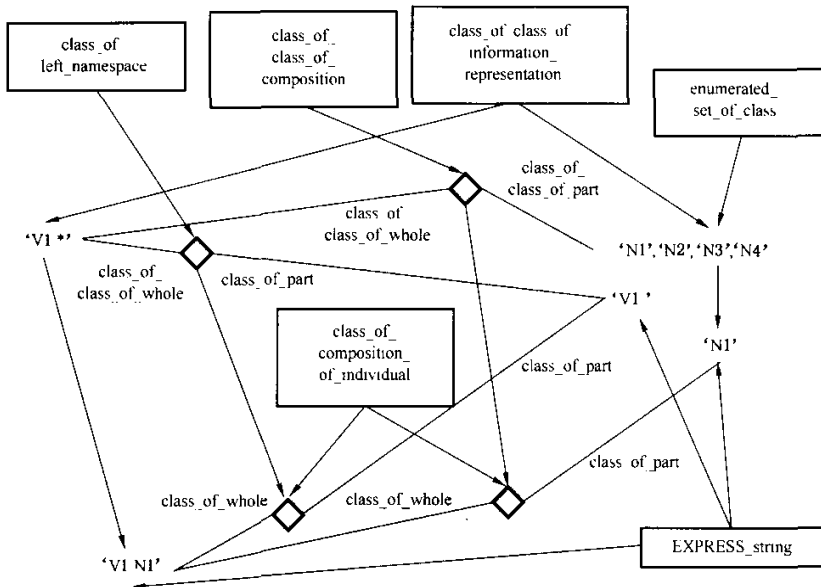


图 102 容器 V1 的管嘴名字空间

示例 2: 图 103 中模式“V1: *”用于容器“V1”管嘴的标识符。一个特定的管嘴 #4643 被标识为字符串模式“V1:N1”。

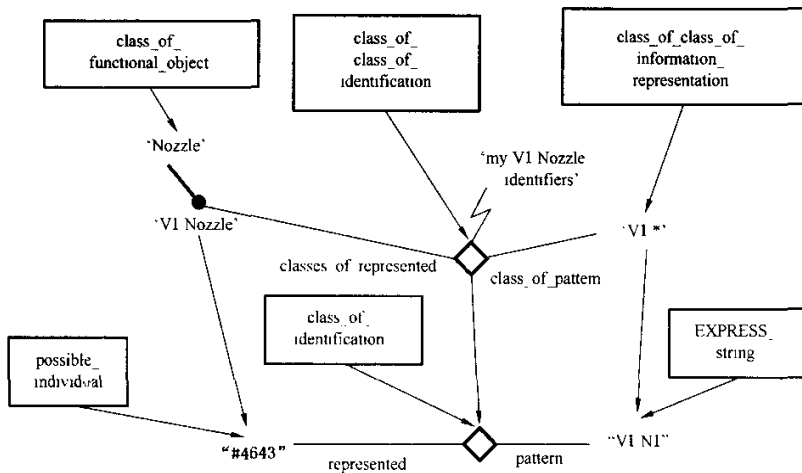


图 103 容器 V1 管嘴 N1 的标识

4.8.4.2.6 统一资源地址

本部分中,统一资源地址被作为一个文档特定产品的地址标识符。

注: URL 在 IETF RTC2396 中定义。

示例: 图 104 中,“X”是“Y”的一个产品,位号 P101 的最终 P&ID,存储在 URL‘http://www.designco.com/plant/pid/p101’上。“X”是一个 functional_physical_object,因为产品的特定的材料是不重要的。

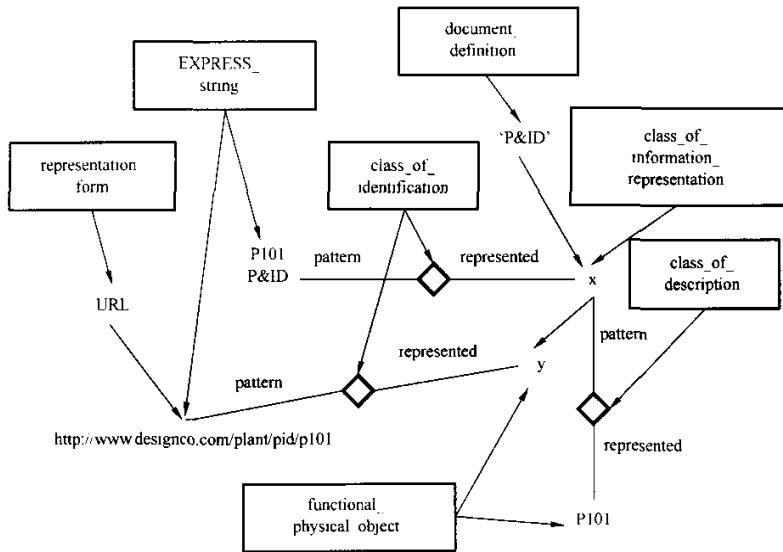


图 104 P101 的 P&ID 的 URL

4.8.4.3 特性

4.8.4.3.1 特性和特性类

property(特性)是一个 class_of_individual,其成员个体有同样的质量和特性的度数和数量(见 5.2.26和图 202)。质量和特性的类型是使用 class_of_property(特性类)定义(见 5.2.27 和图 203)。class_of_property 区分了特性连续域和特性的枚举集。图 105 显示了特性实体类型。

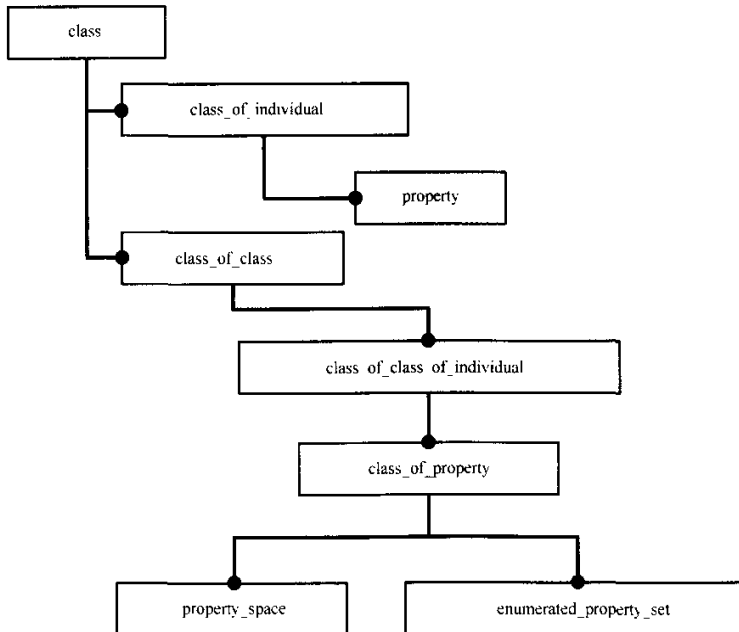


图 105 特性和特性类

示例：图 106 显示“21 ℃”是一个 property。热度的质量“温度”是 class_of_property。possible_individual “A”和“B”都是“温度”类的“21 ℃”。

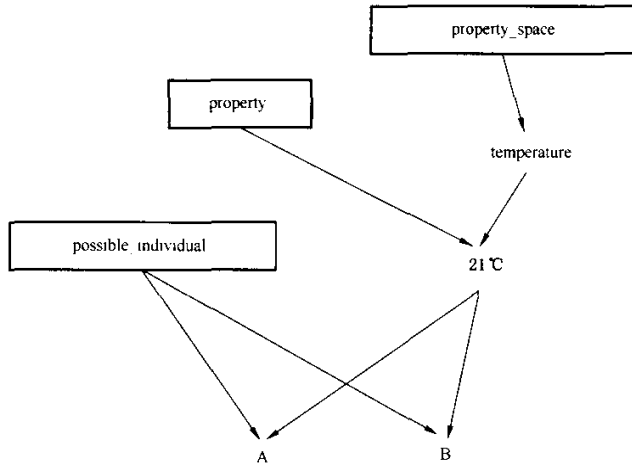


图 106 温度特性

4.8.4.3.2 特性量化

一个 property 由 functional_mapping(功能映射)关系量化成特定数量(见 5.2.26.5 和图 202)。映射所使用的 scale(比例)是一个 class_of_isomorphic_functional_mapping(见 5.2.15 和图 191)。图 107 显示了 property_quantification(特性量化)模型。property_quantification 是一个 functional_mapping 关系,指明了 arithmetic_number(算术数量)所映射的 property。4.8.5.1 中描述的 arithmetic_number 包括整数和实数(见 5.2.5.1 和图 181)。

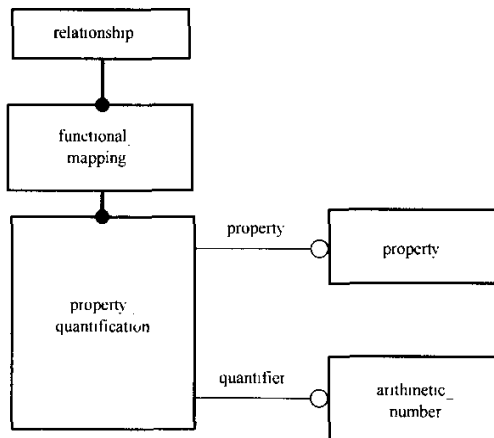


图 107 特性量化

property_quantification 关系可以依据 scale 指明量化的单位。scale 是一个 class_of_functional_mapping(功能映射类),其成员把 property_space(特性空间)域的成员映射到 number_space(数量空间)上域的成员上,如图 108 所示(见 5.2.28 和图 204)。一 property_space 是一个连续的特性(见 5.2.27.7 和图 203)。一个 number_space 是一个连续的数量(见 5.2.5.10 和图 181)。real_number(实数)和 integer_number(整数)都是 number_space。

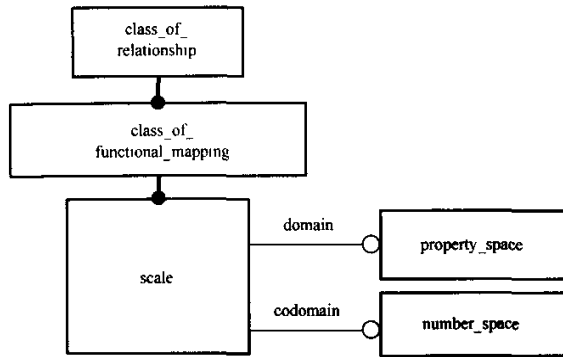


图 108 比例

例子, 图 109 显示了“21.0 °C”的温度映射到“摄氏度”比例值的数量 21.0 上。

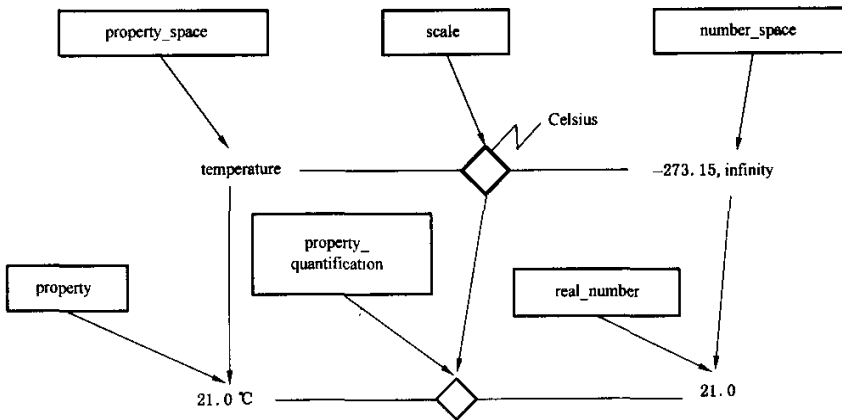


图 109 21.0 °C 的量化

4.8.4.3.3 非直接特性

indirect_property(非直接属性)是把一个 possible_individual 连接到一个 property 的连接关系(见 5.2.26.3 和图 202)。如图 110 所示, class_of_indirect_property(非直接特性类)给出了 indirect_property 的性质(见 5.2.26.1)。

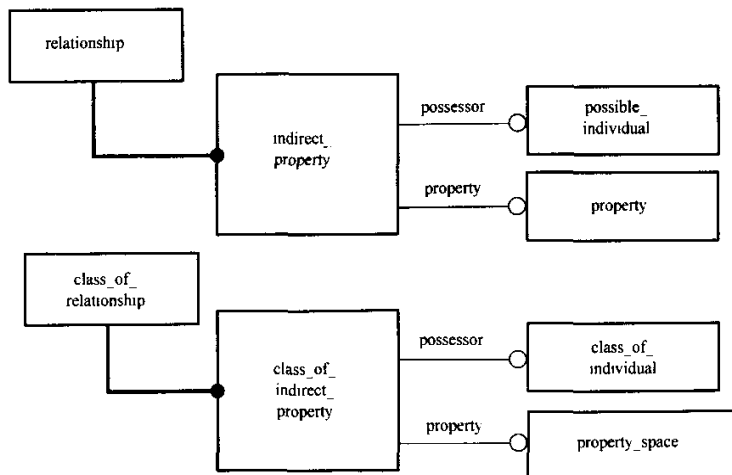


图 110 非直接特性模型

在本部分中,在不可能使用一个特性进行简单分类时,就应当使用 indirect_property。当特性是从 possible_individual 或其特定部件的特性中起源和考虑的时,这种情况就会出现。

示例 1:“压力下降”是一个 indirect_property。“压力下降”不是一个 class_of_property,因为它的成员资格与“压力”的成员资格无法区分开来。图 111 给出节流阀 10 bar 压降的实例数据。#S1 为运行节流阀的一个临时部件。

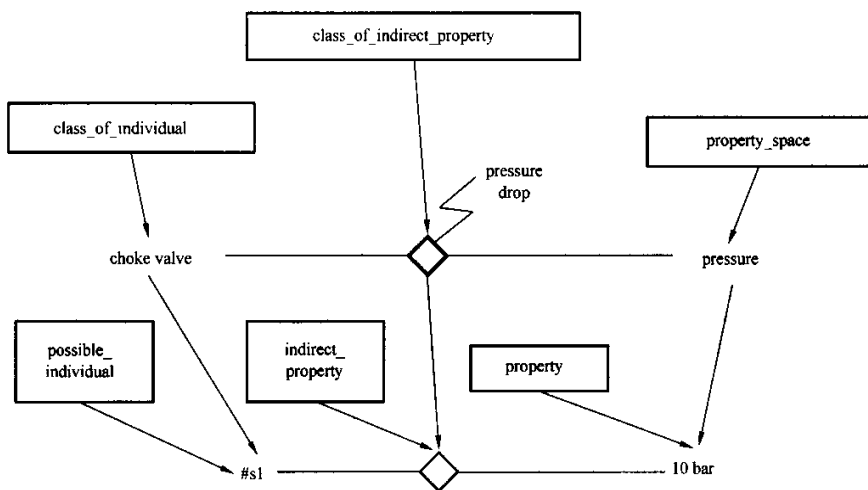


图 111 节流阀压力下降

示例 2: 图 112 为一台额定工作压力低于 27 bar 的罐 #C1。依据 27 bar 压力 property,“最大允许工作压力”是一 indirect_property。“最大允许工作压力”定义为 class-of-indirect-property。

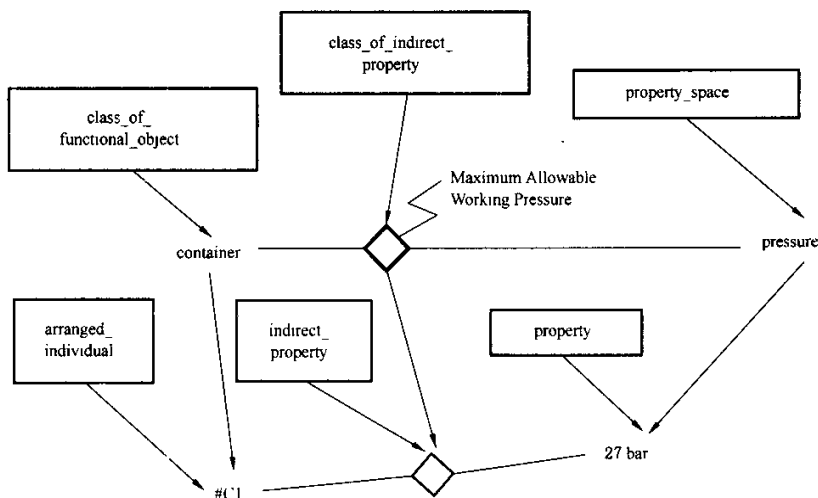


图 112 最大允许工作压力

4.8.4.3.4 特性比较

图 113 给出的 comparison_of_property(特性比较)是一个依据数量指明两个特性顺序的 relationship(见 5.2.26.2 和图 202)。

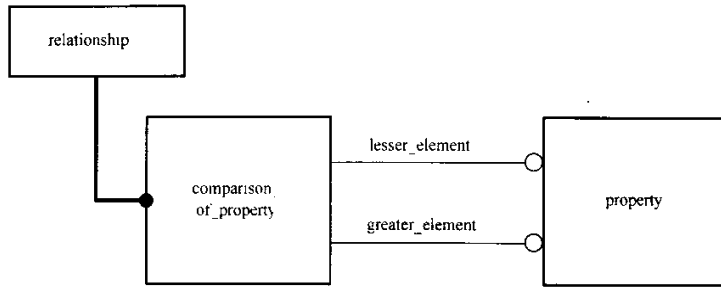


图 113 特性比较

示例：图 114 显示了 property“A”比 property“B”大。

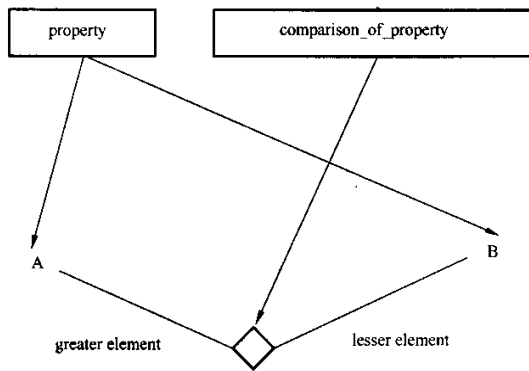


图 114 两个特性比较

4.8.4.3.5 一维特性空间

property_space 是带有数量特性连续统的一个 class_of_property 类型。在本部分中定义了两个与一维特性有关的子类型,如图 115 所示(见图 203)。一个 single_property_dimension(单个特性尺寸)是一个单独的完整特性连续统(见 5.2.27.8)。一个 property_range(特性值域)也是一个单独的连续统,但有上界和下界(见 5.2.27.6)。

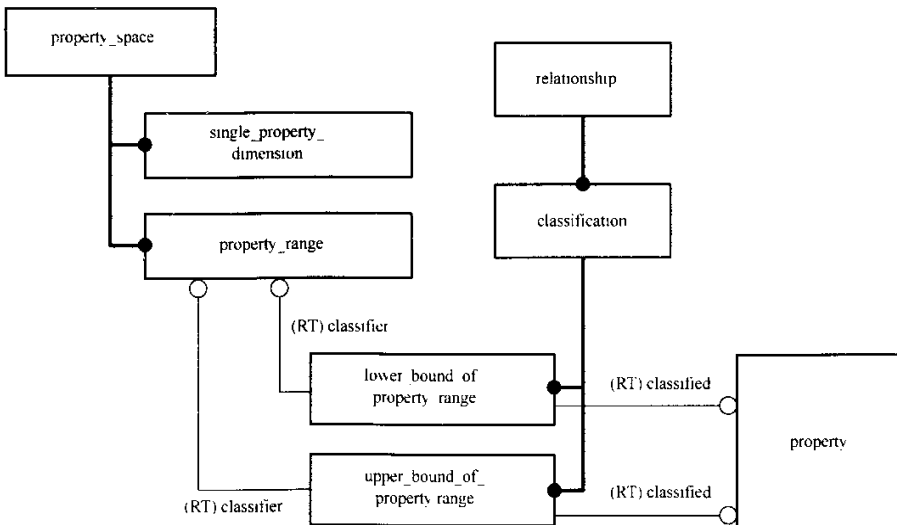


图 115 特性空间

示例 1：“温度”是一个 single_property_dimension。20 ℃～40 ℃之间的温度值域是一个 property_range，如图 116 所示。

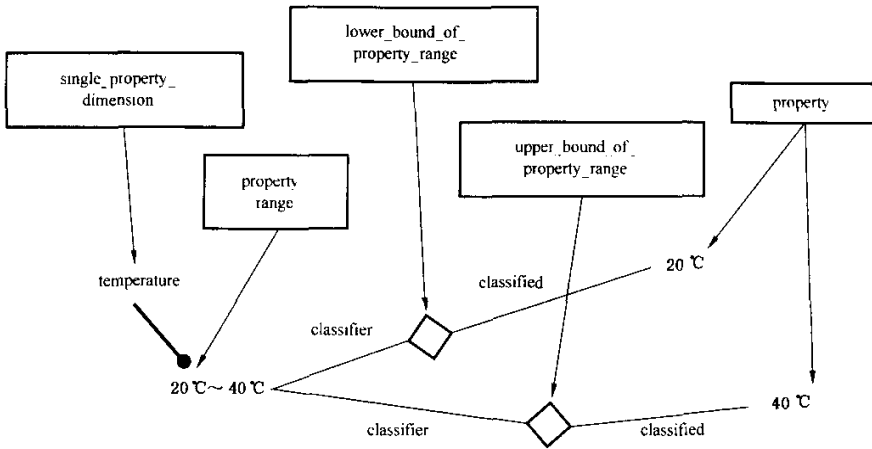


图 116 温度范围 20 ℃～40 ℃

property_range 可以被用于描述个体类成员资格的规则,约束成员的特性。

示例 2：一个“A类封条”有一个“工作温度”范围,即 20 ℃～40 ℃。#S1,一个特定封条的临时部件的“工作温度”是 24 ℃,如图 117 所示。

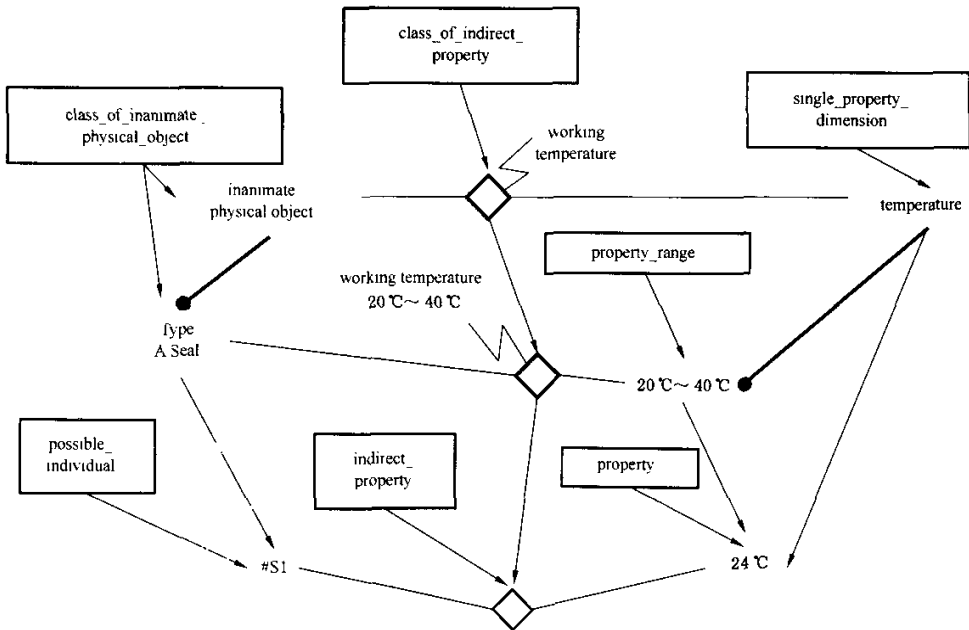


图 117 A 类型封条工作压力范围

4.8.4.3.6 多维特性

multidimensional_property(多维特性)是特性的一个有序表。一个 multidimensional_property_space(多维特性空间)对 multidimensional_property 的实例进行了分类,如图 118 所示(见 5.2.27.5 和图 203)。

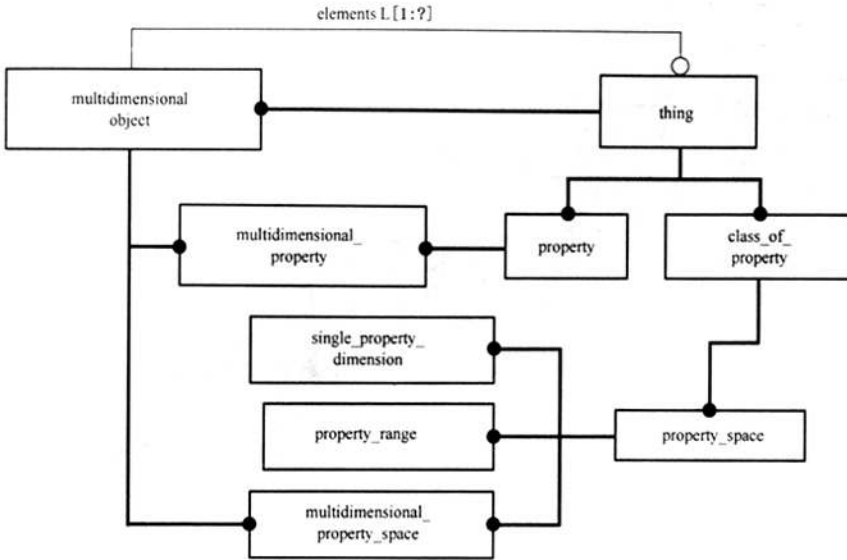


图 118 多维特性模型

示例 1: 一台泵的流量压头特性曲线是一 multidimensional_object(多维对象),由流量 Q 和压头 H 组成。每一对属性 Q_i 和 H_i 都是一个多维特性,其中 Q_i 是特定的流量, H_i 是特定的压头。 Q 和 H 属性对是连续的,即 (Q, H) 是一 multidimensional_property_space。图 119 所示的 $[Q, H]_i$ 是一特定泵的 QH 特性曲线,也是 multidimensional_property_space $[Q, H]$ 。

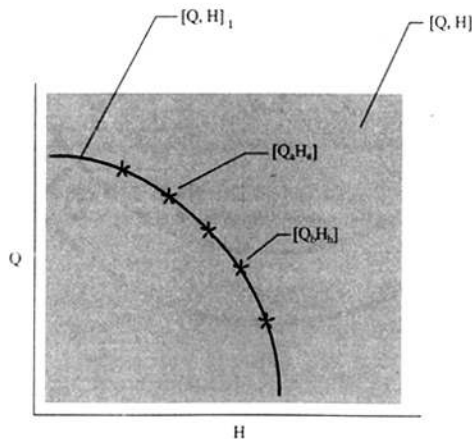


图 119 一个泵物流特性

示例 2: multidimensional_property $[Q_i, H_i]$ 和 $[Q_i, H_i]$ 是有序的特性对。它们是 multidimensional_property_space $[Q, H]$ 的成员,它是 multidimensional_property_space $[Q, H]$ 的子集和特殊化。接下来, $[Q, H]$ 是 single_dimensional_property(单维特性) Q 和 H 构成的有序对,如图 120 所示。

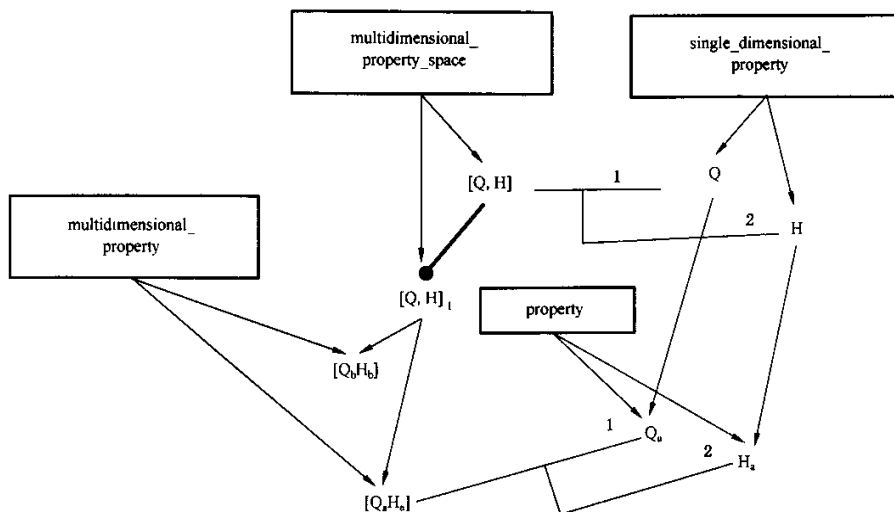


图 120 泵物流特征

示例 3：图 121 显示了一个运行中的“24 类型泵”#pz87 的临时部件，它的流速是 Q_i ，头是 H_i 。“24 类型泵”有一个操作特性即 $[Q, H]_i$ ，意味着任何运行中的“24 类型泵”的 QH 都是 $[Q, H]_i$ 的成员。

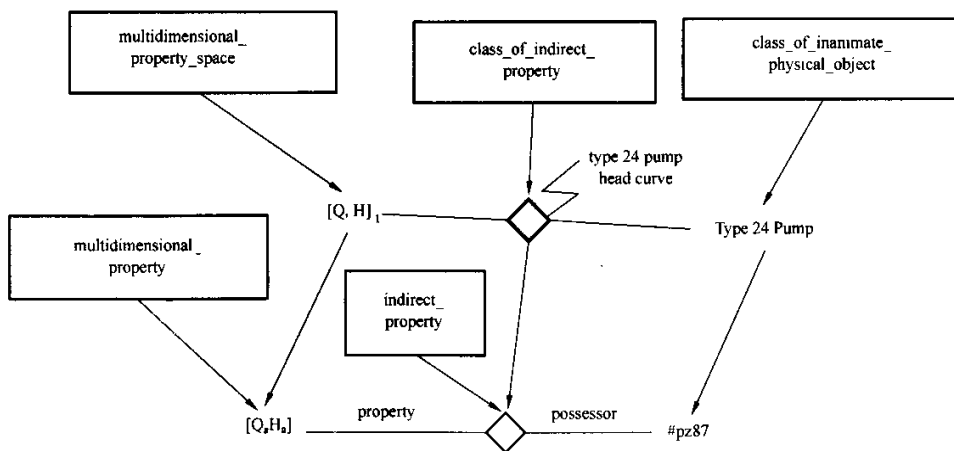


图 121 24 类泵头物流特性

4.8.4.3.7 位置和坐标

位置也是一个 multidimensional_property。从广义上说三维空间中的位置可以用对三个点的引用来定义。这三个引用构成了多维特性的尺寸。

位置特性可以被映射到 multidimensional_number(多维数量)上。一个特定的位置可以有多个映射，分别对应于不同的 coordinate_system(坐标系)。

coordinate_system 是一个 multidimensional_scale(多维刻度)，它把一个 property_space 映射到一

个 multidimensional_number_space(多维数量空间)(见 5.2.28.2 和图 204),其模型如图 122 所示。coordinate_system 是一个由它的三个成员定义的三维空间。

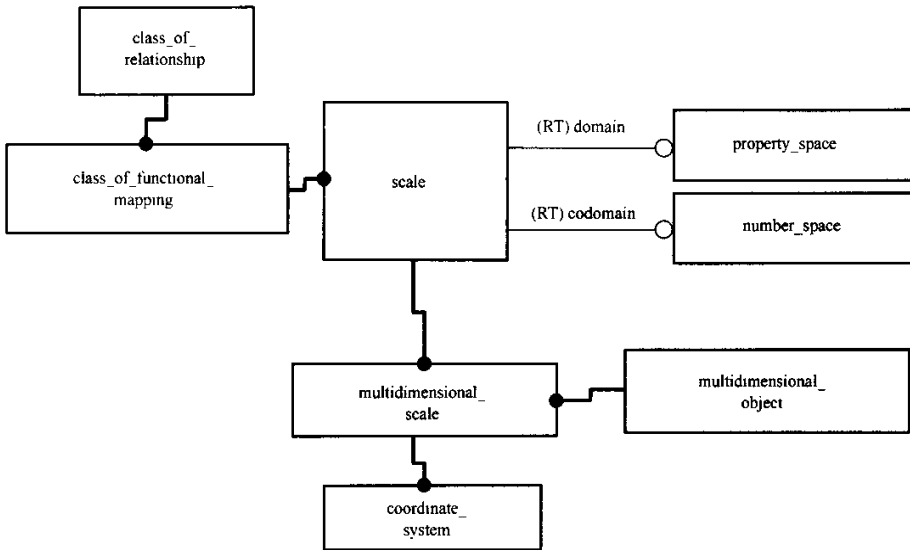


图 122 坐标系统模型

示例：图 123 显示了 possible_individual“A”所处的位置被描述成“CS21” coordinate_system 中的 [2;3;4]。

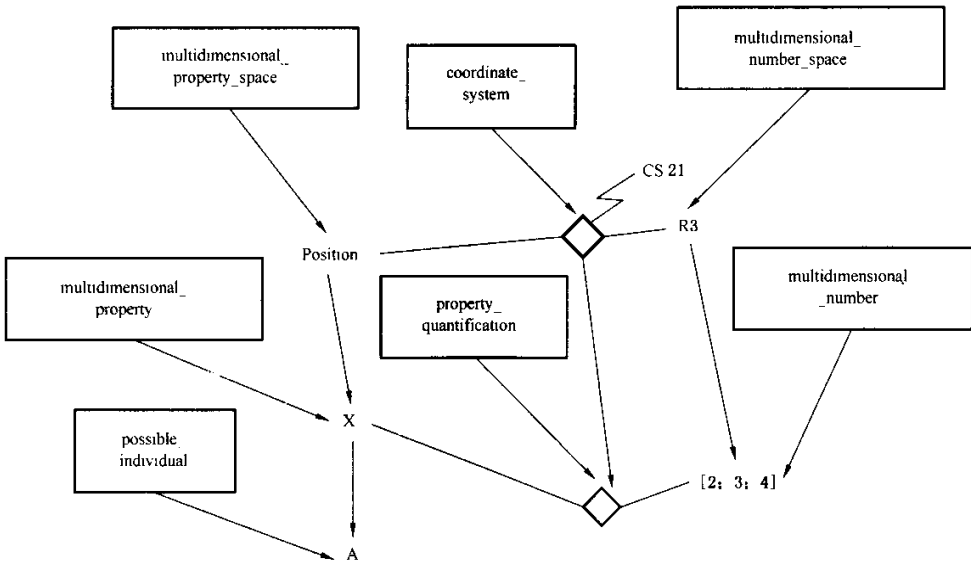


图 123 坐标系统 CS21

4.8.4.4 状态和状态类

possible_individual 可以用 status(状态)分类(见 5.2.7.13 和图 183)。一个状态反映了 possible_individual 的条件或状态,它是不能量化和排序的。status 与 property 的区别在于,状态通常是描述型的,并不能通过映射到明显有序的数量上进行量化。status 的不同类型可以通过 class_of_status(状态类)进行区分(见 5.2.7.11)。见图 124。

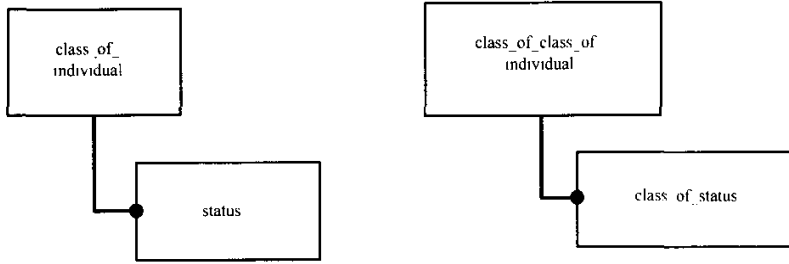


图 124 状态和状态类

示例 1：“有凹痕的”、“挖凿的”、“刮的”和“凹的”是 status 的例子，它们是 class_of_status “表面条件”的成员。图 125 显示了这样的数据。“管子外部”#rc359 是一个“凹的”status 种类中的 actual_individual。

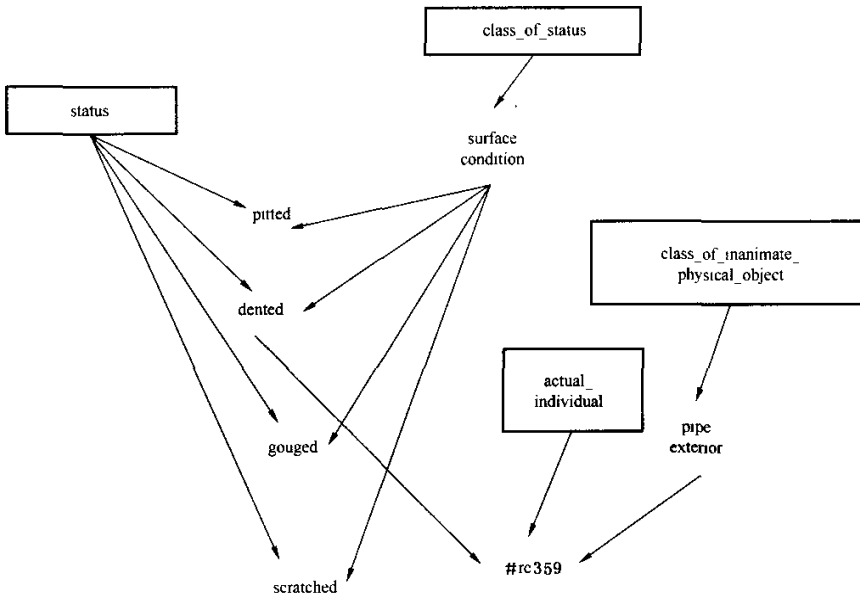


图 125 表面条件状态

示例 2：“磨损程度”，例如“没有磨损”、“轻微磨损”、“非常磨损”和“磨穿”就不是一个 status，因为它可以被排序并映射到一个数量刻度。

示例 3：“阀门打开程度”不是一个 status，因为它可以被排序并映射到一个数量刻度。

4.8.4.5 形状和尺寸

4.8.4.5.1 个体尺寸

有空间对称性的个体经常可以用尺寸描述。在这个意义上，一个尺寸是一个 class_of_individual，其成员有一个共同的长度，并且与它们所描述个体的边界有多个交集(见 5.2.29.6 和图 205)。图 126 显示了 individual_dimension(个体尺寸)的模型。

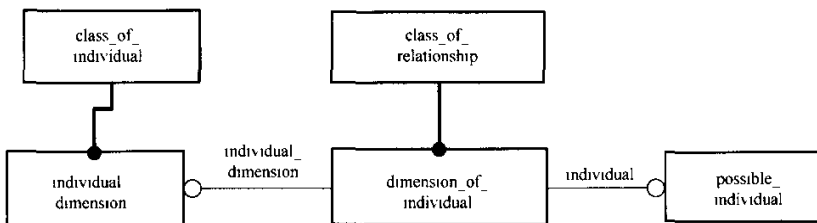


图 126 个体尺寸

individual_dimension 是一个 class_of_relationship, 其成员指明了一个 individual_dimension 是一个 possible_individual 的尺寸(见 5.2.29.4)。

示例 2: 图 127 显示了“桌子”的宽度是一个 individual_dimension。

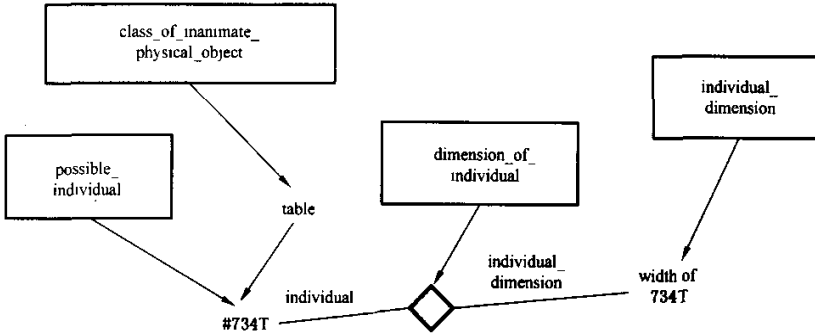


图 127 我的桌子的宽度

一个 individual_dimension 不必是它所标注的个体的部件。

示例 2: 管子的内径不是管子的部件。

4.8.4.5.2 个体尺寸的特性

individual_dimension 的成员通常有一个或多个共同的属性。图 128 中定义了一个 specialization 关系的类型(见 5.2.29.11 和图 205)。

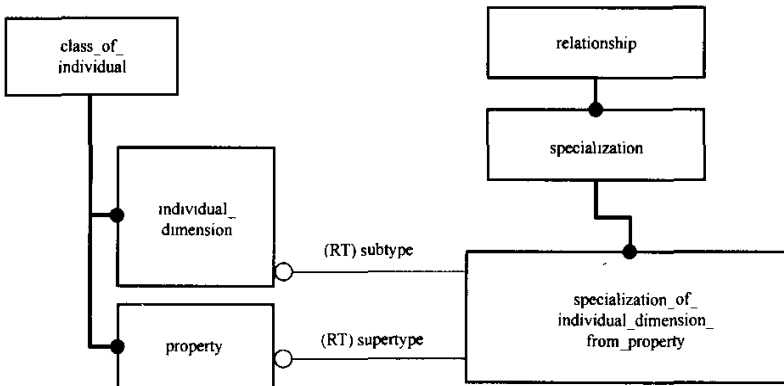


图 128 个体尺寸的特性

示例: 图 129 显示桌子的准确长度是“520 mm”。

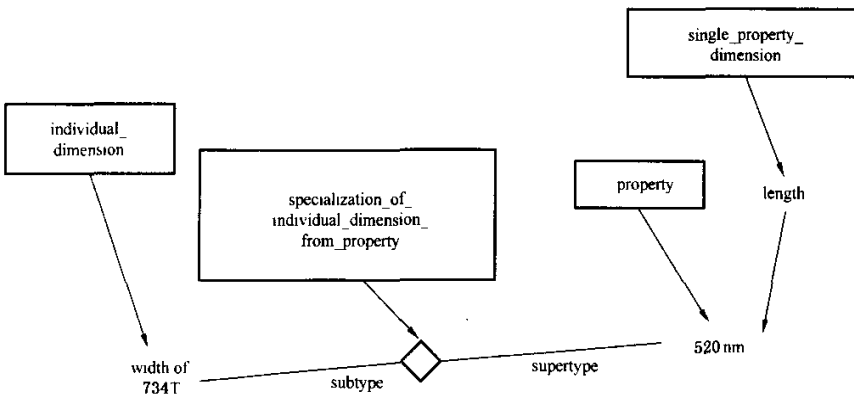


图 129 我的桌子宽度为 520 mm

4.8.4.5.3 形状

shape(形状)描述了与一个边界相关的空间位置的不变性质(见 5.2.29 和图 205)。shape 是一个 property 和一个 class_of_individual, 如图 130 所示。class_of_shape(形状类)使不同类型的 shape 如圆、长方形等得以被识别。

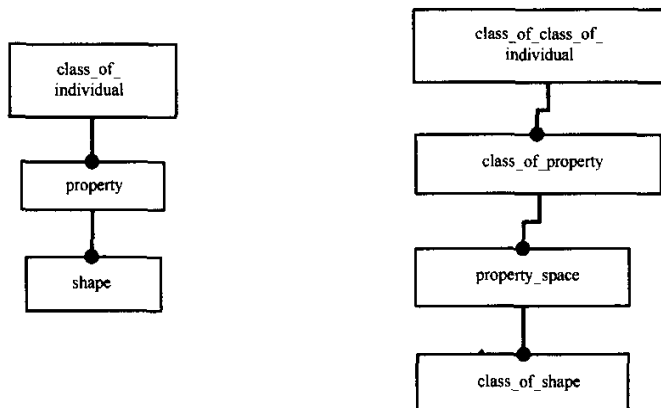


图 130 形状和形状类

示例：图 131 显示了一个经度量后为 2 cm×1 cm 的 possible_individual, 它是一个 2 cm×1 cm 的 shape, 并被归类到“长方形”中。

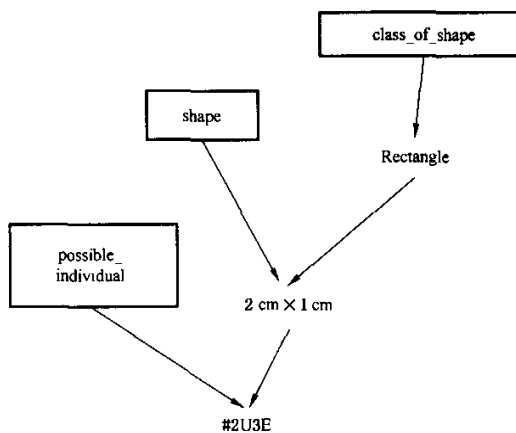


图 131 长方形形状

4.8.4.5.4 形状的尺寸

shape 也可以有尺寸, 对 shape 的所有成员是共同的。一个 shape_dimension(形状尺寸)是一个 class_of_class_of_individual(个体类的类), 其成员有一个共同的长度, 并且与它们所描述 shape 的边界有多个交集(见 5.2.29.10)。交集的类型由 class_of_shape_dimension(形状尺寸类)(见 5.2.29.3)进行区分, 其模型如图 132 所示。

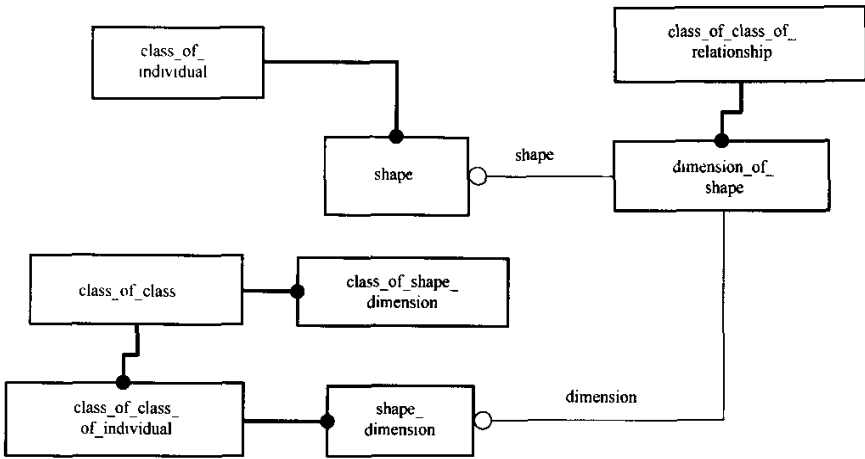


图 132 形状的尺寸

dimension_of_shape(形状的尺寸)是一个 class_of_class_of_relationship,其成员把 shape 类的成员与 shape_dimension 的成员联系起来。dimension_of_shape 可以归类 dimension_of_individual 的成员(见 5.2.29.4)。

示例:图 133 显示了“10 cm 圆”是 shape。一个“10 cm 直径”是一个“10 cm 圆”shape_dimension。“10 cm 直径”是“10 cm 圆”的成员的尺寸。

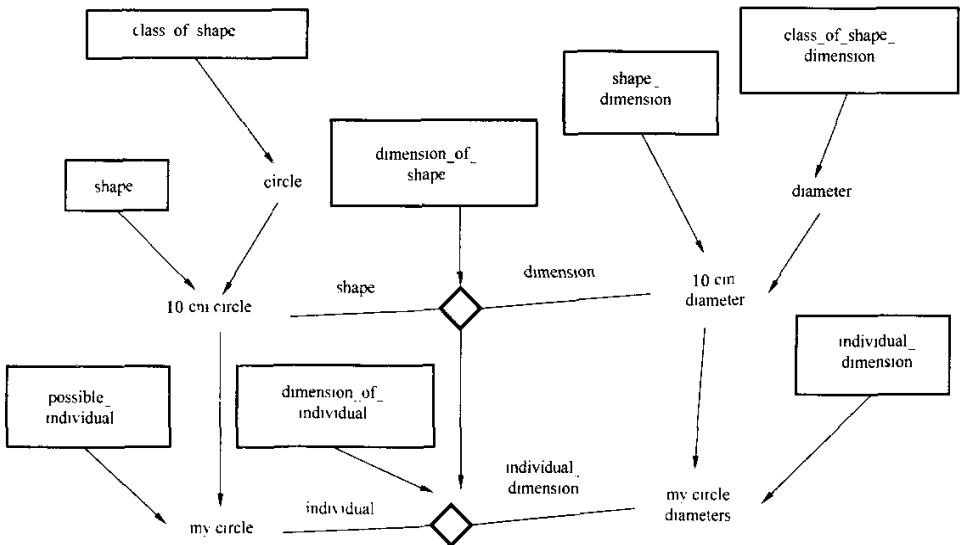


图 133 10 cm 直径圆

4.8.4.5.5 形状尺寸的特性

shape_dimension 的成员有一个或多个共同特性。图 134 定义了一个 class_of_relationship property_for_shape_dimension(形状尺寸特性关系类)来表示。

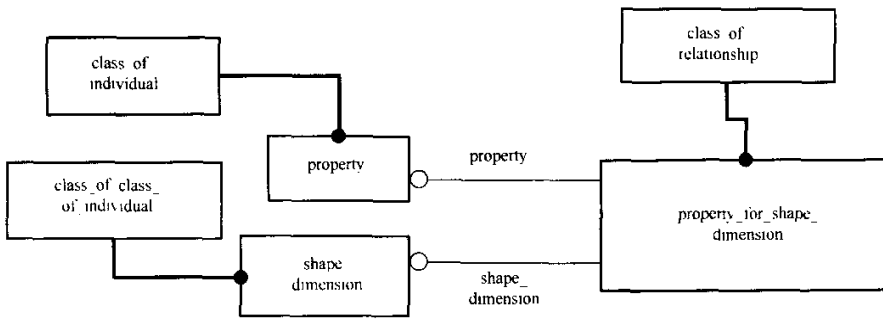


图 134 形状尺寸的特性

示例：图 135 显示了 shape_dimension “10 cm 直径”的成员都有“10 cm”的长度。

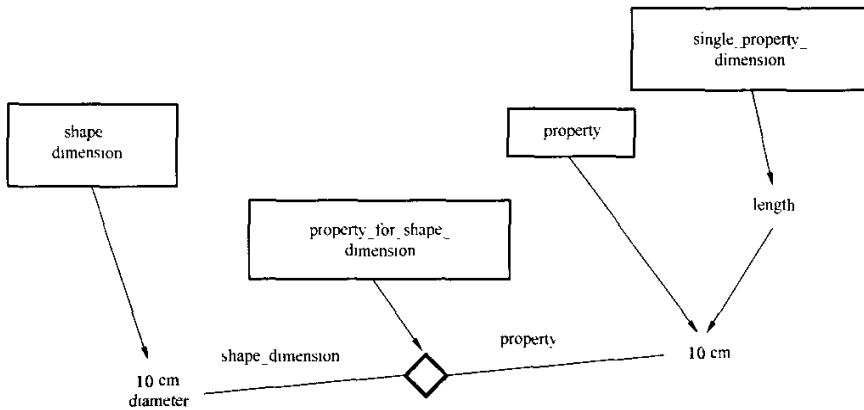


图 135 10 cm 长度直径

4.8.4.5.6 形状尺寸类

shape_dimension 可以依据尺寸相对其所标注个体的位置进行分类。图 136 显示了 class_of_shape_dimension(形状尺寸类)的定义(见 5.2.29.3 和图 205)。

示例 1：“直径”、“高度”、“长度”和“宽度”都是 class_of_dimension_for_shape(形状的尺寸类)。class_of_dimension_for_shape 使一种尺寸成为一种 shape 的特征。

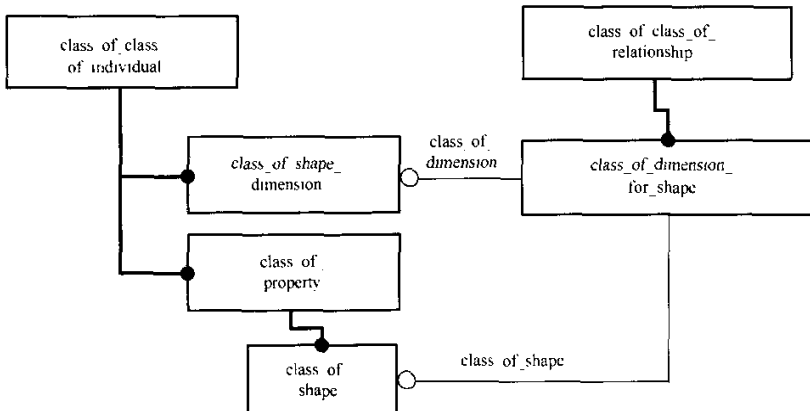


图 136 形状尺寸类

示例 2：如图 137 所示，“直径”是“圆”的尺寸。

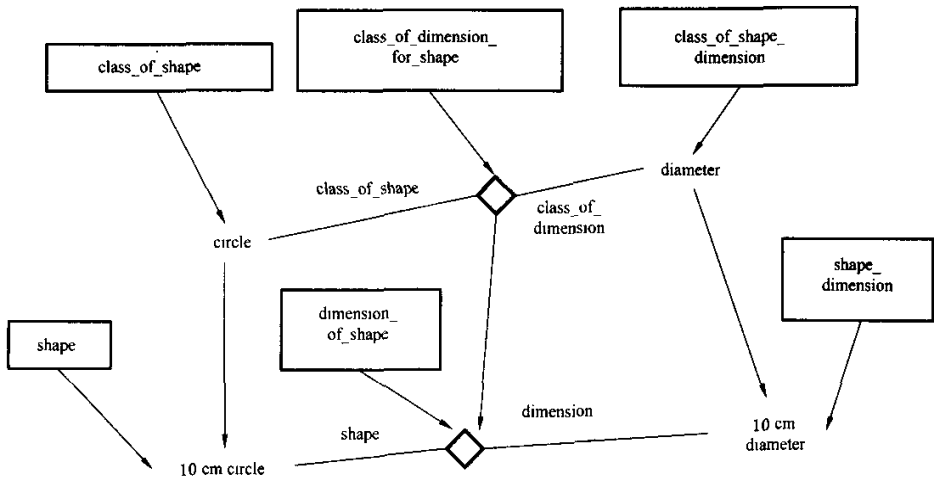


图 137 圆的直径

4.8.4.5.7 形状尺寸类的特性类

某种类型的 shape_dimension 有某种类型的 property(见 5.2.29.8 和图 205), 图 138 显示了提供支持的模型图。

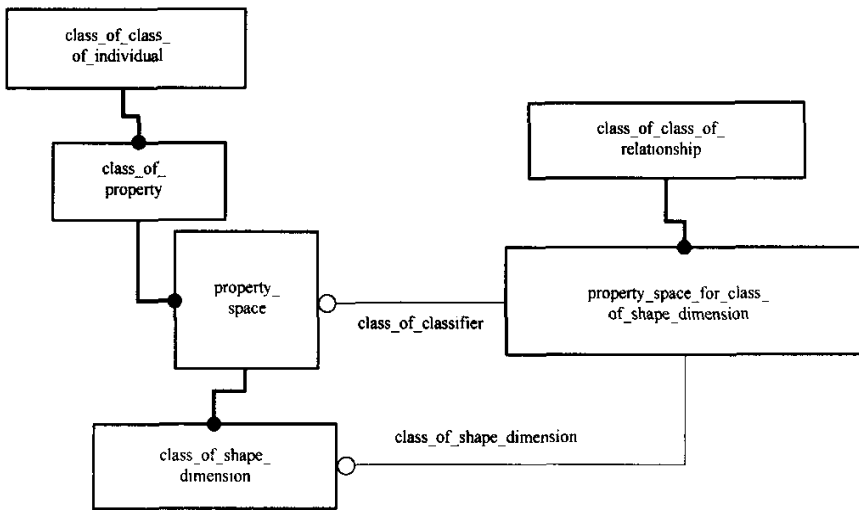


图 138 形状尺寸特性类

示例：图 139 显示了一个“直径”有一个“长度”。

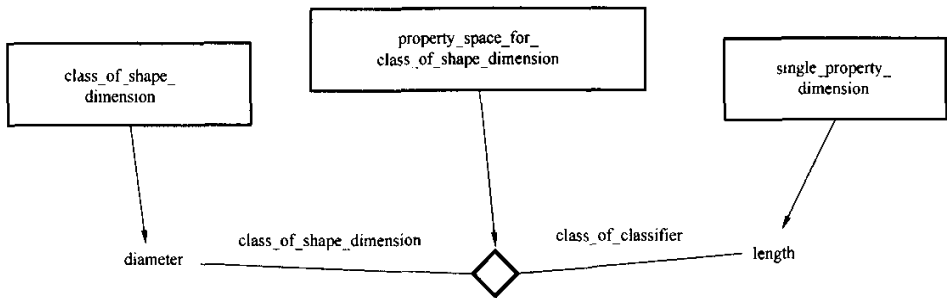


图 139 直径长度

4.8.4.6 事件和时间点类

一个 event(事件)和 point_in_time(时间点)可以被分类。event 类可以由 class_of_point_in_time (时间点类)的成员 event 决定(见 5.2.7.10 和图 183),其模型如图 140 所示。

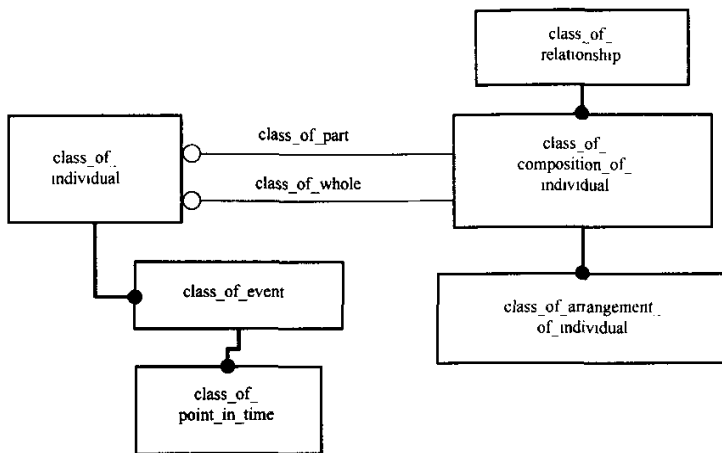


图 140 事件和时间点类

示例：图 141 显示了“起飞”是一个 class_of_event(事件类),标记了飞机从地面到空中变化。“午夜”是一个 class_of_point_in_time。“午夜起飞”是一个 class_of_event,即“起飞”的特殊化。所有的“午夜起飞”是 point_in_time“午夜”的部件。

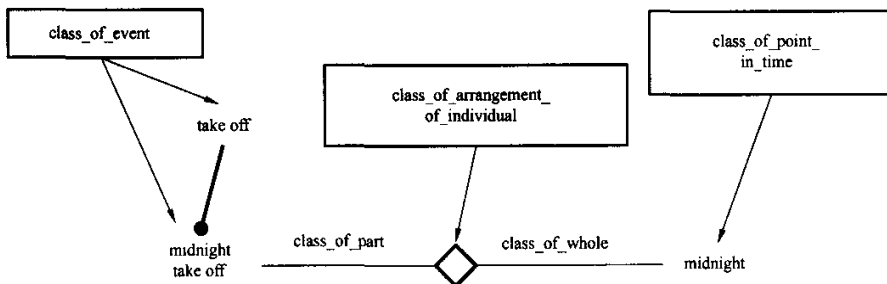


图 141 午夜起飞事件

4.8.4.7 时间段类

period_of_time(时间段)可以被分类(见 5.2.7.9 和图 183)。class_of_temporal_sequence(时间顺序类)使 period_of_time 的成员类可以被排序,其模型如图 142 所示。

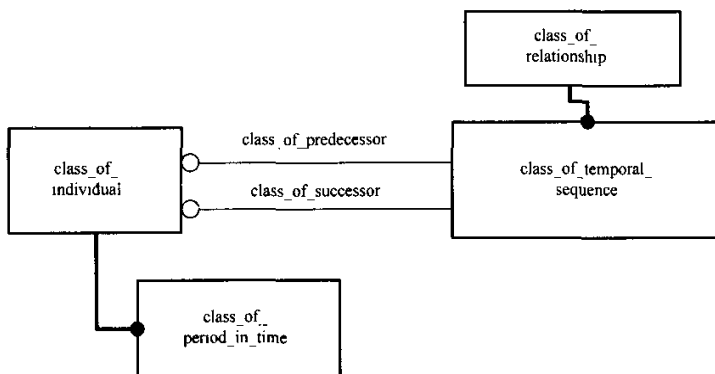


图 142 时间段类

示例：图 143 显示“六月”和“七月”是 class_of_period_in_time(时间段类)，每个“七月”都在一个“六月”后面。

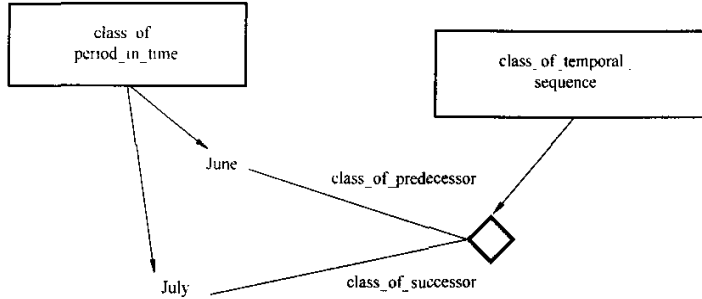


图 143 七月在六月后

4.8.4.8 角色和域

role_and_domain(角色和域)是基于在活动或关系背景中某些事物做了什么的一种 class(见 5.2.13.5 和图 189)。有些 role_and_domain 类仅仅是纯角色的，即其成员资格除了角色外不受其他条件的约束(见 5.2.13.4)。为此定义了 role(角色)的子类型。然而大多数 role_and_domain 类是一个纯 role 和一个域的交集，域限制了能够扮演该 role 的事物类型，其模型如图 144 所示。

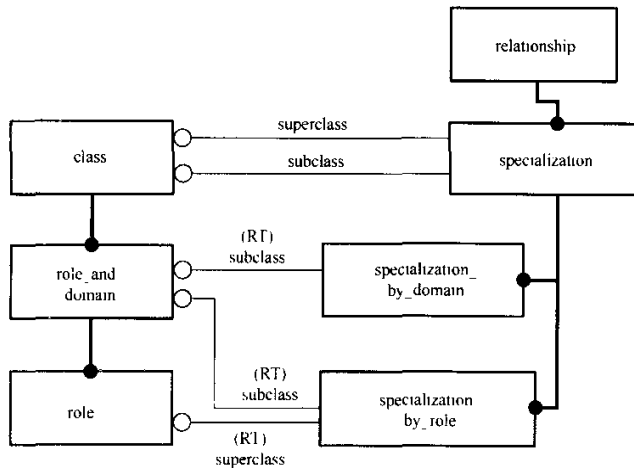


图 144 角色和域

特殊化关系 specialization_by_role(角色特殊化)使一个角色和域的角色超类能被定义(见 5.2.13.7)。specialization_by_domain(域特殊化)关系使 role_and_domain 的域超类能被定义(见 5.2.13.6)。

示例：图 145 显示了“控制者”是一个 role。“控制人”是 role_and_domain，它是 role “控制者”和域“人”的特殊化。“控制人”role_and_domain 不包含机械“控制者”。

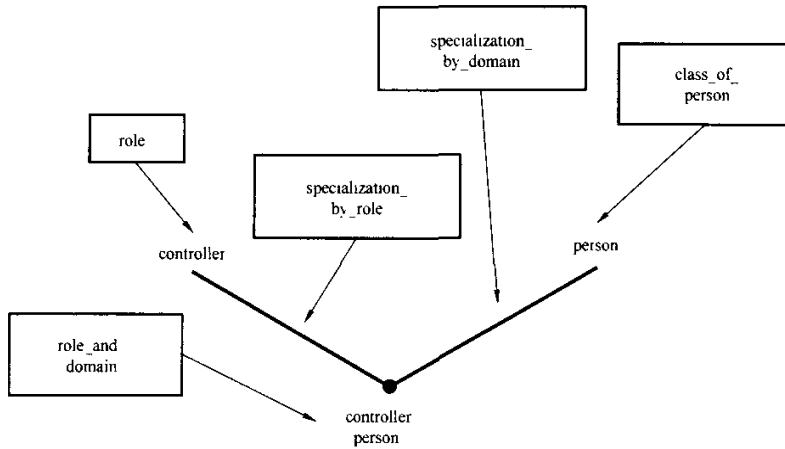


图 145 控制人的角色和域

4.8.4.8.1 预期的和可能的角色

基于其特征和特性,一些个体可以扮演特定的角色。这适合域个体和特定个体类的成员(见 5.2.24 和图 200)。图 146 显示的模型识别了这些 relationship 的类型和这种性质的 class_of_relationship。

intended_role_and_domain(预期角色和域)关系标明个体的目的或被设计的 role_and_domain,或反之。

示例 1: 一个特定的“带锥形端部的 2 m 钢棒”将预期扮演“杠杆”的角色。一个 possible_role_and_domain(可能的角色和域)关系标明了一个个体可以承担的 role_and_domain。这不是由于设计,而是因为个体有适合于角色的属性(见 5.2.24.4)。

示例 2: 一个特定“20 kg 的混凝土立方块”可能扮演“锚”的角色。

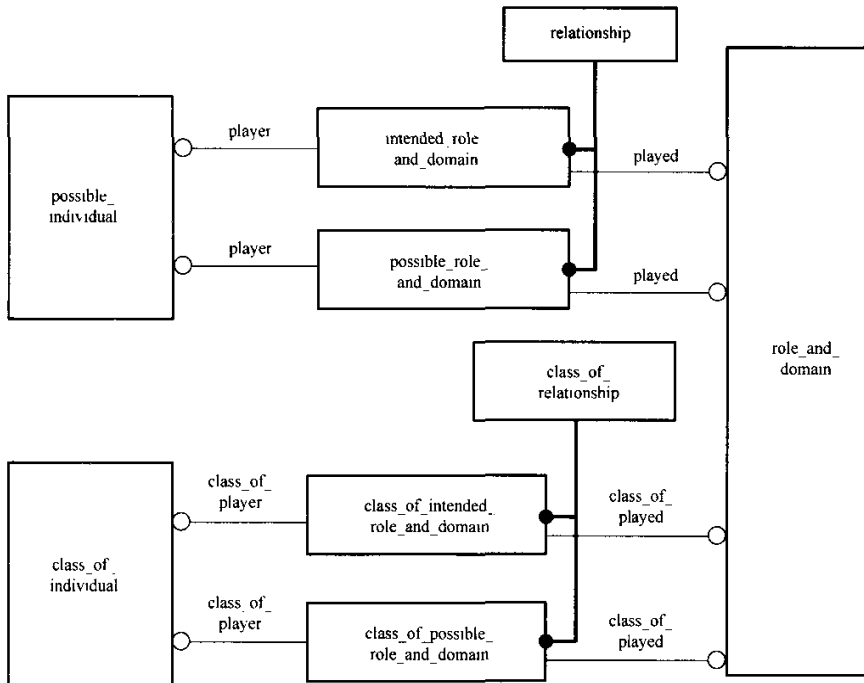


图 146 预期的和可能的角色和域

class_of_intended_role_and_domain(预期角色和域类)指明了 class_of_individual 成员被设计扮演的 role_and_domain(见 5.2.24.1)。class_of_individual 成员共同的特性和特征被有意地选择出来,使成员能履行指定的 role_and_domain。

区分 role_and_domain 类和 class_of_functional_object 类很重要。role_and_domain 类的成员包含在活动中,然而 class_of_functional_object 类的成员是有能力被包含进来。通常这些类有不同的含义,但有同样的名字。

示例 3: 图 147 显示了“泵”的成员,即 class_of_functional_object,被预期扮演“抽水机”的角色。“抽水机”是 role_and_domain “表演者”的特殊化。“抽水机”是实际事物的抽象。

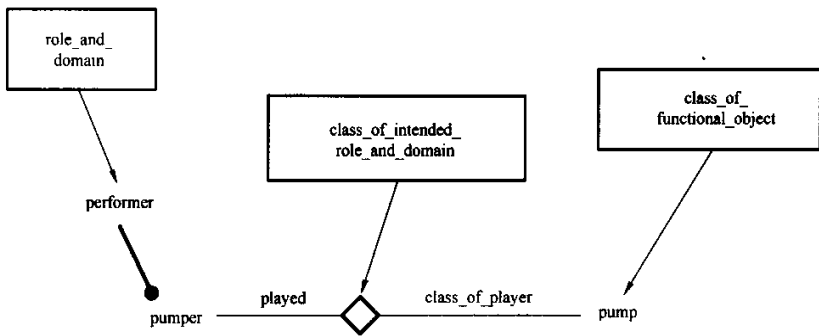


图 147 泵的预期性能角色

class_of_possible_role_and_domain 指明了 class_of_individual 成员能够扮演的 role_and_domain,尽管其设计意图不是这样(见 5.2.24.2)。

示例 5: “杯子”的成员是 class_of_functional_object,它能扮演“糖碗”的角色。

4.8.4.8.2 参与角色和域

从广义上说任何 thing 都可以是 role_and_domain 类的成员,这包括个体和抽象对象。然而,因为活动中的参与者都应当是个体,子类型 participating_role_and_domain(参与角色和域)限定了其成员是个体(见 5.2.13.3 和图 189),其模型如图 148 所示。

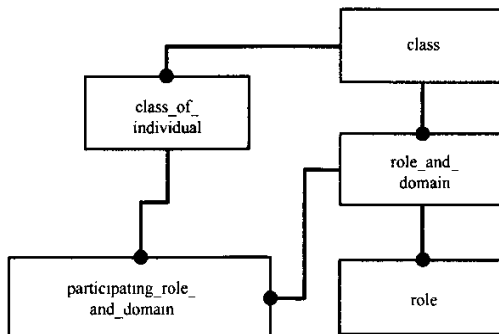


图 148 参与角色和域

4.8.4.9 活动类

活动类可以通过限定包含在活动类成员中的事物类型来定义(见 5.2.10 和图 186),其模型如图 149 所示。

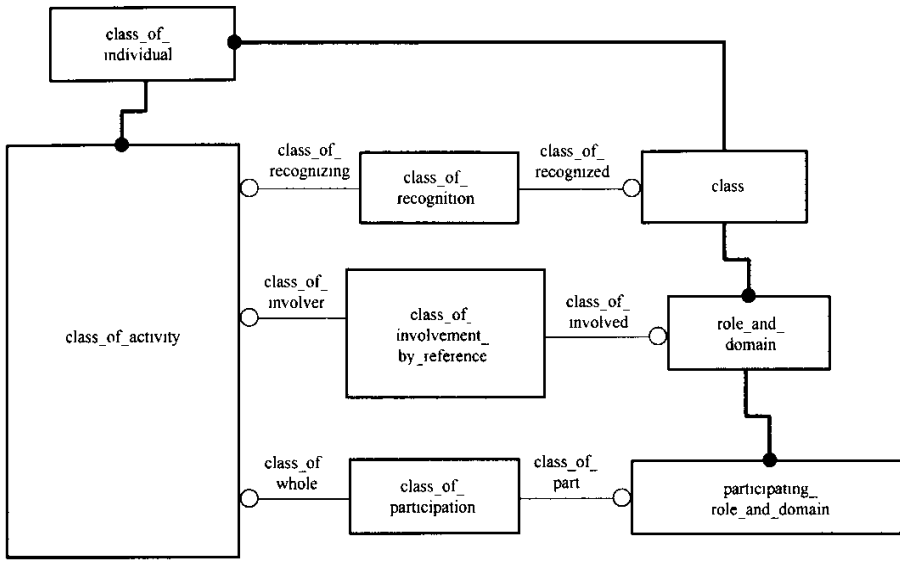


图 149 活动类

示例 1：“使用 167 类型仪表度量流体压力”是一个 class_of_activity(活动类)，并且是 class_of_activity “度量”的特殊化。class_of_participation(参与类)限制了活动类成员中的个体类型及其角色(见 5.2.10.5)。一个参与角色和域是一个 role_and_domain 与 class_of_individual 的交集。

示例 2：图 150 显示了用作“测量器具”的“167 类型仪表”是一个 participating_role_and_domain。连接流体度量类和这个 participating_role_and_domain 的一个 class_of_participation 限制了使用“167 类仪表”所进行的测量。# 789 是一个“167 类型仪表”，参与到了测量 activity# 1234 中。

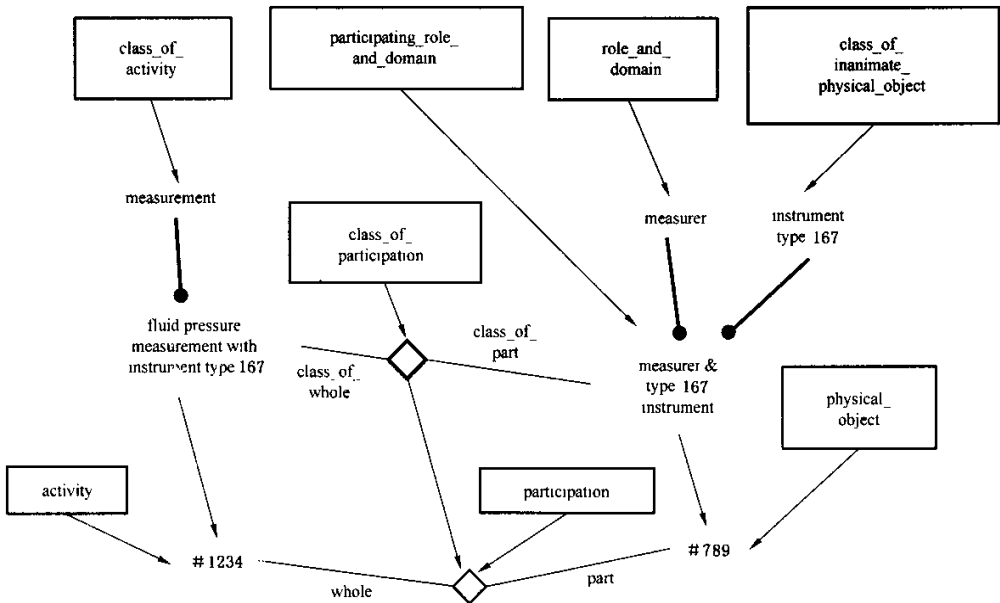


图 150 使用 167 型仪表的流体测量活动

可以为 class_of_activity 定义任意个 participating_role_and_domain。

示例 3：图 151 显示“被测量”的角色“流体”是一个 participating_role_and_domain。连接流体度量类和这个 participating_role_and_domain 的一个 class_of_participation 限制了测量流体的测量方法。# S27 是一股 stream 的临时部件在

测量 activity# 1234 中被测量。

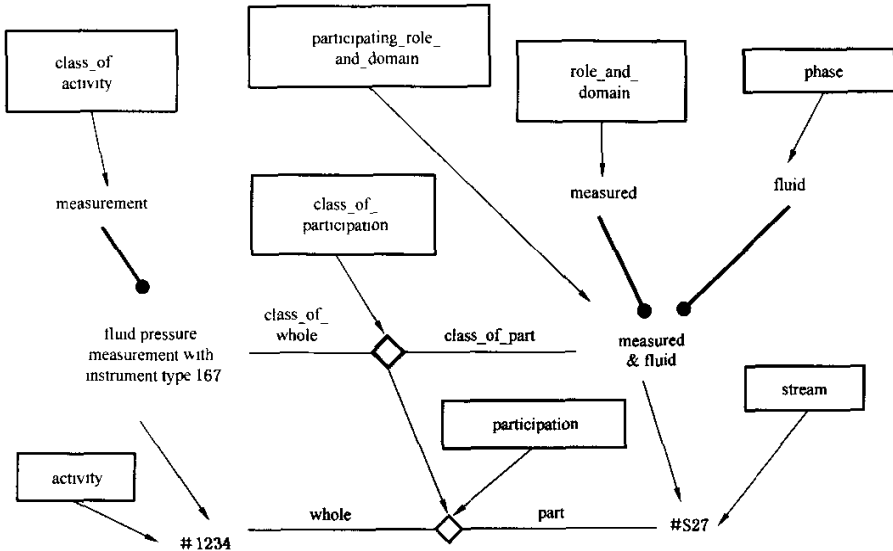


图 151 流体压力测量活动

class_of_recognition(识别类)是一个 class_of_relationship(关系类)指明了可以被视为 class_of_activity 成员结果的 thing 的 class(见 5.2.10.6)。

示例 4: 图 152 显示了一个流体测量活动,其结果是将被测量的流体划分到一个“压力”特性类。这个特定的测量 activity# 1234 的结果把 classification# S27 归类到“26 bar”“压力”等级。

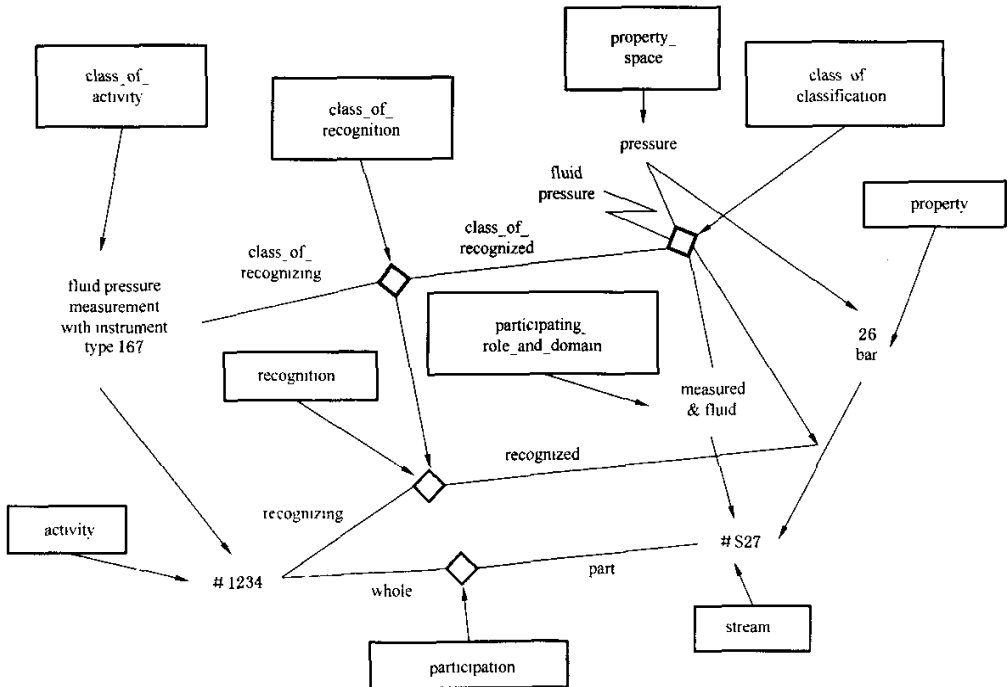


图 152 流体压力度量

4.8.4.10 个体类的类

class_of_individual 的其他子类型没有被显式地建模,但可以用 class_of_class_of_individual(个体

类的类)定义并用来区分所需要的 class_of_individual(见 5.2.7.4 和图 183)。图 153 显示了 class_of_class_of_individual 的一些重要类型的显式模型。

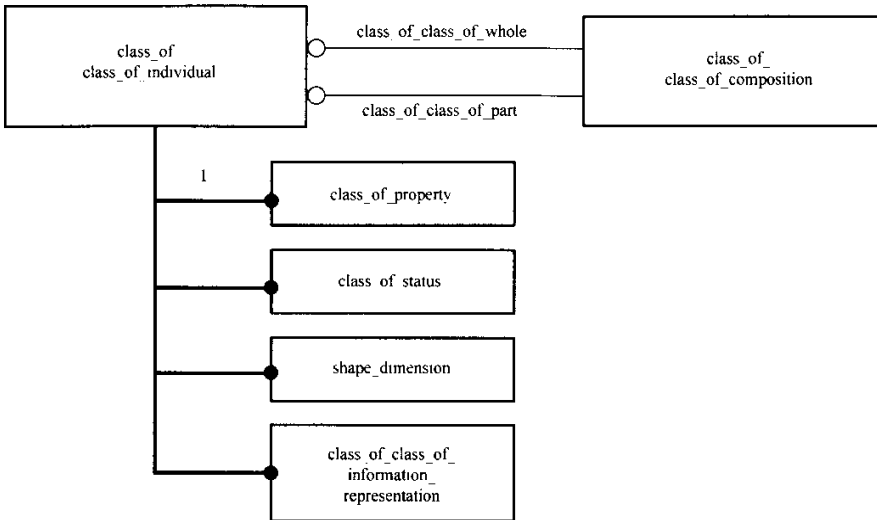


图 153 个体类的类

4.8.5 数量

4.8.5.1 算术数量

arithmetic_number(算术数量)是一个 class_of_class(类的类),如图 154 所示(见 5.2.5.1 和图 179)。在本部分中,整数与实数不同,但都是单维数量。

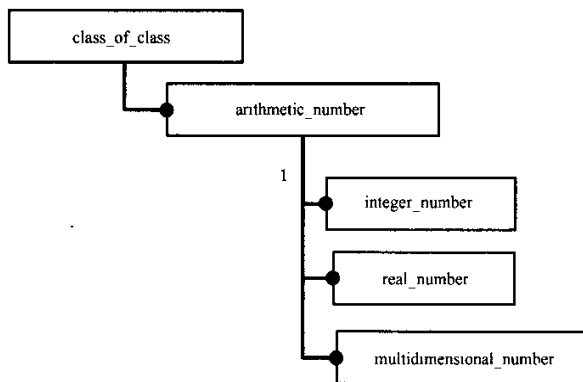


图 154 算术数

注:数量的表示可以被 class_of_information_representation 处理。

示例 1: 图 155 显示了用 EXPRESS_real 12.7 表示的一个 real_number 类“X”。

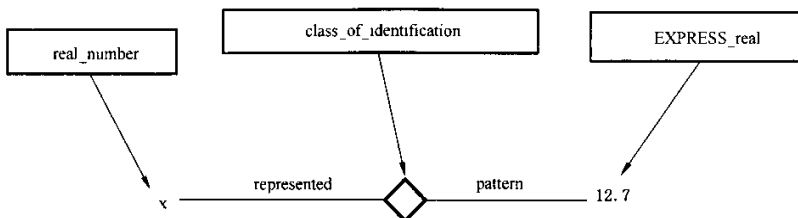


图 155 实数的表示

multidimensional_number (多维数)使 arithmetic_number 的有序对、三元组等能够被定义 (见 5.2.5.7,图 180 和图 181)。

示例 2: 图 156 显示了坐标三元组 [1.2, 2.3, -6.8]是一个 multidimensional_number。元素数的数量是有意义的。三元组 [2.3, 1.2, -6.8]是另一个不同的三元组。

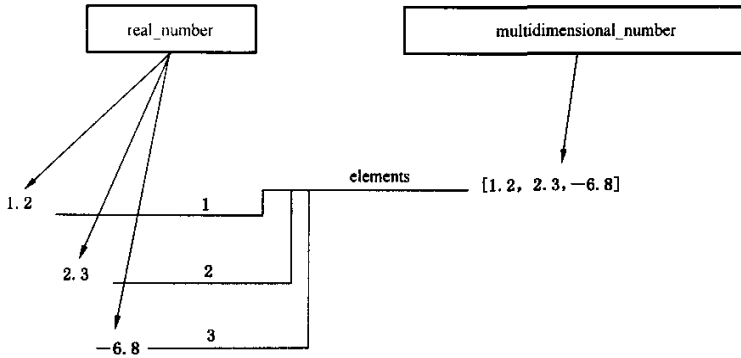


图 156 多维数量

4.8.5.2 数量的类

class_of_number(数量的类)是一个 class_of_class,包含了数量的连续和离散集合(见 5.2.5.3,图 179 和图 181),其模型如图 157 所示。

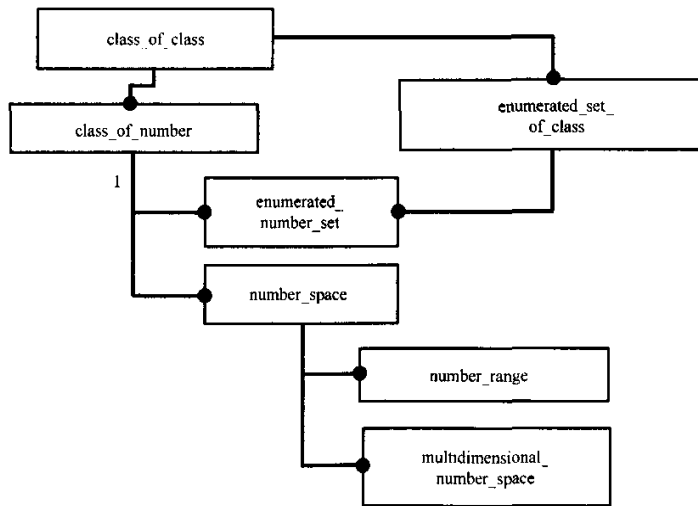


图 157 数量的类

4.8.5.2.1 枚举数量集合

enumerated_number_set(枚举数量集合)使数量的非连续集合可以被定义,它可以全部是实数或整数或二者的混合(见 5.2.5.4)。

示例: 图 158 显示了 integer_number45, 59, 73 是 enumerated_number_set 的成员,并且没有隐含任何顺序。

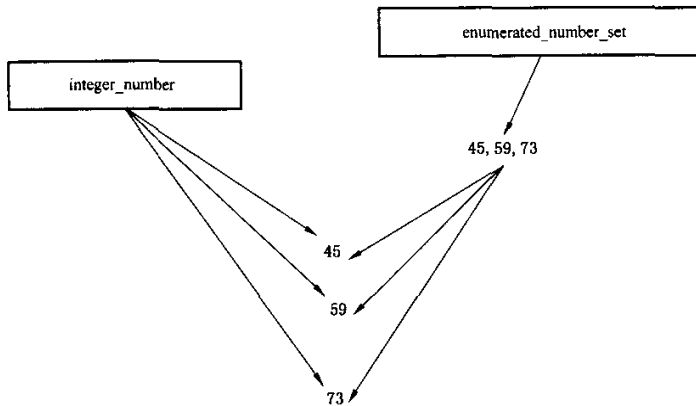


图 158 枚举数量集合

4.8.5.2.2 数量值域

number_range(数量值域)是限定于单维 number_space 的(见 5.2.5.9),值域的上界和下界是 number_range 的特定成员,其模型如图 159 所示。

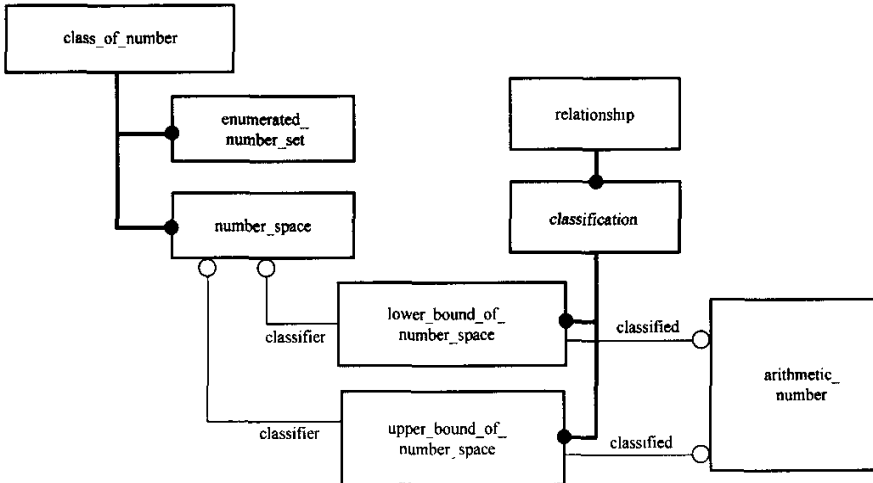


图 159 数量值域边界

示例: 图 160 显示了值域 5.2~9.3 中的 real_number 是一个有下边界 5.2 和上边界 9.3 的 number_range。

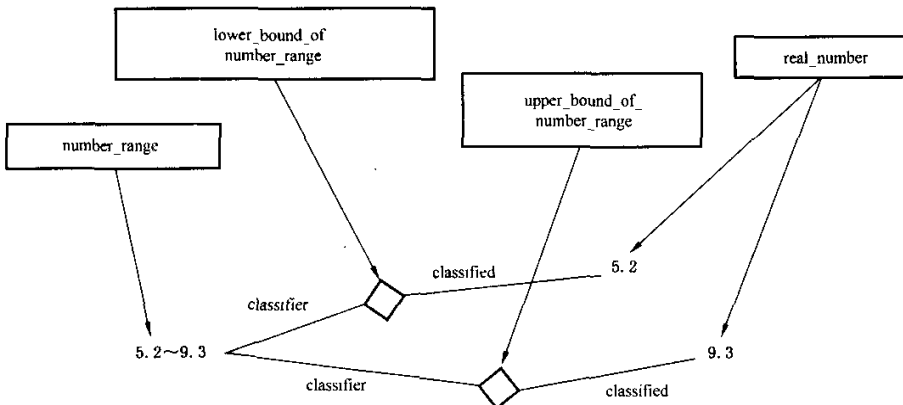


图 160 数量值域 5.2~9.3

4.8.5.2.3 多维数量空间

multidimensional_number_space(多维数量空间)是 multidimensional_number 的连续(见 5.2.5.8,图 180 和图 181)。

示例 1: 图 161 显示所有实数的连续统“R1”是一个 number_space。三维实数 number_space “R3”是一个 multidimensional_number_space,并有“R1”作为其第 1、2、3 个元素。

注: 本模型中的“R1”和实体类型 real_number 是相同的事物。

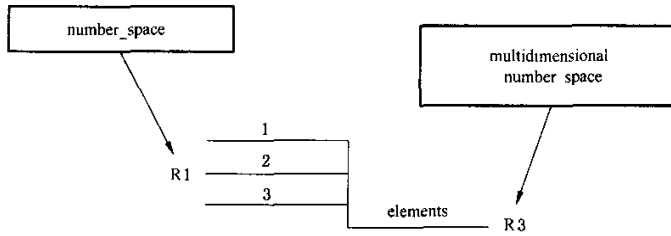


图 161 R3 实数空间

示例 2: 图 162 显示了“复数”是一个 multidimensional_number_space。复数 7.1 + 9.3i 是一个 multidimensional_number,其实部是 7.1,虚部是 9.3。“复数”中的元素由 role_and_domain 中的“实部”和“虚部”定义。这些是实虚角色和 real_number 域的结合。[7.1;9.3]作为“复数”决定了这两个数量的角色。

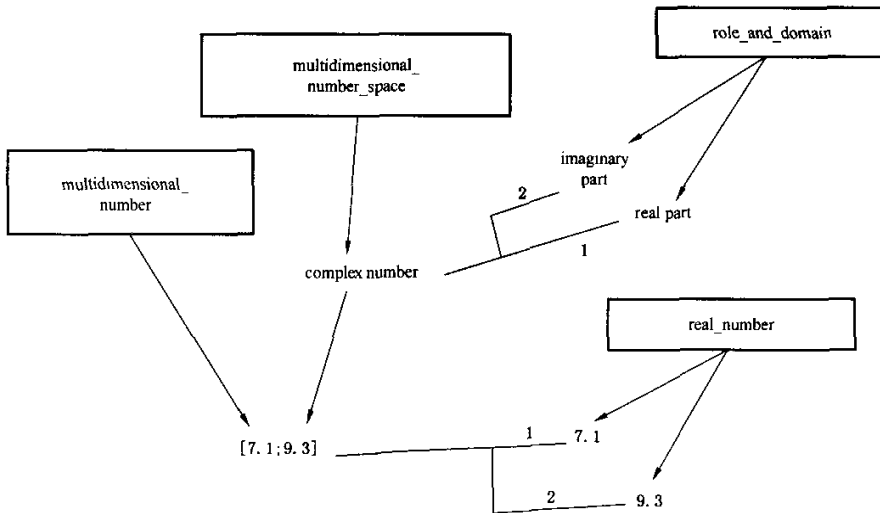


图 162 复数

4.9 功能映射

functional_mapping 是把一个输入映射到一个结果的一种 relationship(见 5.2.15.3 和图 191)。这样的映射可以是 1 对 1 或多对 1。1 对 1 是同构映射。功能映射关系的集合或类形成了函数。图 163 显示了 functional_mapping 的模型元素。

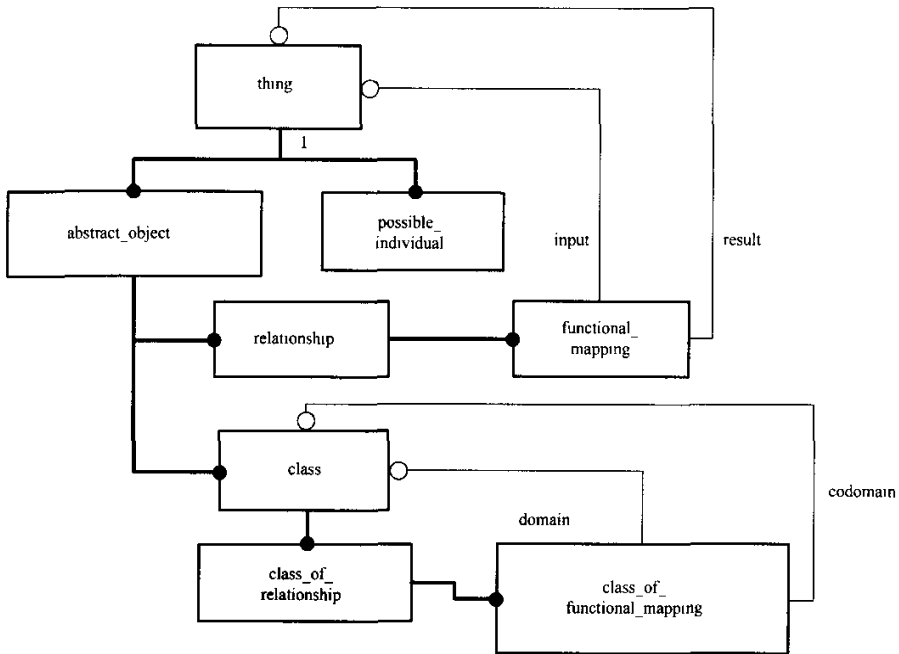


图 163 功能映射

示例 1: 图 164 显示了函数 X^2 的数据, 其中 X 是任意 `real_number`。

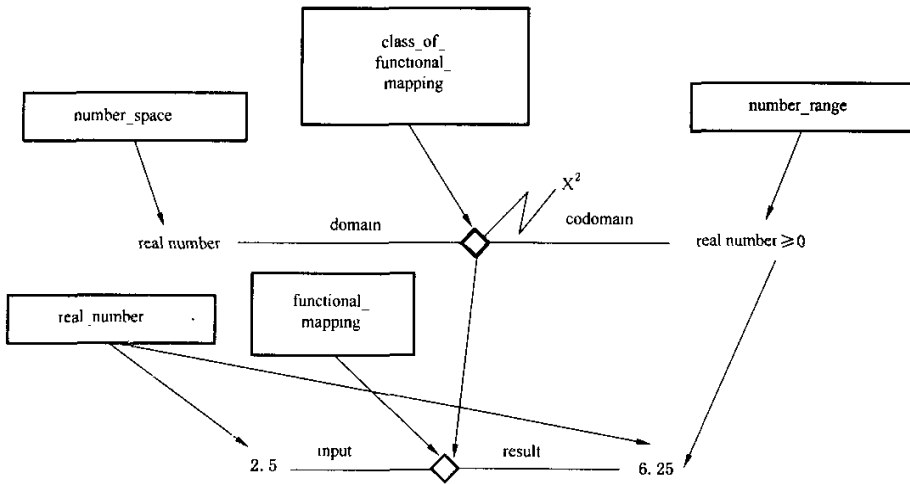


图 164 函数 X^2 的数据

`functional_mapping` 并不限于数。

示例 2: 图 165 显示了一个压力“差”的 `functional_mapping`。“a”是一个过滤器的“上游”压力,“b”是“下游”压力。压力“c”是压力“a”和“b”之间的“差”。这个 `class_of_functional_mapping` (功能映射类)“差”把一个由两个 `role_and_domain` 组成的 `class_of_multidimensional_object` 即输入参数, 映射到差异输出 `role_and_domain`。这个映射不是同构的, 因为压力“c”可以是许多“a”和“b”组合的结果。

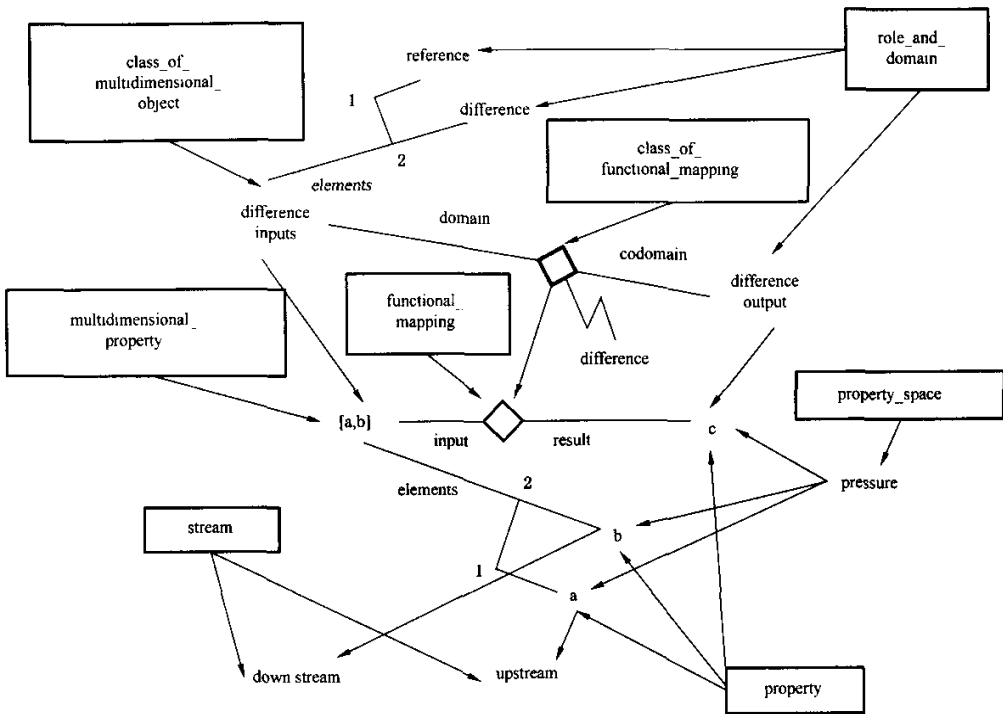


图 165 压差功能映射

功能映射的一些类可以建模成 `functional_mapping` 的显式子类(见图 191),如图 166 所示。三个子类分别完成集合的交、并和差操作(见 5.2.25)。 `property_quantification` 和它的集合副本 `scale` 在 4.8.4.3.2 中描述。

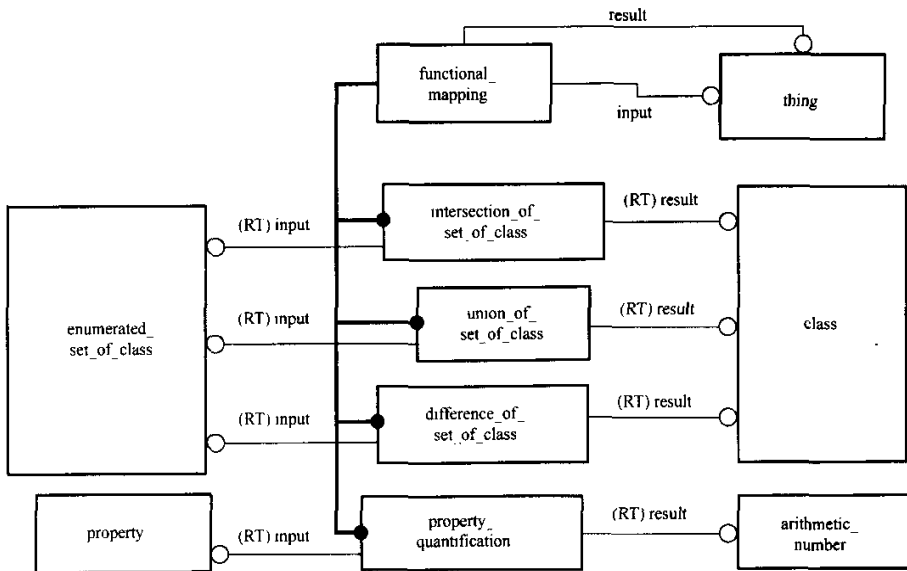


图 166 功能映射子类型

示例 3: 图 167 显示了三个维恩图,分别定义了类“*I*”是类“*A*”、“*B*”和“*C*”的交集,类“*U*”是类“*A*”、“*B*”和“*C*”的并集,类“*D*”是类“*A*”、“*B*”和“*C*”的差集。图 168 显示了用 `property_quantification` 类型表示的模型。

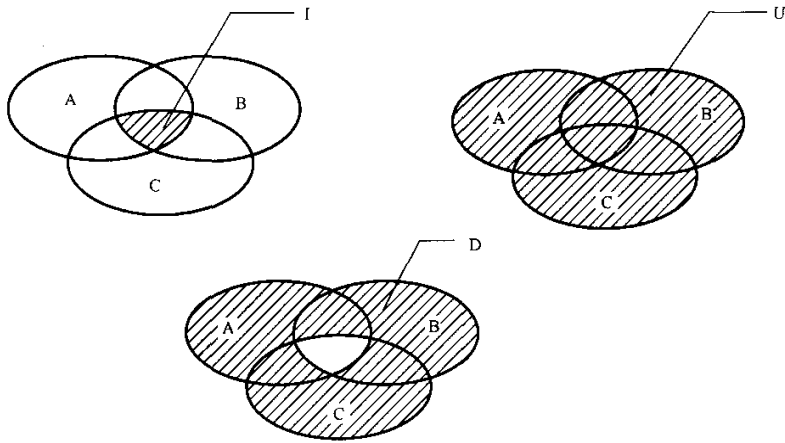


图 167 类 ABCIU 和 D 的维恩图

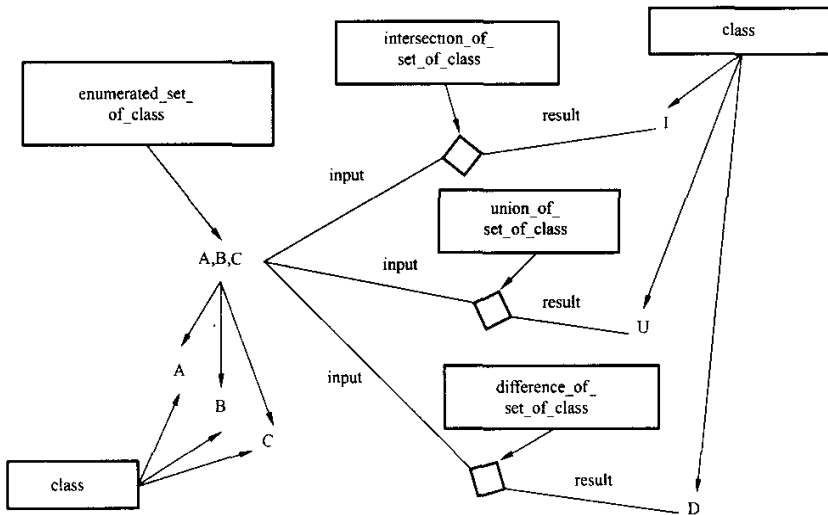


图 168 类 A/B/C 的交集、并集和差集

4.10 其他用户定义关系

4.10.1 其他关系

许多种类的关系没有被显式地建模。为了使这些关系能够被表示，已经定义了实体类型 `other_relationship` (其他关系) (见 5.2.11.1 和图 187)，其模型如图 169 所示。

任何两个 `thing` 都可能被包含在一个 `other_relationship` 中，用 `end_1` 和 `end_2` 角色将它们区分开来。`other_relationship` 不包括 `relationship` 显式子类型的成员。一个 `other_relationship` 的意义和含义可以通过把它归到一个或多个 `class_of_relationship` 来确定 (见 5.2.12 和图 188)。

示例：图 170 显示了一个个体的有序对，由一段时间内的“我”和一辆“汽车”组成。这是一个 `other_relationship`，可以被归到所有权关系。然而并没有确定是我拥有汽车，还是汽车拥有我。

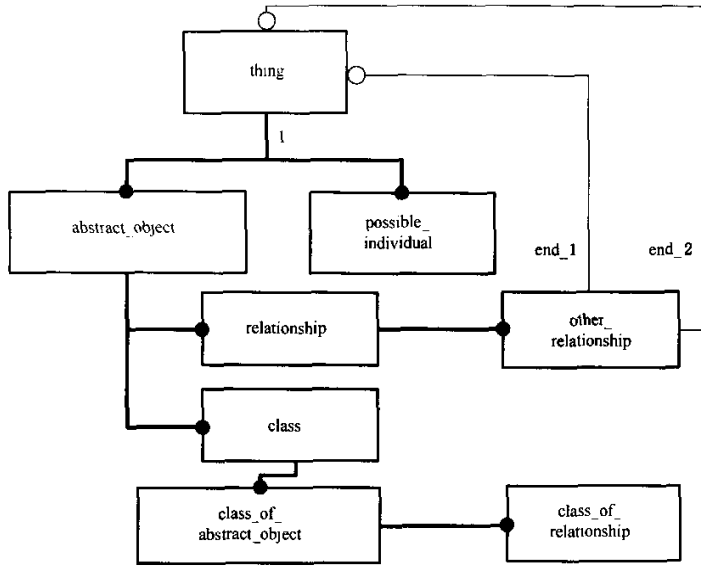


图 169 其他关系

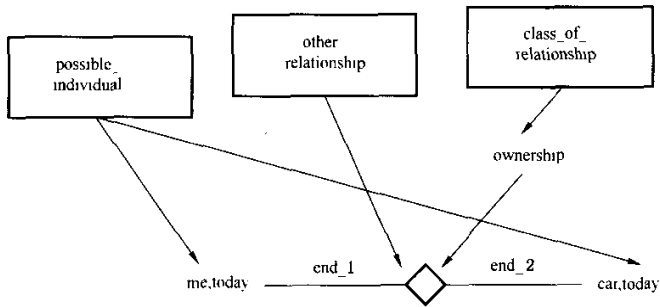


图 170 所有权关系

4.10.2 签名关系类

other_relationship 的类型也可以使用 class_of_relationship_with_signature(带签名的关系类)确定(见 5.2.13.2 和图 189)。这就使得成员关系的意义和含义、成员关系参与者的角色、成员关系参与者的域和类型的约束都可以被描述。

图 171 显示了 class_of_relationship_with_signature 的模型。role_and_domain 类被用来限制成员关系的两端。

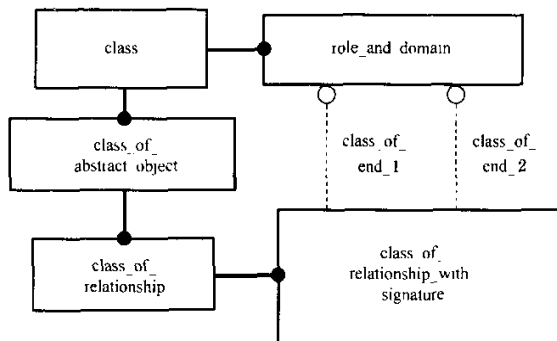


图 171 签名关系类

示例 1: 上个例子中忽略的“所有者”和“被所有”的角色可以通过把 other_relationship 确定为含“所有者”和“被所有”角色的 class_of_relationship_with_signature 来定义,其数据如图 172 所示。“所有者”和“所有权”都是 role 的成员,因为它们不受域的限制。同样 possible_individual 也是相应角色的成员。

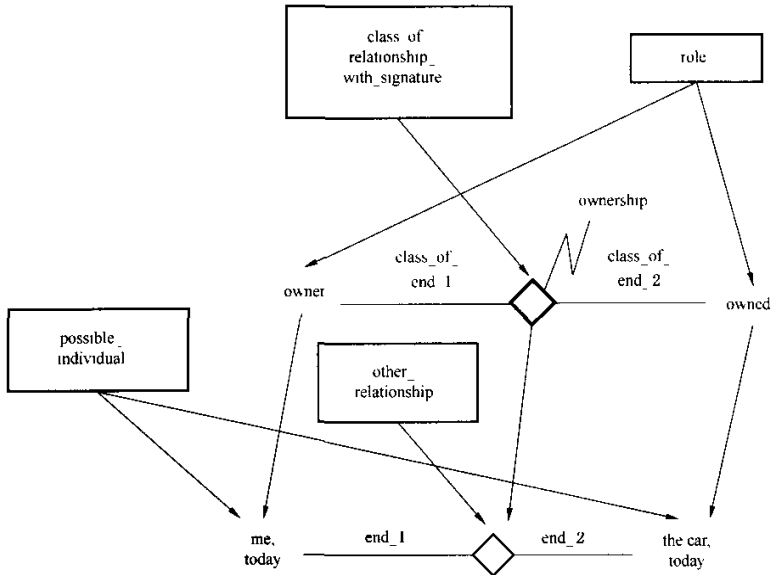


图 172 签名关系类的所有权

示例 2: 图 173 显示了使用 class_of_relationship_with_signature,对关系类“个体插入”进行建模。class_of_end_1 指向 role_and_domain 的“插入与个体”,class_of_end_2 指向 role_and_domain 的“主体与个体”。连接 physical_object # 1234 和 # AC6756 的 other_relationship 是 class_of_relationship “个体插入”的成员。

class_of_relationship “管热电偶套管插入”是“个体插入”的特殊化,它把 class_of_end_2 的域限定在“管道”,class_of_end_1 的域限定在“管热电偶套管”而不是任何个体。

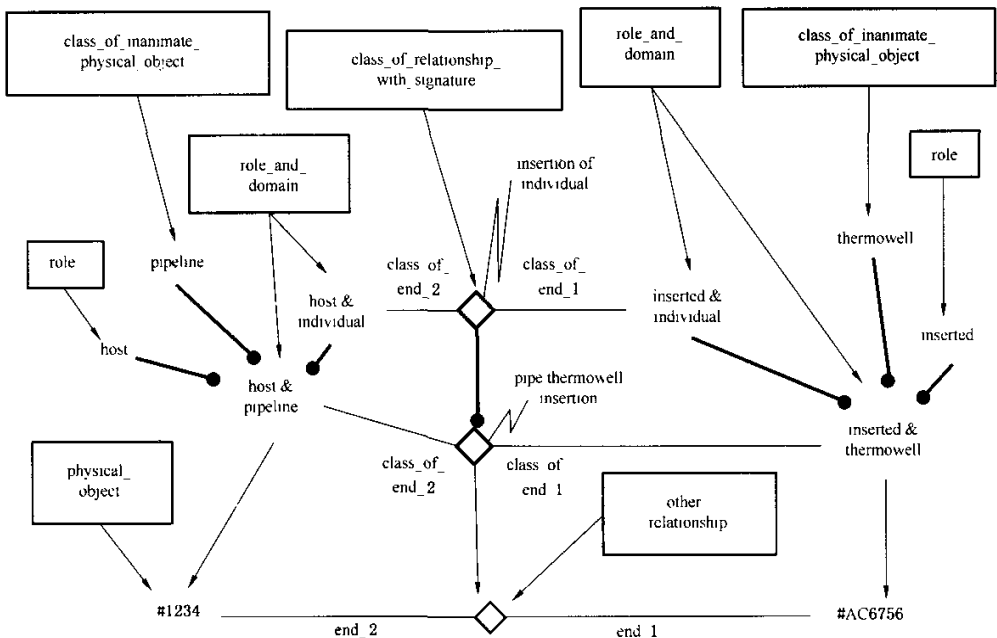


图 173 个体的交集

physical_object # 1234 是 role_and_domain “主体和管道”的成员,即“主体”role 和“管道”域的组合。“主体和管道”是 role_and_domain “主体和个体”的子集(特殊化)。physical_object # AC6756 是 role_and_domain “插入 & 热电偶套管”的成员,即 role“插入”和域“热电偶”的特殊化。

4.10.3 基数约束

基数约束可以应用到 class_of_relationship_with_signature 的成员上。

示例: 图 174 显示了使用基数约束和 class_of_relationship_with_signature 来定义 class_of_inanimate_physical_object “6 个 M8 螺栓”,即每个成员是 6 个螺栓。class_of_relationship_with_signature “6 个 M8 螺栓”有一个基数,这样每一个“6 个 M8 螺栓”总是都与不同的 6 个 M8 螺栓有连接关系。一包螺栓的基数是 0 到 1,因为一个螺栓可以不在某 6 个螺栓组成的包中,但也只能在一个包中出现一次。class_of_relationship_with_signature 是更一般的 class_of_relationship “个体组装”的特殊化。图 174 显示了 6 个特定 M8 螺栓的关系。

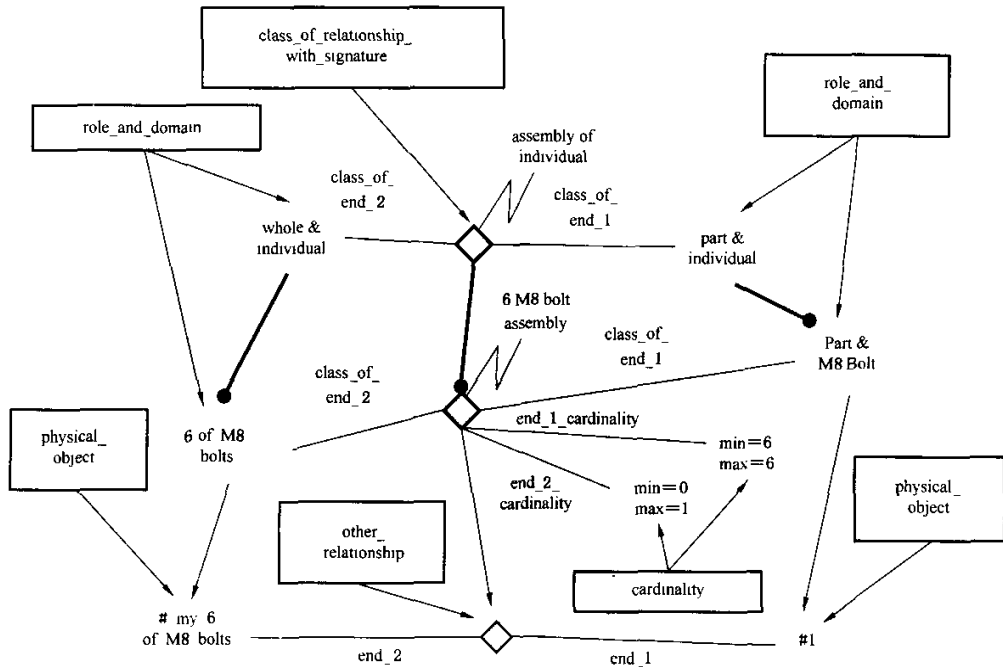


图 174 6 个 M8 螺栓

4.10.4 非对称的其他关系类

class_of_relationship_with_signature 可以与 class_of_relationship_with_related_end_1(端 1 关系类)和 class_of_relationship_with_related_end_2(端 2 关系类)结合起来,去识别非对称的非显式 class_of_relationship(见 5.2.13.2,5.2.12.3,5.2.12.4 和图 188),其模型如图 175 所示。

示例: 图 176 显示了使用 class_of_relationship_with_signature 定义 class_of_relationship “制作”。“制作”关系把“制造商”和“产品”联系起来。另一个 class_of_relationship_with_signature 也是一个 class_of_relationship_with_related_end_1,被用来定义“制作”关系的特殊化,即“制造”由“Bloggs & 公司”完成。连接个体 # 1234 和“Bloggs & 公司”的 other_relationship 是 class_of_relationship“Bloggs 制作”的成员。

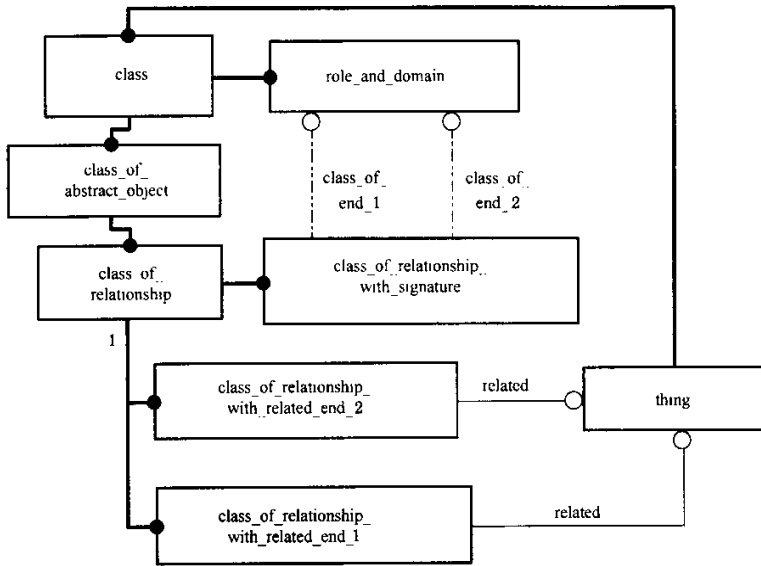


图 175 带签名的非对称关系类

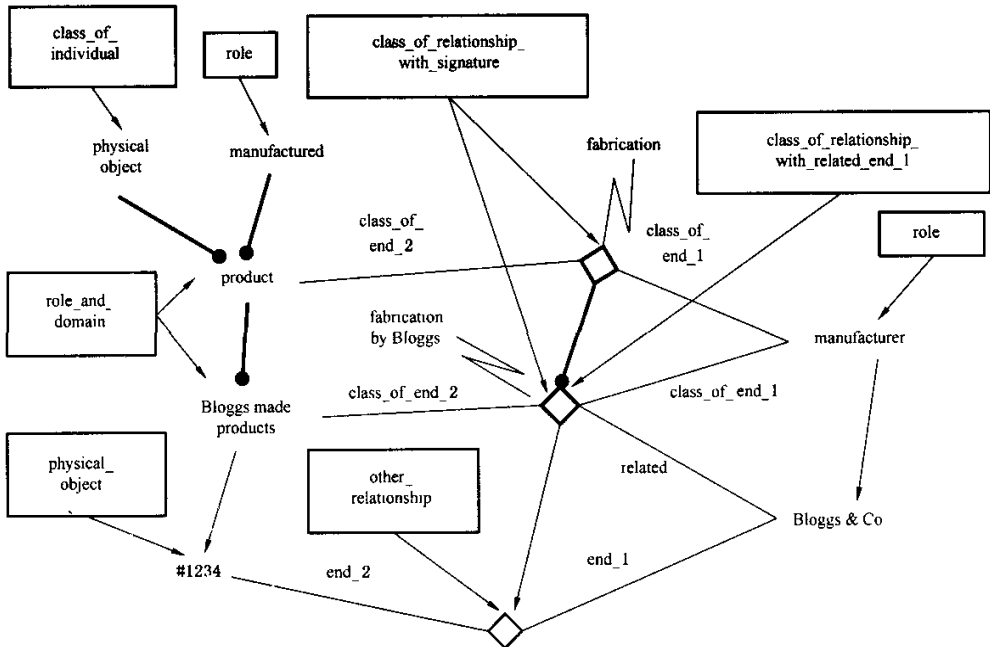


图 176 Bloggs 的产品

5 生命周期集成模式

5.1 概述

本章规定了支持生命周期集成的模式。只是为了叙述上的方便把本章分成许多条,并不表示每条描述的主题是单独的或可分离的模式。

注 1：本部分规定了一系列完整的 EXPRESS 模式，没有注或其他解释性文本，在互联网（参见附录 B）上可以找到这些模式。

注 2：本模式使用了 EXPRESS 语言的部分规定。没有使用附录 C 中的规定列表。

5.2 模式定义

下列 EXPRESS 声明从 lifecycle_integration_schema 开始。

EXPRESS 描述：

```

* )
SCHEMA lifecycle_integration_schema;
( *
    
```

5.2.1 所有事物

本条包含实体数据类型 thing 的声明，它是 lifecycle_integration_schema 的根实体数据类型。

注：图 177 是本条所定义实体数据类型的图。

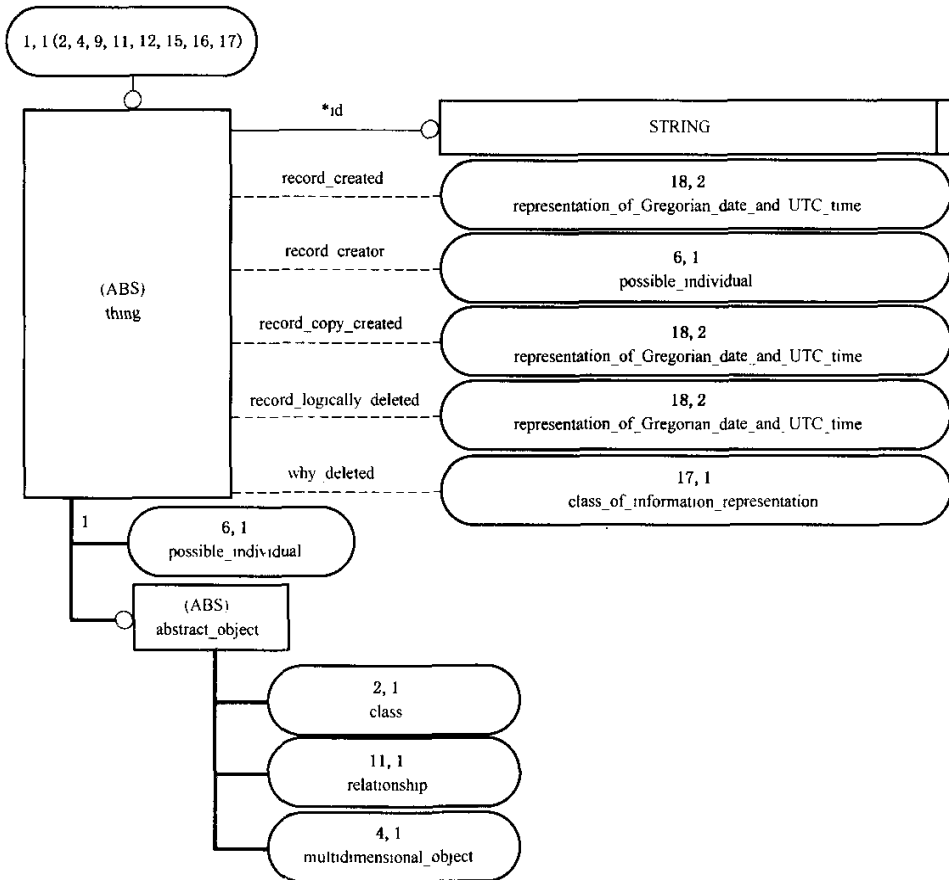


图 177 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 1 个)

5.2.1.1 抽象对象

abstract_object(抽象对象)是时空中不存在的 thing。

EXPRESS 描述:

*)

ENTITY abstract_object

ABSTRACT SUPERTYPE

SUBTYPE OF(thing);

END_ENTITY;

(*

5.2.1.2 事物

thing 是或可能是想到或察觉到的任何事,包括物质和非物质的对象、概念和行动。

每个 thing 是 possible_individual,或是 abstract_object。

注 1: 在系统中,每个 thing 是可标识的。为了将来作为 identification 引用,可以存储其他系统生成的和作为数据交换部分而接收到的系统标识符,并引用原始的机构或系统。

注 2: 为其他实体数据类型(本模式中声明的)提供的每个例子也是 thing 的例子。

EXPRESS 描述:

*)

ENTITY thing

ABSTRACT SUPERTYPE OF (ONEOF(possible_individual, abstract_object));

id : STRING;

record_copy_created : OPTIONAL representation_of_Gregorian_date_and_UTC_time;

record_created : OPTIONAL representation_of_Gregorian_date_and_UTC_time;

record_creator : OPTIONAL possible_individual;

record_logically_deleted : OPTIONAL representation_of_Gregorian_date_and_UTC_time;

why_deleted : OPTIONAL class_of_information_representation;

UNIQUE UR1 : id;

END_ENTITY;

(*

属性定义:

id: thing 的标识符(为了在系统中管理记录)。

record_copy_created: 在当前系统中复制记录时的日期和时间。当前系统不是原始系统时,该属性将只有一个值。

record_created: 在原始系统中记录第一次生成时的日期和时间。

record_creator: 在原始系统中第一次生成该记录的人、机构或系统。

record_logically_deleted: 逻辑删除该记录的日期和时间。

why_deleted: 逻辑删除记录的原因。

注: 逻辑删除意味着系统中历史记录存在错误时,该记录仍然可用,从来就不认为它是一个有效生命。也就是说,认为它永远不是真。

形式化声明:

UR1: thing 的 id 在一个系统中应该是唯一的。

5.2.2 类

本条定义了表达类的实体数据类型。

注: 图 178 是本条定义的实体数据类型的图(参见 4.6.3 和 4.8)。

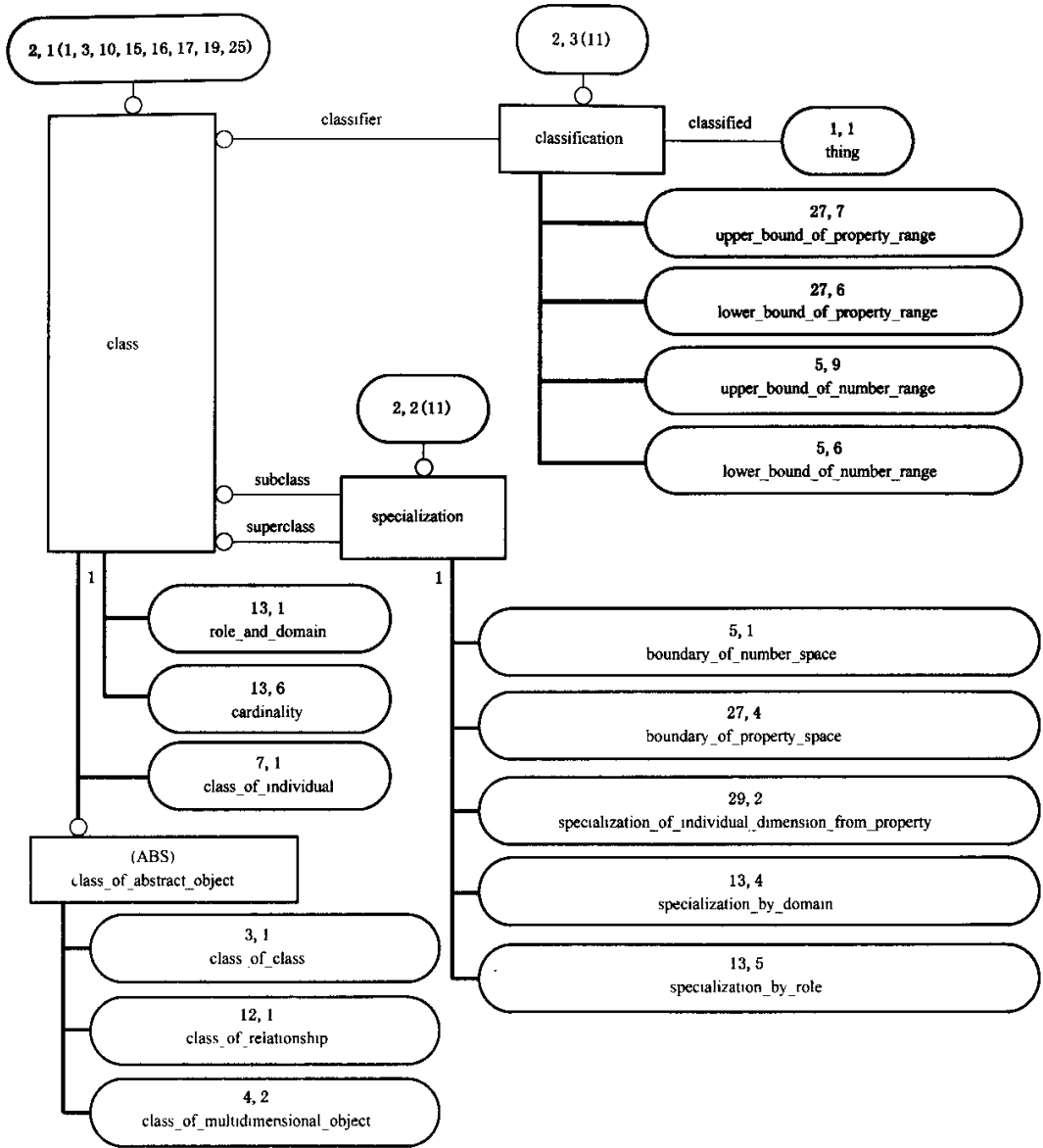


图 178 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 2 个)

5.2.2.1 类

class 是 thing, 它理解 thing 的自然属性并按照一个或多个标准把 thing 分成类的成员。通过类的成员最终定义其标识。没有两个类具有相同的成员关系。然而, 必须区分具有成员的 class 和那些已知的成员, 因此在一个信息系统中可能随时改变已记录的成员, 即使不改变真正的成员关系。

注 1: 根据本模式, 作为时空范例的结果, class 的成员关系是不变的。

class 可以是另一个 class 或其自身的成员。

注 2: 本模型中应用于类的集合理论是不完善的集合理论^[3](参见 D.2.4)。这样允许声明“类是类的成员”,不允许声明传统的集合理论,例如:标准文本^[4]中创建的 Zermelo-Fraenkel 集合理论。

有没有成员的空 class。

注 3: 用 classification 标识 class 的已知成员。

示例 1: “Centrifugal pump”是一个类。

示例 2: “Mechanical equipment type”是一个类。

示例 3: “Temperature”是一个类。

示例 4: “Commercial fusion reactor”是一个类。

注 4: 虽然只有一个没有成员的 class,但是可能有在现实生活中没有成员的 class。这些 class 在其他可能的世界中有成员。

示例 5: “Centigrade scale”是一个类。

EXPRESS 描述:

*)

ENTITY class

SUPERTYPE OF (role_and_domain ANDOR cardinality ANDOR ONEOF(class_of_individual, class_of_abstract_object))

SUBTYPE OF(abstract_object);

END_ENTITY;

(*

5.2.2.2 抽象对象的类

class_of_abstract_object(抽象对象的类)是其成员分类 abstract_object 成员的 class。

EXPRESS specification:

*)

ENTITY class_of_abstract_object

ABSTRACT SUPERTYPE

SUBTYPE OF(class);

END_ENTITY;

(*

5.2.2.3 分类

classification 是一种 relationship,它暗示被分类的 thing 是(统一处理“分类器”)class 的成员。

分类不是可传递的。

注: relationship 的子类是可传递的,如果 A 与 B 关联,并且 B 与 C 以同样的方式关联,那么 A 与 C 必然以那种方式关联。specialization 和 composition 是 relationship 子类传递的示例。然而,因为 classification 是不可传递的,并不意味着 A 不能与 C 以相同的方式关联,只是不需要从 A 关联 B 和 B 关联 C 来导出。

示例 1: 显示“London”为已知类“capital city”成员的 relationship 是 classification。

示例 2: 显示“pump”为已知类“equipment type”成员的 relationship 是 classification。

EXPRESS 描述:

*)

ENTITY classification

SUBTYPE OF(relationship);

classified : thing;

classifier : class;

END_ENTITY;

(*

属性定义:

Classified; thing, 是分类器 class 的一个成员。

Classifier; 被分类的 thing 是其成员的 class。

5.2.2.4 专门化

specialization 是 relationship, 它显示子类的所有成员是超类的成员。specialization 是可传递的。

注: 如果 A 是 B 的 specialization, 且 B 是 C 的 specialization, 那么 A 必然是 C 的 specialization。

示例: “Centrifugal pump”是“pump”的 specialization。

EXPRESS 描述:

*)

ENTITY specialization

```

SUPERTYPE OF (ONEOF(
    boundary_of_number_space,
    boundary_of_property_space,
    specialization_by_domain,
    specialization_by_role,
    specialization_of_individual_dimension_from_property))

```

```

SUBTYPE OF (relationship);

```

```

subclass : class;

```

```

superclass : class;

```

END_ENTITY;

(*

属性定义:

subclass; 是超类 class 专门化的 class。

superclass; 是子类 class 通用化的 class。

5.2.3 类分类

本条定义了表达类分类的实体数据类型。

注: 图 179 是本条定义的实体数据类型图。

5.2.3.1 类的类

class_of_class(类的类)是其成员为 class 实例的 class。

注: 当有必要对 class_of_class 进行分类时, 可以使用另外一个 class_of_class。这是因为 class_of_class 是 class。

EXPRESS 描述:

*)

ENTITY class_of_class

```

SUPERTYPE OF (ONEOF( arithmetic_number, class_of_class_of_individual,
    class_of_class_of_relationship, class_of_number, class_of_property_space,
    class_of_shape_dimension) ANDOR enumerated_set_of_class)

```

```

SUBTYPE OF (class_of_abstract_object);

```

END_ENTITY;

(*

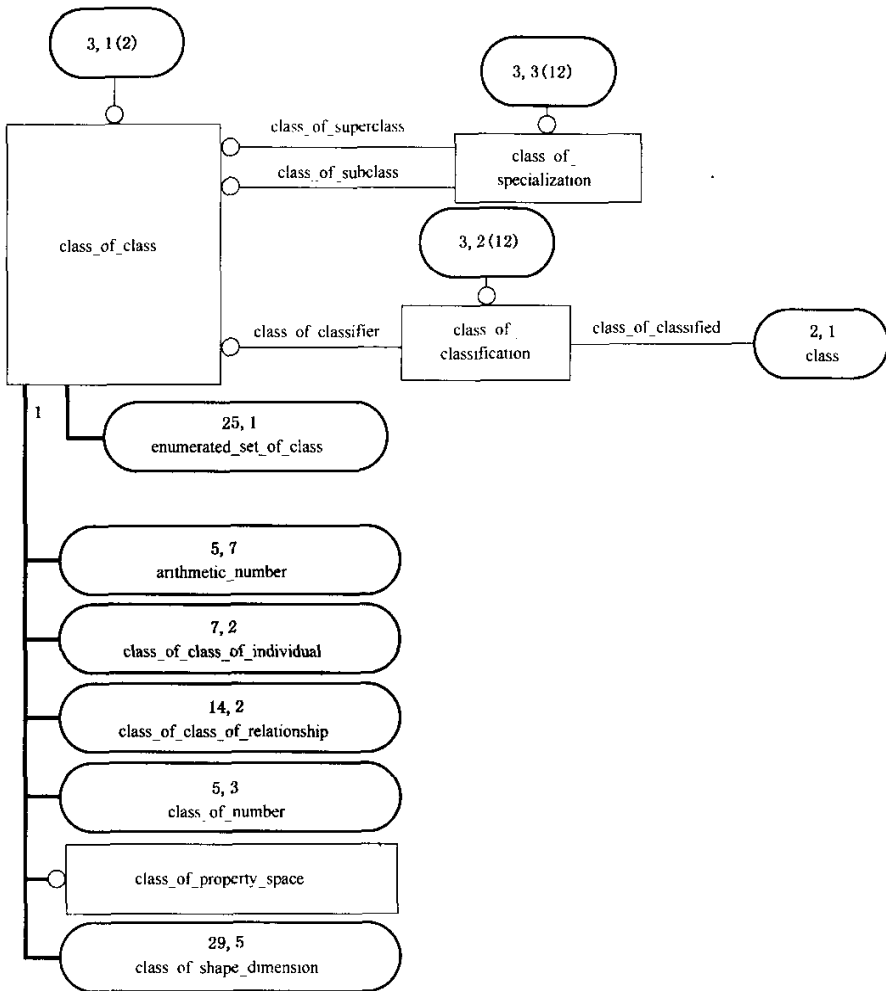


图 179 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 3 个)

5.2.3.2 分类的类

class_of_classification(分类的类)是其成员为 classification 成员的 class_of_relationship。class_of_relationship 显示用一个或多个 class_of_classifier class_of_class 成员分类 class_of_classified class 的成员。

示例：可以用 class_of_classification 的实例表达 class“centrifugal pump”和 class_of_property “RPM”之间的连接(指示“centrifugal pump”至少是一个“RPM” class 的成员)。

EXPRESS 描述：

*)

ENTITY class_of_classification

SUBTYPE OF(class_of_relationship)；

```

class_of_classified ; class;
class_of_classifier ; class_of_class;
END_ENTITY;

```

(*

属性定义:

class_of_classified: 是 class_of_classification 中 class_of_classified 的 class。

class_of_classifier: 是 class_of_classification 中 class_of_classified 的 class_of_class。

5.2.3.3 特性空间类

class_of_property_space(特性空间类)是其成员是 property_space 成员的 class_of_class。

示例 1: 不同空间和自由度的属性曲线、属性区域和属性量是 class_of_property_space 的成员。

示例 2: “Pump performance curve”是 class_of_property_space 的例子。

EXPRESS 描述:

*)

```

ENTITY class_of_property_space
    SUBTYPE OF(class_of_class);
END_ENTITY;

```

(*

5.2.3.4 专门化类

class_of_specialization(专门化类)是其成员是 specialization 实例的 class_of_relationship。它显示 class_of_subclass 成员是 class_of_superclass 成员的子类。

示例: class_of_specialization 显示类“family of ASME bolts”的成员(例如:3 in 螺钉、2 in 螺钉)是 enumerated_property_set “set of bolt lengths”成员的专门化(例如:3 in、2 in)。

EXPRESS 描述:

*)

```

ENTITY class_of_specialization
    SUBTYPE OF(class_of_relationship);
    class_of_subclass ; class_of_class;
    class_of_superclass ; class_of_class;
END_ENTITY;

```

(*

属性定义:

class_of_subclass: 在 class_of_specialization 成员中,其成员是子类的 class_of_class。

class_of_superclass : 在 class_of_specialization 成员中,其成员是超类的 class_of_class。

5.2.4 多维对象

本条定义了表达多维对象的实体数据类型。

注: 图 180 是本条定义的实体数据类型图(参见 4.6.5、4.8.4.3.6 和 4.8.5.2.3)。

5.2.4.1 多维对象类

class_of_multidimensional_object(多维对象类)是 class,其成员是 multidimensional_object 的实例。在 roles 属性中的相同位置规定每个位置在被分类 multidimensional_object 中所扮演的角色。在 parameters 属性中的相同位置规定应用于 roles 任何位置的常量。用 cardinalities 属性中的相同位置规定 roles 属性的 cardinalities。

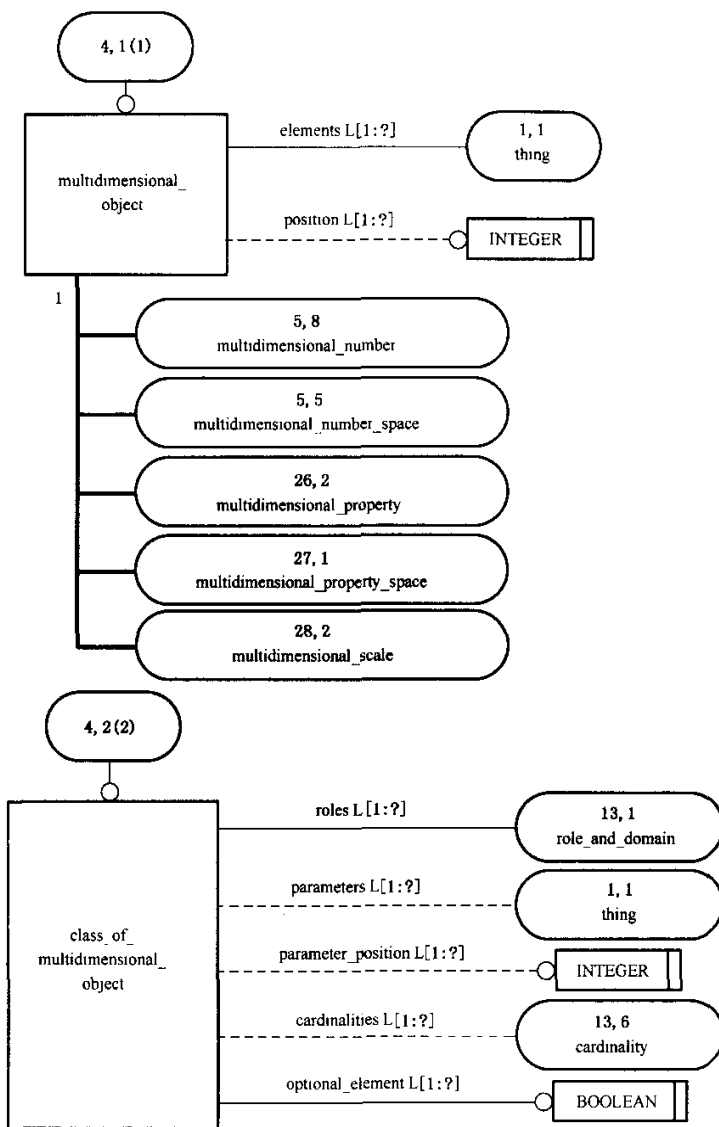


图 180 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 4 个)

示例：将参数[a,b,x]输入 multidimensional_object 的定义函数 $y = a + bx$ 中，使摄氏度转换成华氏度，并以 parameter_position 表[1,2]列出参数 parameters 表[32,1,8]，这就是 class-of-multidimensional-object 的一个例子。

EXPRESS 描述：

*)

ENTITY class_of_multidimensional_object

SUBTYPE OF(class_of_abstract_object);

cardinalities ; OPTIONAL LIST [1:?] OF cardinality;

```

optional_element ; LIST [1:?] OF BOOLEAN;
parameters ; OPTIONAL LIST [1:?] OF thing;
parameter_position ; OPTIONAL LIST [1:?] OF INTEGER;
roles ; LIST [1:?] OF role_and_domain;

```

END_ENTITY;

(*

属性定义:

cardinalities:应用于角色的基数列表。如果没有规定基数,那么对基数没有约束。如果规定了基数,那么应该为所有的角色规定基数。

optional_element:值 TRUE 意味着其是可选的,值 FALSE 意味着其是强制性的。

parameters:与角色关联的参数列表。

parameter_position:与参数列表角色相关的位置列表。

注:因为 EXPRESS LIST 数据类型不允许为空,所以本属性是必须的。位置列表提供角色位置的映射。

Roles:与被分类 multidimensional_object 相关联的角色。

5.2.4.2 多维对象

multidimensional_object(多维对象)是已排序 thing 列表的 abstract_object。class_of_multidimensional_object 的成员(指示了每个成员所扮演的角色)确定了 multidimensional_object 的重要性。

注:multidimensional_object [A,B,C]与[B,C,A]不同。

示例:[32, 1.8, 20]是 multidimensional_object,为了把 20 摄氏转换到华氏,可以规定函数 $y=a+bx$ 的输入参数。

EXPRESS 描述:

*)

ENTITY multidimensional_object

```

SUPERTYPE OF (ONEOF(multidimensional_property_space,
    multidimensional_number,
    multidimensional_property,
    multidimensional_number_space,
    multidimensional_scale))

```

```

SUBTYPE OF(abstract_object);

```

```

elements ; LIST [1:?] OF thing;

```

```

position ; OPTIONAL LIST [1:?] OF INTEGER;

```

END_ENTITY;

(*

属性定义:

Elements:组成 multidimensional_object 的 thing 列表。通过分类确定每个 thing 的角色。

Position:在分类 class_of_multidimensional_object 中与角色列表相关的元素位置。元素应该按递增顺序排列。缺少某些元素时就需要这个属性。EXPRESS 列表数据类型不允许列表中有空元素。当这个属性有值时,它提供映射信息。当这个属性没有值时,提供所有元素。

5.2.5 数

本条规定了表达数的实体数据类型。

注:图 181 是本条定义的实体数据类型图(参见 4.8.5)。

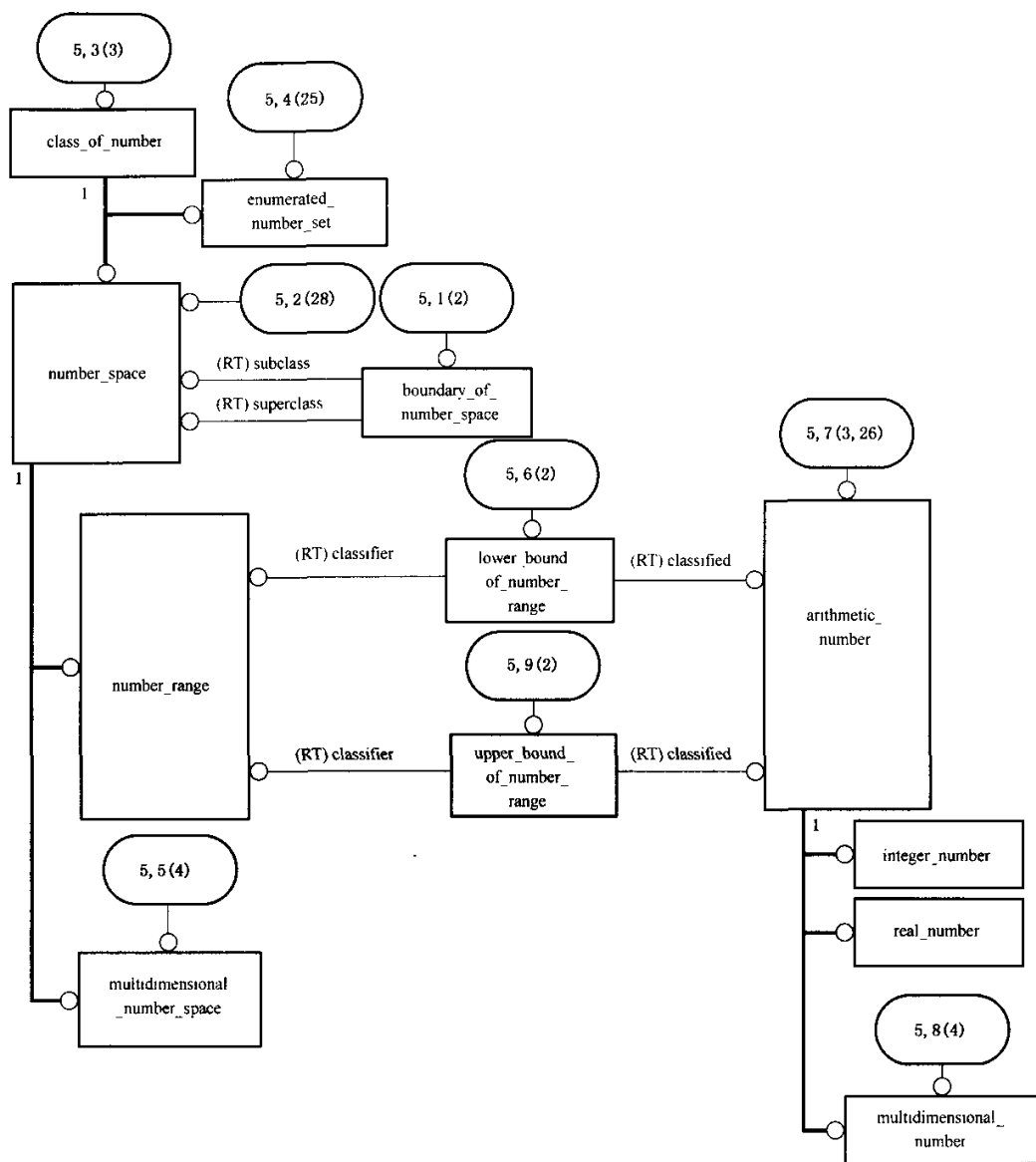


图 181 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 5 个)

5.2.5.1 算术数

arithmetic_number(算术数)是 class_of_class, 它的成员类具有相同的符号和大小。arithmetic_number 是数的本身, 不是数的任何表达形式。

注: integer_number 不是 real_number 的子类。integer_number 的成员是一部分的 real_number 成员, 与其子集不是同形。

示例 1: 可以用 arithmetic_number 的实例表达数 2 和数 2.0。

示例 2: 十五(该数本身不是英文“fifteen”)是 arithmetic_number。可以用 EXPRESS_integer 表达它, 也可以用“XV”、二进制或十六进制表达方式表达。

EXPRESS 描述:

*)


```

ENTITY arithmetic_number
    SUPERTYPE OF (ONEOF(real_number, integer_number, multidimensional_number))
    SUBTYPE OF(class_of_class);
END_ENTITY;
( *

```

5.2.5.2 数字空间的边界

boundary_of_number_space(数字空间的边界)是 specialization,它显示 number_space 是另一个 number_space 的边界。

示例:在 R3 中,立方体的侧面是平面 number_space,并且是立方体 number_space 的边界。

EXPRESS specification:

```

* )
ENTITY boundary_of_number_space
    SUBTYPE OF(specialization);
    SELF\specialization.subclass : number_space;
    SELF\specialization.superclass : number_space;
END_ENTITY;
( *

```

5.2.5.3 数类

class_of_number(数类)是 class_of_class。其成员是 arithmetic_number 的成员。

示例:可以用 class_of_number 的实例表达主要数的类。

EXPRESS 描述:

```

* )
ENTITY class_of_number
    SUPERTYPE OF (ONEOF(number_space, enumerated_number_set))
    SUBTYPE OF(class_of_class);
END_ENTITY;
( *

```

5.2.5.4 枚举型数集

enumerated_number_set(枚举型数集)是 class_of_number 和 enumerated_set_of_class。

示例:可以用 enumerated_number_set 的实例表达整数集{3,4,5}。

EXPRESS 描述:

```

* )
ENTITY enumerated_number_set
    SUBTYPE OF(class_of_number, enumerated_set_of_class);
END_ENTITY;
( *

```

5.2.5.5 整数

integer_number 是整数 arithmetic_number。

示例:1、2 和 10 是 integer_number 的表示形式。

EXPRESS 描述:

```

* )

```

```

ENTITY integer_number
    SUBTYPE OF(arithmetic_number);
END_ENTITY;
( *

```

5.2.5.6 数域下限

lower_bound_of_number_range(数域下限)是一种关系,它指出了 arithmetic_number 是 number_range 的最小值。

示例: 3.1 是域[3.1~5.3]的下限。

EXPRESS 描述:

```

* )
ENTITY lower_bound_of_number_range
    SUBTYPE OF(classification);
    SELF\classification.classified : arithmetic_number;
    SELF\classification.classifier : number_range;
END_ENTITY;
( *

```

属性定义:

classified: 作为被划分对象,是 number_range 下限的 arithmetic_number。

classifier: 作为分类器,用 arithmetic_number 划界的 number_range。

5.2.5.7 多维数

multidimensional_number(多维数)是 arithmetic_number,也是 multidimensional_object。

示例: [3.2, 5.4, 55.6]是 multidimensional_number。

EXPRESS 描述:

```

* )
ENTITY multidimensional_number
    SUBTYPE OF(arithmetic_number, multidimensional_object);
END_ENTITY;
( *

```

5.2.5.8 多维数空间

multidimensional_number_space(多维数空间)是 number_space 和 multidimensional_object。

示例: R3(定义为所有的三维实数,例如:(1.0, 2.1, 5.4))是 multidimensional_number_space。

EXPRESS 描述:

```

* )
ENTITY multidimensional_number_space
    SUBTYPE OF(number_space, multidimensional_object);
END_ENTITY;
( *

```

5.2.5.9 数域

number_range(数域)是一维的 number_space。

示例: 从-273.1 到无穷大的 number_space 是 number_range。

EXPRESS 描述:

*)

```
ENTITY number_range
    SUBTYPE OF(number_space);
END_ENTITY;
```

(*)

5.2.5.10 数空间

number_space(数空间)是连续的 class_of_number。

示例：1~5 的整数和 0.000~1.000 的实数是 number_space 的例子。

EXPRESS 描述：

*)

```
ENTITY number_space
    SUPERTYPE OF (ONEOF(number_range, multidimensional_number_space))
    SUBTYPE OF(class_of_number);
END_ENTITY;
```

(*)

5.2.5.11 实数

real_number(实数)是实数 arithmetic_number。

示例：3.2116 是 real_number 的表现形式。

EXPRESS 描述：

*)

```
ENTITY real_number
    SUBTYPE OF(arithmetic_number);
END_ENTITY;
```

(*)

5.2.5.12 数域上限

upper_bound_of_number_range(数域上限)是 relationship,它指出了 arithmetic_number 是 number_range 的最大值。

示例：5.3 是域[3.1~5.3]的上限。

EXPRESS 描述：

*)

```
ENTITY upper_bound_of_number_range
    SUBTYPE OF(classification);
    SELF\classification.classified : arithmetic_number;
    SELF\classification.classifier : number_range;
END_ENTITY;
```

(*)

属性定义：

classified:作为被划分对象,是 number_range 上限的 arithmetic_number。

classifier:作为分类器,在 upper_bound_of_number_range 中划界的 number_range。

5.2.6 可能个体

本条定义了表达可能个体的实体数据类型。

注：图 182 是本条定义的实体数据类型图(参见 4.6.2 和 4.7)。

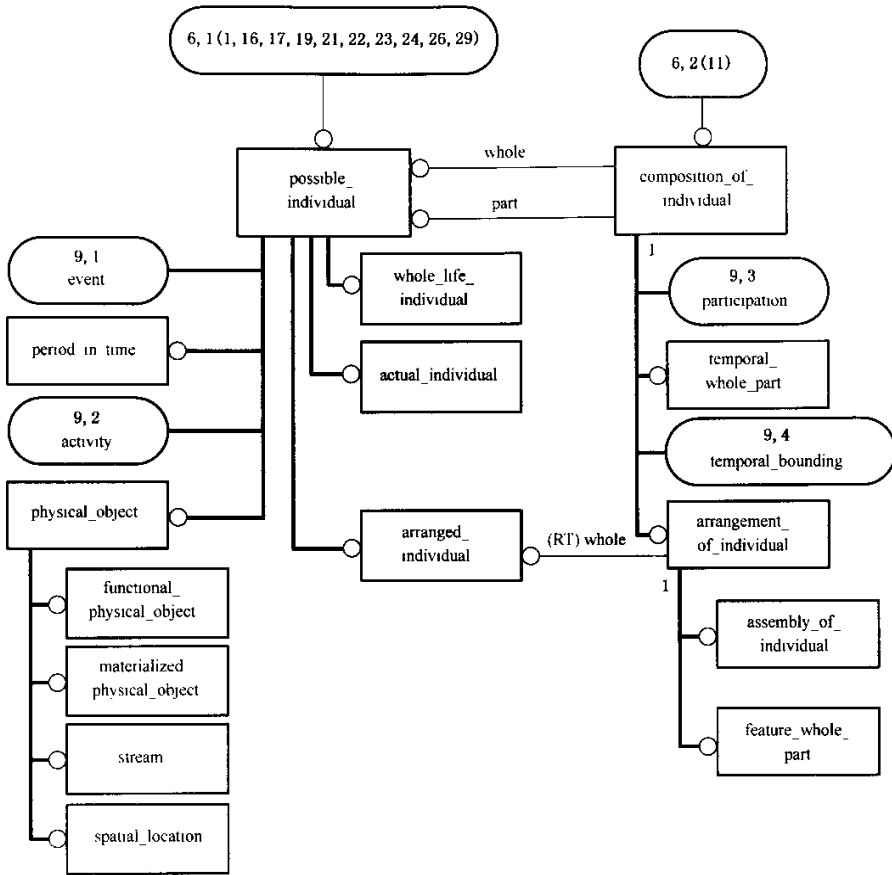


图 182 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 6 个)

5.2.6.1 实际个体

actual_individual(实际个体)是 possible_individual,它是我们所熟悉的时空统一体的一部分。与某些想象的世界相对,它存在于我们现实世界的现在、过去和将来。

注:我们计划的事物通常只能假设为某些想象世界的一部分,直到它们真的发生。

示例 1:埃菲尔铁塔是 actual_individual。

示例 2:用于编辑本部分的计算机是 actual_individual。

示例 3:虚构的人物 Sherlock Holmes 是不是 actual_individual 的 possible_individual。

示例 4:2300 年的地球(假设仍然存在)是 actual_individual。

EXPRESS 描述:

*)

```
ENTITY actual_individual
    SUBTYPE OF (possible_individual);
```

END_ENTITY;

(*

5.2.6.2 已安排个体

arranged_individual(已安排个体)是有部分的 possible_individual,它扮演区别于整体的角色。

arranged_individual的质量区别于其部分的质量。

示例 1: 带系列号 V-1234 的容器是 arranged_individual。

示例 2: 公司 Bloggs & Co 是 arranged_individual。

示例 3: 由主要部件移动 CD-ROM、软盘驱动器和电源电缆组成的膝上型电脑是 arranged_individual。

EXPRESS 描述:

```
*)
ENTITY arranged_individual
    SUBTYPE OF(possible_individual);
END_ENTITY;
(*
```

5.2.6.3 个体的安排

arrangement_of_individual(个体的安排)是指零件是一部分 arranged_individual 的 composition_of_individual。部分的临时范围是整体的临时范围。

arrangement_of_individual 可以是 assembly_of_individual。

注 1: 术语“已安排的”指出了零件相对整体来说具有特殊的任务。

注 2: 安排关系没有规定与整体中其他零件的关系本质。与 connection_of_individual 和 relative_location 一样的关系将暗示这一点。

示例 1: 当可以用 arrangement_of_individual 的实例表达编队的某部分时,关系指出了具体的飞机正在飞行。

示例 2: 当可以用 arrangement_of_individual 的实例表达规划设计时,关系指出了仓库中的特殊箱柜是仓库的一部分。

EXPRESS 描述:

```
*)
ENTITY arrangement_of_individual
    SUPERTYPE OF (ONEOF(assembly_of_individual, feature_whole_part))
    SUBTYPE OF(composition_of_individual);
    SELF\composition_of_individual.whole ; arranged_individual;
END_ENTITY;
(*
```

属性定义:

whole:arranged_individual 是 arrangement_of_individual 中的整体。

5.2.6.4 个体组装

assembly_of_individual(个体组装)是 arrangement_of_individual,它暗示了部分直接或间接地连接整体的其他部分。部分和整体是超分子对象。

注: 通过 class_of_arrangement_of_individual 的实例表达分子和更小物体的成分。

示例: 当可以用 assembly_of_individual 的实例表达组合泵时,关系指出了叶轮的临时部分是组合泵的一部分。

EXPRESS 描述:

```
*)
ENTITY assembly_of_individual
    SUBTYPE OF(arrangement_of_individual);
END_ENTITY;
(*
```

5.2.6.5 个体的成分

composition_of_individual(个体的成分)是关系,它指出了部分 possible_individual 是整体 possible_individual 的一部分。指出简单的成分,除非也举出了子类。composition_of_individual 是可传递的。

注:例如,简单成分意味着没有必要包含或关心部件安排。在有部件安排的地方,通过 arrangement_of_individual 指出它,也可以通过一个子类暗示简单成分。

示例:一颗沙子是一堆沙子的一部分是 composition_of_individual 的实例。

EXPRESS 描述:

*)

ENTITY composition_of_individual

SUPERTYPE OF (ONEOF(arrangement_of_individual, temporal_whole_part,
participation, temporal_bounding))

SUBTYPE OF(relationship);

part : possible_individual;

whole : possible_individual;

END_ENTITY;

(*

属性定义:

part:是整体 possible_individual 一部分的 possible_individual。

whole:在 composition_of_individual 中是整体的 possible_individual。

5.2.6.6 特征整体零件

feature_whole_part(特征整体零件)是一个 arrangement_of_individual,它指出该零件是一个不能分隔的紧接的整体零件。

注:它包括不能破坏性分解和组装的整体,例如:泵上铸造的进口法兰。

示例:可以用 feature_whole_part 实例表达指出法兰的表面是法兰部件的关系。

EXPRESS 描述:

*)

ENTITY feature_whole_part

SUBTYPE OF(arrangement_of_individual);

END_ENTITY;

(*

5.2.6.7 功能物质对象

functional_physical_object(功能物质对象)是 physical_object,作为其基本的标识,它具有功能联系性,而不仅仅是材料。functional_physical_object 邻近的临时部件不需要有通用的物质或能量,如果每个临时部件的物质或能量实现同样的功能。

示例:可以用 functional_physical_object 实例表达知名的 E-4507 热交换系统,E-4507 是蒸馏转换系统的部件。它明显不同于“制造系列号是 ES/1234 的壳和管道热交换器”,当第一次建立工厂和以后设备破损拆除时,为了用不同系列号的新的热交换器代替,安装 ES/1234 代替 E-4507。可以用 materialized_physical_object 实例表达“制造系列号是 ES/1234 的壳和管道热交换器”以及其不同系列的替代品。安装 ES/1234 代替 E-4507 时,ES/1234 的临时部件也是 E-4507 的临时部件。

EXPRESS 描述:

*)

ENTITY functional_physical_object

```

SUBTYPE OF(physical_object);
END_ENTITY;
( *

```

5.2.6.8 材料物质对象

materialized_physical_object(材料物质对象)是一个 physical_object,作为其基本的标识,它具有物质和/或能量。物质或能量的连续性要求某些物质或能量与 materialized_physical_object 邻近的临时部件是通用的。偶尔替换某些部件不会产生新的标识。

示例: 可以用 materialized_physical_object 的实例表达制造系列号为 ES/1234 的壳和管道热交换器。

EXPRESS 描述:

```

* )
ENTITY materialized_physical_object
SUBTYPE OF(physical_object);
END_ENTITY;
( *

```

5.2.6.9 时域

period_in_time(时域)是一个 possible_individual,它是时间部件(宇宙的时间部件)的所有空间。

示例 1: 2000 年 7 月是 period_in_time 的实例。

示例 2: 世界标准时间 2000-11-21T06:00 到 2000-11-21T11:53 描述的时期是一个符合 ISO 8601 的 period_in_time 实例。

EXPRESS 描述:

```

* )
ENTITY period_in_time
SUBTYPE OF(possible_individual);
END_ENTITY;
( *

```

5.2.6.10 物质对象

physical_object(物质对象)是一个 possible_individual,它是物质、能量或两者的分类。

示例 1: 一块金属是一个 physical_object。

示例 2: 一颗树是一个 physical_object。

示例 3: 标签 P101 标识的东西是一个 physical_object。

示例 4: 光束是一个 physical_object。

示例 5: 现场建造和拆除的罐都是 materialized_physical_object 和 functional_physical_object。

EXPRESS 描述:

```

* )
ENTITY physical_object
SUBTYPE OF(possible_individual);
END_ENTITY;
( *

```

5.2.6.11 可能的个体

possible_individual(可能的个体)是存在于时空中的事物。它包括:

——任何时空趋于零的空间中的事物;

- 任何时间或所有时间的所有空间以及任何空间中的那些事物；
- 所有时空中的所有事物；
- 实际存在或已经存在的事物；
- 过去、现在或将来虚拟或猜测的，以及可能存在的事物；
- 其他个体的临时部件(状态)；
- 具有特殊位置，但是在一个或多个方向没有长度的事物，例如：点、线和面。

在这个范围中，存在是基于某些一致性逻辑中的想象，包括实际的、假设的、计划的、希望的或需要的个体。

示例：可以用的实例表达系列号为 ABC123 的泵、Battersea 发电站、Sir Joseph Whitworth、莎士比亚和恒星飞船“Enterprise”。

EXPRESS 描述：

*)

ENTITY possible_individual

SUBTYPE OF(thing)；

END_ENTITY；

(*

5.2.6.12 空间位置

spatial_location(空间位置)是一个 physical_object，它具有相对位置的连续性。

示例：地理数据、执照区块、建筑区域、国土、空中走廊、海上交通带、危险控制区域、四维点、线、面、实体。

EXPRESS 描述：

*)

ENTITY spatial_location

SUBTYPE OF(physical_object)；

END_ENTITY；

(*

5.2.6.13 流

stream(流)是一个 physical_object，它是沿着一个通道移动的材料或能量，通道是基本的标识并可以约束。流由那些事物(当他们在通道中时是在流中)的临时部件组成。

示例 1：流量是通道穿过一个表面的、四维约束的 stream 例子。

示例 2：在原油蒸馏设备与铂重整装置间的管道中流动的石油是 stream。

EXPRESS 描述：

*)

ENTITY stream

SUBTYPE OF(physical_object)；

END_ENTITY；

(*

5.2.6.14 临时整体部件

temporal_whole_part(临时整体部件)是一个 composition_of_individual，它指出了一个人 possible_individual 是另一个 possible_individual 的临时部件。临时部件的空间范围是临时部件存在期的临时整体。

用于所有 possible_individual 的关系也可以用于 possible_individual 的临时部件，除非关系与整体

的临时本质有关。因此,如果连接一个 possible_individual,也可以连接其所有的临时部件,除非 whole_life_individual 不是通过其临时部件继承的。

注:因为 temporal_whole_part 是可传递的(从其超类继承),临时部件的层次可能是在顶部有一个 whole_life_individual。

示例 1: 可以用 temporal_whole_part 的实例表达指示了泵的操作期是泵的一个临时部件的关系。

示例 2: 可以用 temporal_whole_part 的实例表达指示了 1999 年三月是 1999 年第一季度一部分的关系。

EXPRESS 描述:

```

* )
ENTITY temporal_whole_part
    SUBTYPE OF(class_of_individual);
END_ENTITY;
( *

```

5.2.6.15 整个生命个体

whole_life_individual(整个生命个体)是一个 possible_individual,它是 class_of_individual 的成员,并且不是任何其他 possible_individual(也是相同 class_of_individual 的成员)的临时部件。whole_life_individual 包括他的过去和将来。

注: whole_life_individual 将来可能的临时部件是 possible_individual,它通过 temporal_whole_part 关系与 whole_life_individual 关联。

示例 1: 可以用 whole_life_individual 的实例表达塑料杯(用其生成和破坏 events 限制)。当杯子站立在桌子上时是本 whole_life_individual 的临时部分。

示例 2: 所有时段的万物是 whole_life_individual。

EXPRESS 描述:

```

* )
ENTITY whole_life_individual
    SUBTYPE OF(possible_individual);
END_ENTITY;
( *

```

5.2.7 个体类

本条声明了表达个体类的实体数据类型。

注: 图 183 是本条定义的实体数据类型图(见 4.8.4 和 4.8.4.10)。

5.2.7.1 个体安排类

class_of_arrangement_of_individual(个体安排类)是其成员是 arrangement_of_individual 实例的 class_of_composition_of_individual。

示例: 水是由 H₂O 分子组成的事实是 class_of_arrangement_of_individual 的实例。

EXPRESS 描述:

```

* )
ENTITY class_of_arrangement_of_individual
    SUPERTYPE OF (ONEOF(class_of_feature_whole_part,
                          class_of_assembly_of_individual, namespace))
    SUBTYPE OF(class_of_composition_of_individual);
    SELF\class_of_composition_of_individual.class_of_whole ;
    class_of_arranged_individual;

```

END_ENTITY;

(*

属性定义:

class_of_whole: 在 class_of_arrangement_of_individual 中是 class_of_whole 的 class_of_arranged_individual。

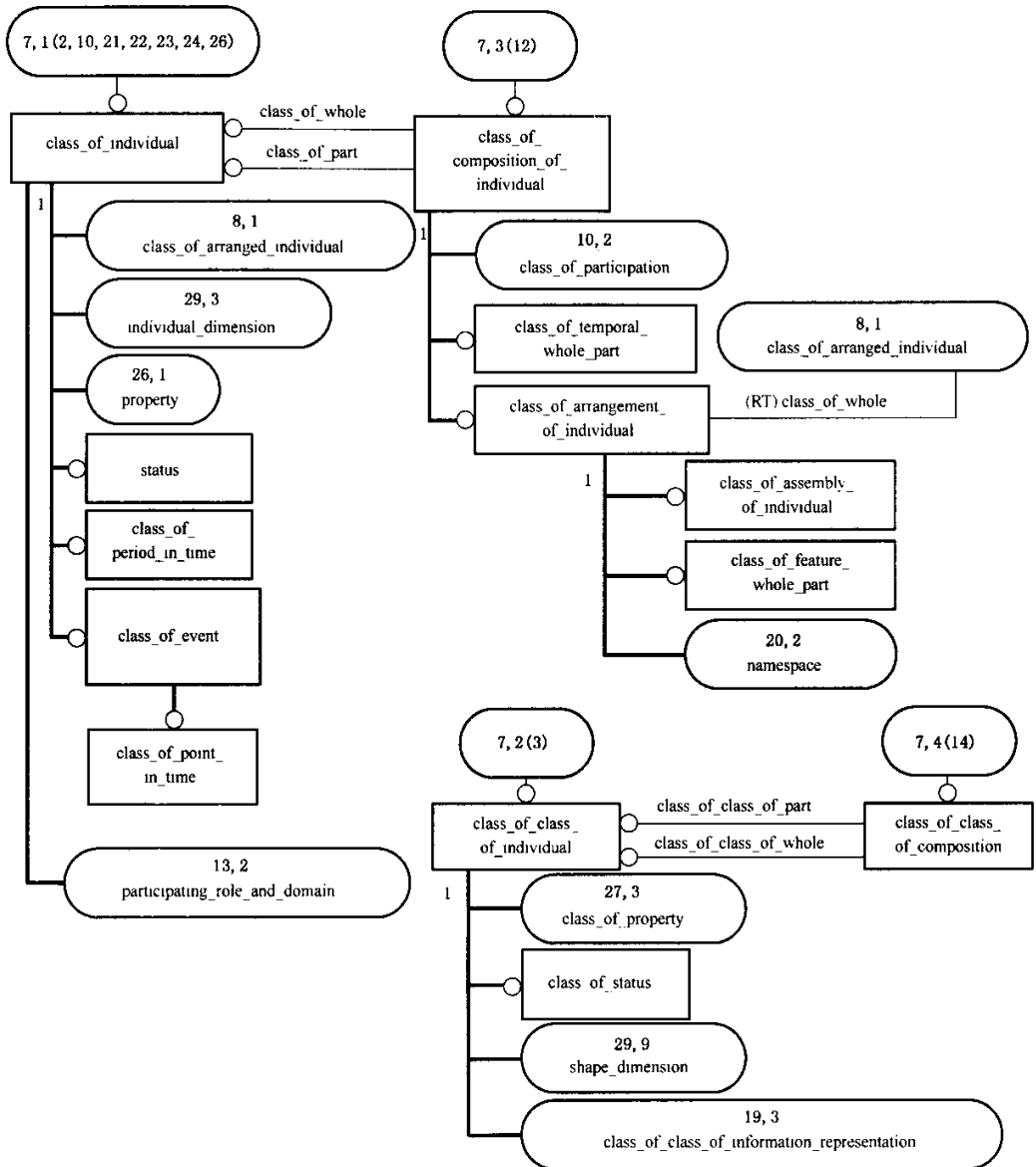


图 183 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 7 个)

5.2.7.2 个体组装类

class_of_assembly_of_individual(个体组装类)是其成员是 assembly_of_individual 实例的 class_of_arrangement_of_individual。

示例: 离心泵的叶轮是 class_of_assembly_of_individual。

EXPRESS 描述:

```

* )
ENTITY class_of_assembly_of_individual
    SUBTYPE OF(class_of_arrangement_of_individual);
END_ENTITY;
( *

```

5.2.7.3 成分类的类

class_of_class_of_composition(成分类的类)是其成员是 class_of_composition 实例的 class_of_class_of_relationship。它指出了 class_of_class_of_part 成员的成员是 class_of_class_of_whole 实例的成员部件。

示例：毒性描述是材料数据表的 class_of_class_of_part,在车用汽油的材料安全数据表中“具有致癌物质”的描述是 class_of_part,而且车用汽油材料安全数据表的副本 #5 把“具有致癌物质”作为部件。

EXPRESS 描述:

```

* )
ENTITY class_of_class_of_composition
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_class_of_part : class_of_class_of_individual;
    class_of_class_of_whole : class_of_class_of_individual;
END_ENTITY;
( *

```

属性定义:

class_of_class_of_part: 在 class_of_class_of_composition 中是 class_of_class_of_part 的 class_of_class_of_individual。

class_of_class_of_whole: 在 class_of_class_of_composition 中是 class_of_class_of_whole 的 class_of_class_of_individual。

5.2.7.4 个体类的类(class_of_class_of_individual)

class_of_class_of_individual 是其成员是 class_of_individual 实例的 class_of_class。

示例：“保险产品”是 class_of_class_of_individual,它把“车用汽油”作为成员。

EXPRESS 描述:

```

* )
ENTITY class_of_class_of_individual
    SUPERTYPE OF (ONEOF(class_of_class_of_information_representation,
                        class_of_property, class_of_status,
                        shape_dimension))
    SUBTYPE OF(class_of_class);
END_ENTITY;
( *

```

5.2.7.5 个体成分类

class_of_composition_of_individual(个体成分类)是其成员是 composition_of_individual 成员的 class_of_relationship。

示例：一堆沙子的部件可以是一颗颗沙子是 class_of_composition_of_individual 的例子。

EXPRESS 描述:

```

* )
ENTITY class_of_composition_of_individual
    SUPERTYPE OF (ONEOF(class_of_arrangement_of_individual,
                        class_of_temporal_whole_part,

```

```

class_of_participation))
SUBTYPE OF(class_of_relationship);
class_of_part ; class_of_individual;
class_of_whole ; class_of_individual;
END_ENTITY;

```

(*

属性定义:

class_of_part; 在 class_of_composition_of_individual 中是 class_of_part 的 class_of_individual。

class_of_whole; 在 class_of_composition_of_individual 中是 class_of_whole 的 class_of_individual。

5.2.7.6 事件类

class_of_event(事件类)是其成员是 event 成员的 class_of_individual。

A is a whose members are members of。

示例: 连续的和瞬间的是实例。连续性事件就像流域界流过一根管子。

EXPRESS 描述:

(*

```

ENTITY class_of_event
SUBTYPE OF(class_of_individual);
END_ENTITY;

```

(*

5.2.7.7 特征整体部件类

class_of_feature_whole_part(特征整体部件类)是其成员是 feature_whole_part 实例的 class_of_arrangement_of_individual。

示例: 热电偶套管有茎干,并在顶部有表是 class_of_feature_whole_part 的例子。

EXPRESS 描述:

(*

```

ENTITY class_of_feature_whole_part
SUBTYPE OF(class_of_arrangement_of_individual);
END_ENTITY;

```

(*

5.2.7.8 个体类

class_of_individual(个体类)是一个类,其成员是 possible_individual 的实例。

示例: 可以用 class_of_individual 的实例表达类“工程师”(在工程原理和实践方面有资格或熟练的人)。

EXPRESS 描述:

(*

```

ENTITY class_of_individual
SUPERTYPE OF (ONEOF(
class_of_event,
class_of_arranged_individual,
class_of_period_in_time,
individual_dimension,
property,
status
)
ANDOR participating_role_and_domain)

```

```

    SUBTYPE OF(class);
END_ENTITY;
( *

```

5.2.7.9 时域类

class_of_period_in_time(时域类)是 class_of_individual,其成员是 period_in_time 的实例。

示例: 星期一和六月是 class_of_period_in_time 的例子。

EXPRESS 描述:

```

* )
ENTITY class_of_period_in_time
    SUBTYPE OF(class_of_individual);
END_ENTITY;
( *

```

5.2.7.10 时间点类

class_of_point_in_time(时间点类)是 class_of_event,其成员是 point_in_time 的成员。

示例: 午夜是 class_of_point_in_time。

EXPRESS 描述:

```

* )
ENTITY class_of_point_in_time
    SUBTYPE OF(class_of_event);
END_ENTITY;
( *

```

5.2.7.11 状态类

class_of_status(状态类)是 class_of_class_of_individual,其成员是 status。

示例: class_of_status 的例子是正式批准的,成员有:没有评估、经批准、被拒绝。

EXPRESS 描述:

```

* )
ENTITY class_of_status
    SUBTYPE OF(class_of_class_of_individual);
END_ENTITY;
( *

```

5.2.7.12 临时整体部件类

class_of_temporal_whole_part(临时整体部件类)是 class_of_composition_of_individual,其成员是 temporal_whole_part 的成员。

示例: 可以用 class_of_temporal_whole_part 的实例表达指示原油蒸馏设备可以有最大石脑油模式的类。

EXPRESS 描述:

```

* )
ENTITY class_of_temporal_whole_part
    SUBTYPE OF(class_of_composition_of_individual);
END_ENTITY;
( *

```

5.2.7.13 状态

status 是用离散无序值描述特征或质量的 class_of_individual。

示例: 可以用 status 的实例表达像“公开的”、“着色的”、“经批准的”、“老的”、“新的”、“用旧的”、“冒险的”、“安全的”、“危险的”、“高兴的”、“悲伤的”、“生锈的”的所有类。

注: 公开或着色的程度用 property 的实例表达,而不是 status 的实例。

EXPRESS 描述:

*)

ENTITY status

SUBTYPE OF(class_of_individual);

END_ENTITY;

(*

5.2.8 已安排个体的类

本条声明了表达已安排个体类的实体数据类型。

注: 图 184 是本条定义的实体数据类型的图(见 4.7.9 和 4.8.4.1)。

5.2.8.1 已安排个体类

class_of_arranged_individual(已安排个体类)是 class_of_individual,其成员是组件的安排。

示例: Robocop 是 class_of_arranged_individual,它有一些是 class_of_inanimate_physical_object 成员的部件,也有一些是 class_of_organism 成员的部件。

注 1: 比方说,某些子类上的 EXPRESS ONEOF(互相排斥)约束条件不能阻止特殊 possible_individual 成为用 class_of_biological_matter 分类的特殊 arranged_individual 的成员和特殊 class_of_composite_material 的成员。它仅仅是类本身(不是多个实体类型的成员)。

注 2: 有用对象的说明和描述通常是多个安排类的相互交叉,并允许进行形状和材料方面的约束。在本部分中,这种交叉是 class_of_arranged_individual、class_of_feature、class_of_inanimate_physical_object、class_of_organization、class_of_activity、class_of_organism 或 class_of_information_object 的成员。

EXPRESS 描述:

*)

ENTITY class_of_arranged_individual

SUPERTYPE OF (ONEOF(

class_of_atom,

class_of_biological_matter,

class_of_composite_material,

class_of_compound,

class_of_functional_object,

class_of_information_presentation,

class_of_information_representation,

class_of_molecule,

class_of_particulate_material,

class_of_sub_atomic_particle,

crystalline_structure,

phase)

ANDOR class_of_organization

ANDOR class_of_activity

ANDOR class_of_information_object

ANDOR class_of_feature

ANDOR ONEOF(class_of_organism,

class_of_inanimate_physical_object))

SUBTYPE OF(class_of_individual);

END_ENTITY;

(*

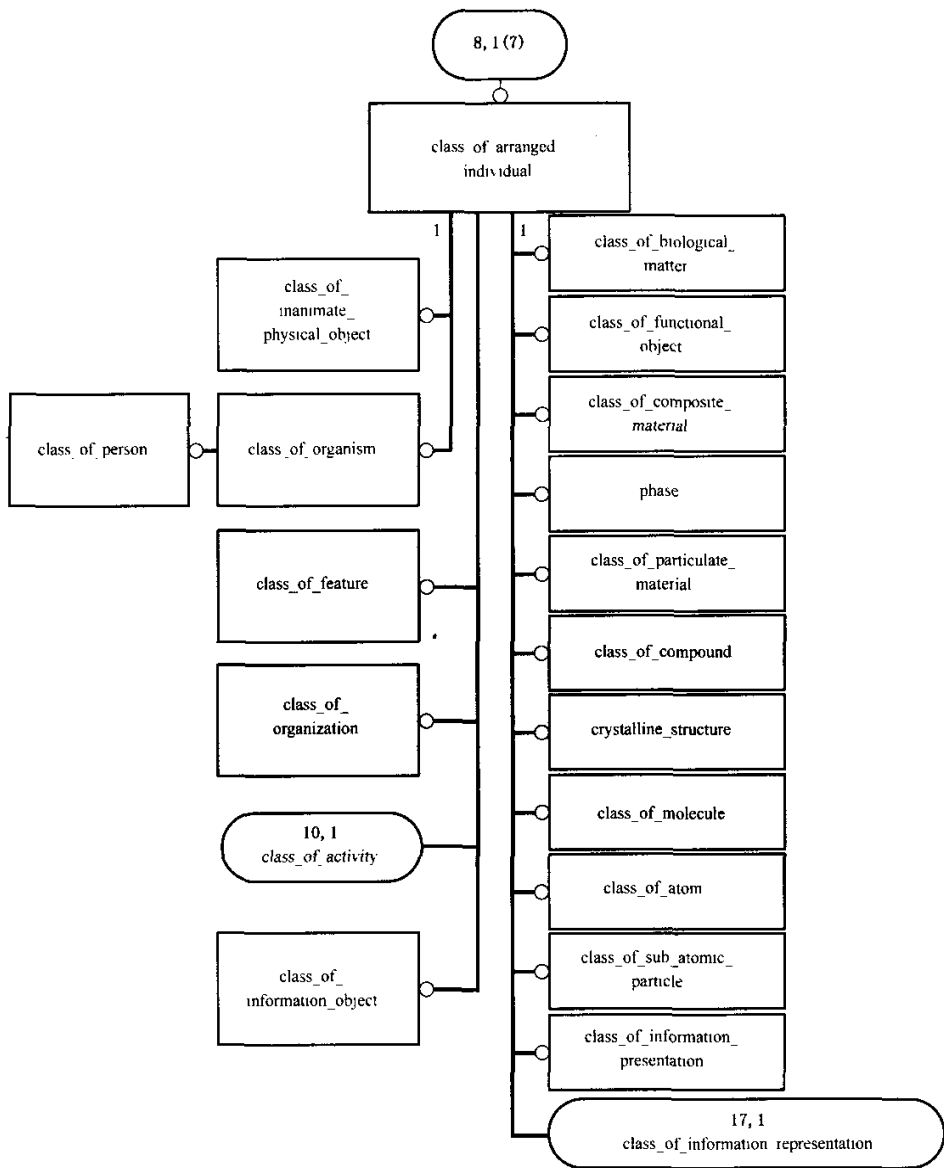


图 184 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 8 个)

5.2.8.2 原子类

class_of_atom(原子类)是 class_of_arranged_individual,其成员是原子。

示例: 可以用 class_of_atom 的实例表达元素周期表上的所有元素。

EXPRESS 描述:

*)

ENTITY class_of_atom

SUBTYPE OF(class_of_arranged_individual);

END_ENTITY;

5.2.8.3 生物物质类

class_of_biological_matter(生物物质类)是 class_of_arranged_individual,其成员是细胞或细胞集

合的特殊类型。

示例：可以用 class_of_biological_matter 的实例表达“血液”、“酶”和“血浆”类。

EXPRESS 描述：

```

* )
ENTITY class_of_biological_matter
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.4 成分材料类

class_of_composite_material(成分材料类)是 class_of_arranged_individual,其成员具有通用的可分类混合物的安排。

示例 1：可以用 class_of_composite_material 的实例表达像夹板、玻璃纤维和碳纤维这样的层压物。

示例 2：可以用 class_of_composite_material 的实例表达木材、肌肉和皮肤。

EXPRESS 描述：

```

* )
ENTITY class_of_composite_material
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.5 混合物类

class_of_compound(混合物类)是 class_of_arranged_individual,其成员由相同或不同类型的分子安排(通过分子间作用力限制在一起)组成。这包括混合物和合金。

示例：可以用 class_of_compound 的实例表达水、硫酸、沙子、石灰石和钢。

EXPRESS 描述：

```

* )
ENTITY class_of_compound
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.6 特征类

class_of_feature(特征类)是 class_of_arranged_individual,其成员是某些 possible_individual 的连接、不可分隔的部件,并且有一个完整定义的边界。

示例：“山脉”、“凹槽”、“边缘”、“喷嘴”、“鼻子”和“凸面”都可以表达成 class_of_feature 的实例。

EXPRESS 描述：

```

* )
ENTITY class_of_feature
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.7 功能对象类

class_of_functional_object(功能对象类)是 class_of_arranged_individual,它指出了对象的功能或用途。

示例：泵、阀和汽车是 class_of_functional_object 的例子。泵、阀和汽车等的特殊模型是 class_of_inanimate_

physical_object 的实例,它是这些 class_of_functional_object 实例的特殊化。

EXPRESS 描述:

```

* )
ENTITY class_of_functional_object
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.8 无生命物理对象类

class_of_inanimate_physical_object(无生命物理对象类)是 class_of_arranged_individual,其成员不是活的。

示例:可以用 class_of_inanimate_physical_object 的实例表达“石油”类。

EXPRESS 描述:

```

* )
ENTITY class_of_inanimate_physical_object
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.9 信息对象类

class_of_information_object(信息对象类)是 class_of_arranged_individual,其成员是零或多个 class_of_information_representation 和零或多个 class_of_information_presentation 的成员。

注:通常,它是分类为 class_of_information_object 的 physical_object(例如纸质文档)。

示例:“报纸”是 class_of_information_object。

EXPRESS 描述:

```

* )
ENTITY class_of_information_object
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.10 信息表达类

class_of_information_presentation(信息表达类)是区别表达信息类型的 class_of_arranged_individual。

示例:可以用 class_of_information_presentation 的实例表达字符类型粗体、斜体、Times New Roman 和 16pt。

EXPRESS 描述:

```

* )
ENTITY class_of_information_presentation
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.11 分子类

class_of_molecule(分子类)是 class_of_arranged_individual,其成员是分子。

示例:可以用 class_of_molecule 的实例表达 H₂O、H₂SO₄ 和 DNA。

EXPRESS 描述:

```

* )

```

```

ENTITY class_of_molecule
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.12 生物类

class_of_organism(生物类)是 class_of_arranged_individual,其成员是活的生物。

示例:人类、绵羊、蚯蚓、橡树和细菌是 class_of_organism 的实例。

EXPRESS 描述:

```

* )
ENTITY class_of_organism
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.13 组织类

class_of_organization(组织类)是 class_of_arranged_individual,其成员是 physical_object 的实例(由临时的人员和其他资产组成),并具有特殊的目的。

示例:可以用 class_of_organization 的实例表达公司、政府或项目团队。

EXPRESS 描述:

```

* )
ENTITY class_of_organization
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.14 特殊材料类

class_of_particulate_material(特殊材料类)是 class_of_arranged_individual,其成员是已安排的大量超分子大小对象。

示例:沙堆、沙子和水泥混合物、袋装螺钉、反应堆催化剂是 class_of_particulate_material 的例子。

EXPRESS 描述:

```

* )
ENTITY class_of_particulate_material
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.8.15 人类

class_of_person(人类)是 class_of_organism,其成员是人。

示例:可以用 class_of_person 的实例表达工程师、工厂经理、学生、男人、女人、老年人、成人、女孩或男孩。也可以用 class_of_functional_object 的实例表达工程师、工厂经理或学生。

EXPRESS 描述:

```

* )
ENTITY class_of_person
    SUBTYPE OF(class_of_organism);
END_ENTITY;
( *

```

5.2.8.16 亚原子颗粒类

class_of_sub_atomic_particle(亚原子颗粒类)是 class_of_arranged_individual,其成员是原子的组成颗粒。

示例:可以用的实例表达质子、电子、介子、中子、正电子、 μ 介子、夸克和微中子。

EXPRESS 描述:

```
*)
ENTITY class_of_sub_atomic_particle
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
(*
```

5.2.8.17 晶体结构

crystalline_structure(晶体结构)是 class_of_arranged_individual,它是一种构成,许多简单元素和其天然混合物通过自然吸引力的作用有规律地集合在这种构成中,它具有明确的内在结构,许多对称性排列的平面包围了实体的外部形状,并且可以从简单的立方体变成复杂得多的几何实体。

示例:铁素体、马氏体和奥氏体是 crystalline_structure 的例子。

EXPRESS 描述:

```
*)
ENTITY crystalline_structure
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
(*
```

5.2.8.18 相位

phase(相位)是基于材料边界行为本性的,导致其原子和分子结合的 class_of_arranged_individual。

注: phase 不包括内部结构类型,例如晶体。

示例:可以用 phase 的实例表达“液相”和“固相”类。

EXPRESS 描述:

```
*)
ENTITY phase
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
(*
```

5.2.9 活动和事件

本条声明了表达活动和事件的实体数据类型。

注:图 185 是本条定义的实体数据类型图(见 4.7.10 和 4.7.17)。

5.2.9.1 活动

activity 是 possible_individual,它通过激发标识 possible_individual 开始,或标识 possible_individual 结束的事件产生变化。

活动由参与活动的那些 possible_individual 成员的临时部件组成。通过指示临时部件在活动中所扮演角色的 participating_role_and_domain 对参与的临时部件进行分类。

示例:可以用 activity 的实例表达用机械泵泵送液体。

EXPRESS 描述:

```
*)
ENTITY activity
```

```

SUBTYPE OF(possible_individual);
END_ENTITY;
( *

```

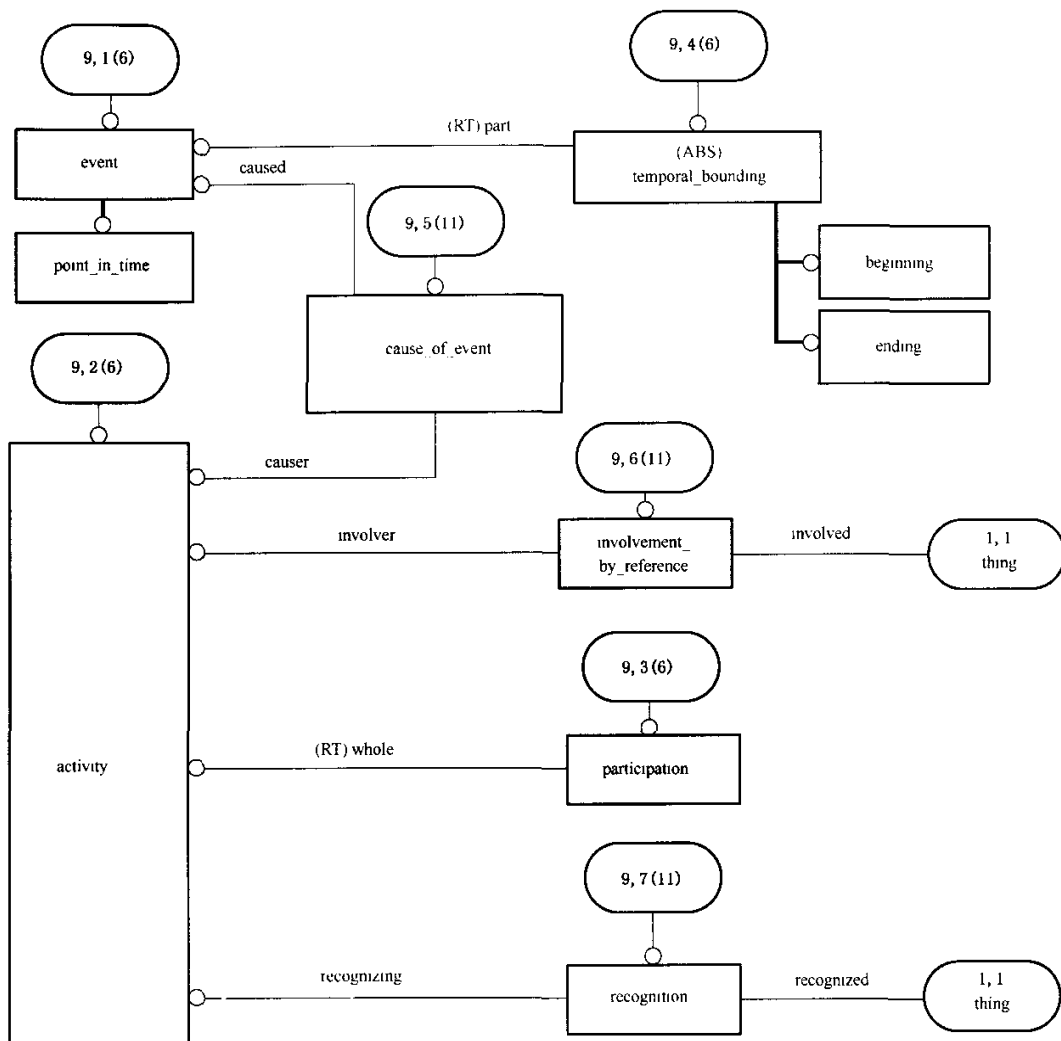


图 185 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 9 个)

5.2.9.2 开始

beginning(开始)是标识 possible_individual 开始时间的 temporal_bounding。

示例 1: 可以用 beginning 的实例表达世界标准时间 1999 年 7 月 1 日零点 point_in_time 是世界标准时间 1999 年 7 月 period_in_time 开始的关系。

示例 2: 可以用 beginning 的实例表达 event“装载完成”标识 possible_individual “装载车间空闲”开始的关系。

EXPRESS 描述:

*)

ENTITY beginning

SUBTYPE OF(temporal_bounding);

END_ENTITY;

(*

5.2.9.3 事件原因

cause_of_event(事件原因)是 relationship,它指出了引起者 activity 所引起的 event。

示例: 可以用 cause_of_event 的实例表达 event“油罐内液体水平面已满”引起油轮装载活动的关系。

EXPRESS 描述:

```
*)  
ENTITY cause_of_event  
  SUBTYPE OF(relationship);  
  caused : event;  
  causer : activity;  
END_ENTITY;
```

(*

属性定义:

caused: cause_of_event 中引起的 event。

causer: 在 cause_of_event 中是引起者的 activity。

• 5.2.9.4 结束

ending(结束)是标识 possible_individual 结束的 temporal_bounding。

示例 1: 可以用 ending 的实例表达世界标准时间 1999 年 7 月 1 日零点 point_in_time 是世界标准时间 1999 年 6 月 period_in_time 结束的关系。

示例 2: 可以用 ending 的实例表达 event“装载完成”标识 possible_individual “装载车间操作阶段 1”(装载车间的临时部分)结束的关系。

EXPRESS 描述:

```
*)  
ENTITY ending  
  SUBTYPE OF(temporal_bounding);  
END_ENTITY;
```

(*

5.2.9.5 事件

event 是时间范围为零的 possible_individual。event 是一个或多个 possible_individual 的临时边界,虽然这些 possible_individual 可能没有任何知识。

示例: 连接泵的动力是标识泵临时开始部分的事件。

EXPRESS 描述:

```
*)  
ENTITY event  
  SUBTYPE OF(possible_individual);  
END_ENTITY;
```

(*

5.2.9.6 引用包含

involvement_by_reference(引用包含)是指示 thing 被引用到 activity 中的 relationship。

注: 本实体类型用于不是 possible_individual 直接 participation 的包含,例如类包含,或 possible_individual 以前或将来的临时部分。

示例: 引用罗马帝国的会话是一个通过 involvement_by_reference 连接到罗马帝国的活动。

EXPRESS 描述:

```
*)
```

```

ENTITY involvement_by_reference
    SUBTYPE OF (relationship);
    involved : thing;
    involver : activity;
END_ENTITY;

```

(*

属性定义:

involved: 引用 activity 中被引用的 thing。

involver: 引用 thing 被包含在其中的 activity。

5.2.9.7 参与

participation 是 composition_of_individual, 它显示 possible_individual 是 activity 中的一个参与者。

注: 在 participation 中是角色的 possible_individual 可以是 whole_life_individual 的临时部分, 用显示其在 activity 中所扮演角色的 role_and_domain 对 whole_life_individual 进行分类。

示例: P1234 的临时部分(在 2002 年 12 月 2 日完成发动机容器 Murex 的排泄任务)和排泄容器活动之间的关系是参与关系。

EXPRESS 描述:

*)

```

ENTITY participation
    SUBTYPE OF (composition_of_individual);
    SELF\composition_of_individual.whole : activity;
END_ENTITY;

```

(*

属性定义:

whole: 在 participation 中是整体的 activity。

5.2.9.8 时间点

point_in_time 是时间范围为零、整个空间范围的 event。

注: 为了使用本部分, 应该用 representation_of_Gregorian_date_and_UTC_time 表达 point_in_time。

示例: UTC 1999-05-13T16:31:23.56 时间是 point_in_time。

EXPRESS 描述:

*)

```

ENTITY point_in_time
    SUBTYPE OF (event);
END_ENTITY;

```

(*

5.2.9.9 识别

recognition 是通过 activity 识别事物的 relationship。

示例: 测量活动 # 358 识别了房间是 20 °C 属性的成员。

EXPRESS 描述:

*)

```

ENTITY recognition
    SUBTYPE OF (relationship);
    recognized : thing;
    recognizing : activity;
END_ENTITY;

```

(*

属性定义:

recognized: activity 识别出的 thing。

recognizing: 导致 recognition 结果的 activity。

5.2.9.10 临时边界

temporal_bounding(临时边界)是 assembly_of_individual, 它显示了局部 event 是整个 possible_individual 的临时边界。

EXPRESS 描述:

*)

ENTITY temporal_bounding

ABSTRACT SUPERTYPE OF (ONEOF(ending, beginning))

SUBTYPE OF(composition_of_individual);

SELF\composition_of_individual. part : event;

END_ENTITY;

(*

属性定义:

part: 在 temporal_bounding 中是局部的 event。

5.2.10 活动种类

本条声明了表达活动类的实体数据类型。

注: 图 186 是本条定义的实体数据类型图(见 4.8.4.9)。

5.2.10.1 活动类

class_of_activity(活动类)是 class_of_arranged_individual, 其成员是 activity 的实例。

示例: 可以用的实例表达钻孔、蒸馏和批准。

注: 在有前提且 class_of_activity 是那些前提反应的地方(例如对触摸热表面的反应), 或在用某些属性或功能描述活动产生方式的地方(用液体的黏性描述液体流动性), 行为是描述 class_of_activity 的术语。

EXPRESS 描述:

*)

ENTITY class_of_activity

SUBTYPE OF(class_of_arranged_individual);

END_ENTITY;

(*

5.2.10.2 个体类开始原因的类

class_of_cause_of_beginning_of_class_of_individual(个体类开始原因的类)是 class_of_relationship, 它显示 class_of_activity 的成员激发 class_of_individual 的成员开始。

示例: 汽车制造活动激发汽车生命的开始。

EXPRESS 描述:

*)

ENTITY class_of_cause_of_beginning_of_class_of_individual

SUBTYPE OF(class_of_relationship);

class_of_begun : class_of_individual;

class_of_causer : class_of_activity;

END_ENTITY;

(*

属性定义:

class_of_begun: class_of_activity 成员生成的 class_of_individual。

class_of_causer: 其成员激发 class_of_individual 成员开始的 class_of_activity。

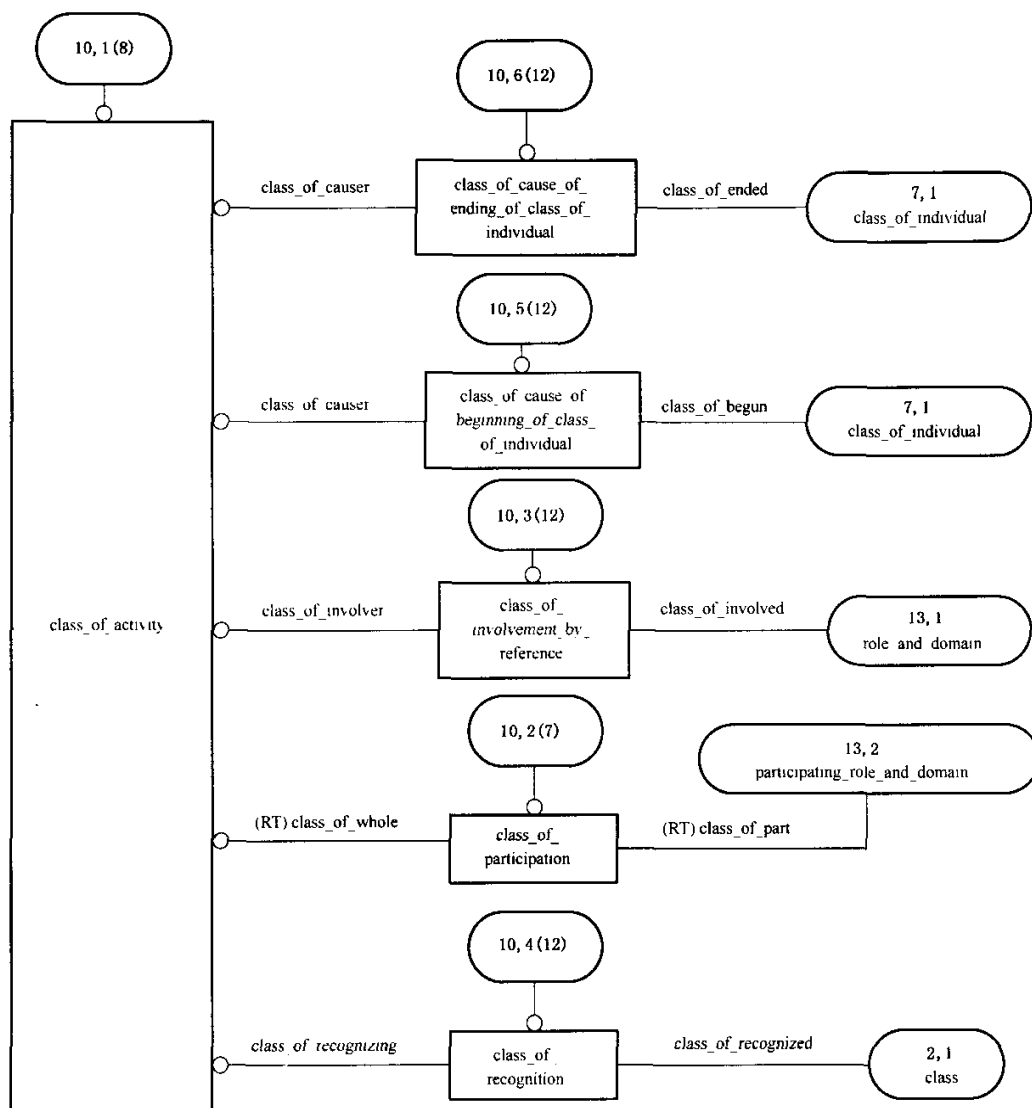


图 186 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 10 个)

5.2.10.3 个体类结束原因的类

class_of_cause_of_ending_of_class_of_individual(个体类结束原因的类)是 class_of_relationship, 它显示 class_of_activity 的成员激发 class_of_individual 的成员结束。

示例：汽车的压碎活动导致汽车生命的结束。

EXPRESS 描述：

```

* )
ENTITY class_of_cause_of_ending_of_class_of_individual
  SUBTYPE OF(class_of_relationship);
  class_of_causer ; class_of_activity;
  class_of_ended ; class_of_individual;
END_ENTITY;
( *
    
```


属性定义:

class_of_causer;其成员激发 class_of_individual 成员生命结束的 class_of_activity。

class_of_ended;class_of_activity 成员结束其成员的 class_of_individual。

5.2.10.4 引用包含类

class_of_involvement_by_reference(引用包含类)是 class_of_relationship,其成员是实例 involvement_by_reference。

示例:讨论历史活动是 class_of_involvement_by_reference 的例子。

EXPRESS 描述:

*)

ENTITY class_of_involvement_by_reference

SUBTYPE OF(class_of_relationship);

class_of_involved ; role_and_domain;

class_of_involver ; class_of_activity;

END_ENTITY;

(*

属性定义:

class_of_involved;具有 class_of_involvement_by_reference 的 role_and_domain。

class_of_involver;具有 class_of_involvement_by_reference 的 class_of_activity。

5.2.10.5 参与类

class_of_participation(参与类)是 class_of_composition_of_individual,它显示 participating_role_and_domain 的实例成员参与了 class_of_activity 的实例成员。

示例:“演奏会指挥者”是 class_of_participation 的一个例子。

EXPRESS 描述:

*)

ENTITY class_of_participation

SUBTYPE OF(class_of_composition_of_individual);

SELF\class_of_composition_of_individual.class_of_part ;

participating_role_and_domain;

SELF\class_of_composition_of_individual.class_of_whole ;

class_of_activity;

END_ENTITY;

(*

属性定义:

class_of_part;具有 class_of_participation 的 participating_role_and_domain。

class_of_whole;具有 class_of_participation 的 class_of_activity。

5.2.10.6 识别类

class_of_recognition(识别类)是 class_of_relationship,它显示 class_of_activity 的成员可能导致 class 的成员识别。

示例:测量活动可能导致通过 property 识别 possible_individual 的 classification。

EXPRESS 描述:

*)

ENTITY class_of_recognition

SUBTYPE OF(class_of_relationship);

```

class_of_recognized : class;
class_of_recognizing : class_of_activity;
END_ENTITY;
( *

```

属性定义:

class_of_recognized:通过 class_of_activity 成员识别其成员的 class。
class_of_recognizing:其成员实现识别 class 的 class_of_activity。

5.2.11 关系

本条声明了表达关系的实体数据类型。

注:图 187 是本条定义的实体数据类型图(见 4.6.4 和 4.10.1)。

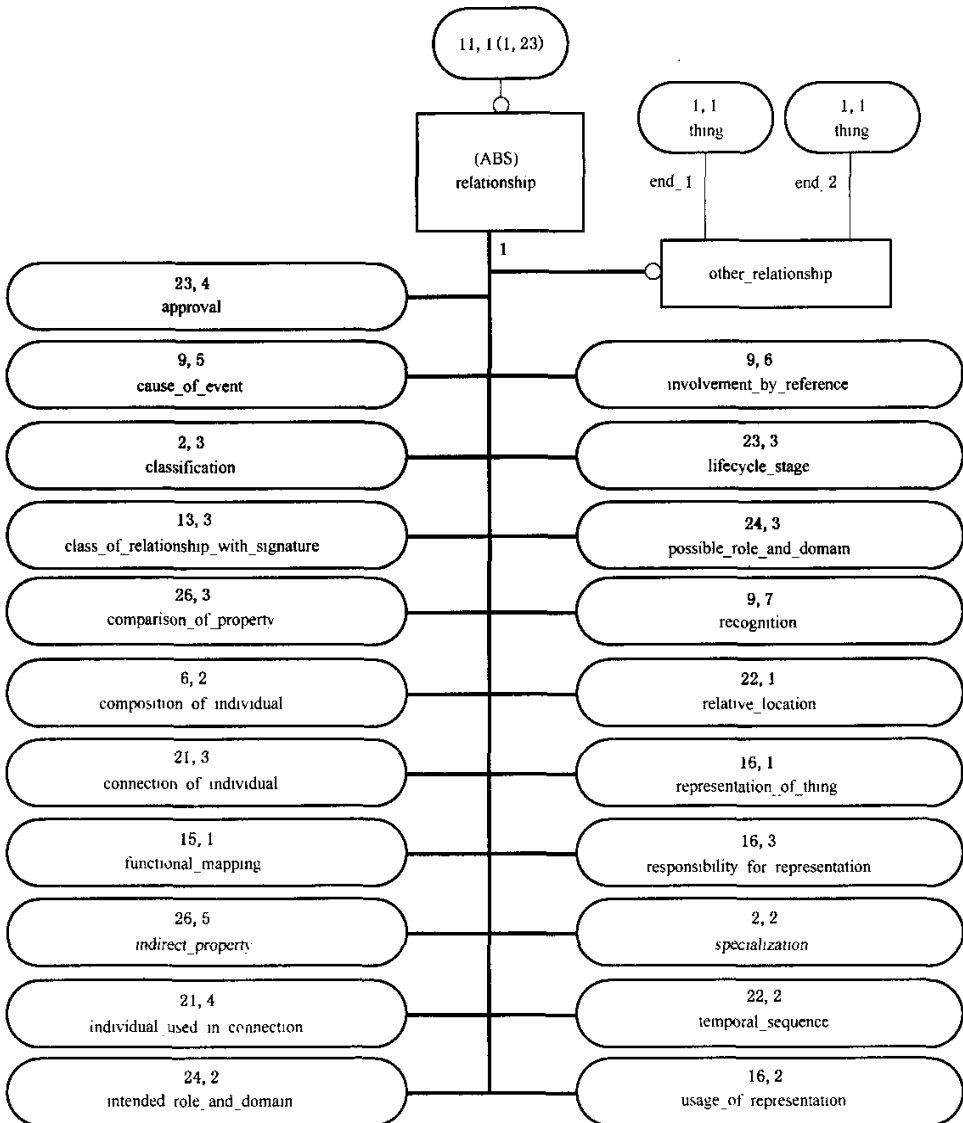


图 187 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 11 个)

5.2.11.1 其他关系

other_relationship(其他关系)是 relationship,它不是任何其他 relationship 显示子类的成员。通过 class_of_relationship_with_signature 的实例,classification 规定了 other_relationship 的含义。

示例: relationship 指出可以用 other_relationship 的实例表达福特制造的汽车。

分别通过 class_of_relationship_with_signature(对 other_relationship 进行分类)的 class_of_end_1 和 class_of_end_2 属性给出对 end_1 和 end_2 属性进行分类的 role_and_domain。在 class_of_relationship_with_signature 也是 class_of_relationship_with_related_end_1 或 class_of_relationship_with_related_end_2 的地方,other_relationship 的 end_1 或 end_2 应该分别取相关属性规定的值。

EXPRESS 描述:

```

* )
ENTITY other_relationship
    SUBTYPE OF (relationship);
    end_1 : thing;
    end_2 : thing;
END_ENTITY;
( *

```

属性定义:

end_1: 相关 thing 实例的第一个实例。

end_2: 相关 thing 实例的第二个实例。

5.2.11.2 关系

relationship 是指出一个事物必须处理另一个事物的 abstract_object。

注: 只支持二元关系的类。可以使用 multidimensional_object 和 class_of_multidimensional_object 指出更复杂的对象。

EXPRESS 描述:

```

* )
ENTITY relationship
    ABSTRACT SUPERTYPE OF (ONEOF(
        approval,
        cause_of_event,
        class_of_relationship_with_signature,
        classification,
        comparison_of_property,
        composition_of_individual,
        connection_of_individual,
        functional_mapping,
        indirect_property,
        individual_used_in_connection,
        intended_role_and_domain,
        involvement_by_reference,
        lifecycle_stage,
        other_relationship,
        possible_role_and_domain,
        recognition,
        relative_location,
        representation_of_thing,

```

responsibility_for_representation,
 specialization,
 temporal_sequence,
 usage_of_representation))

SUBTYPE OF(abstract_object);

END_ENTITY;

(*

5.2.12 关系类型

本条声明了表达关系类型的实体数据类型。

注：图 188 是本条定义实体数据类型图（见 4.8.3.3）。

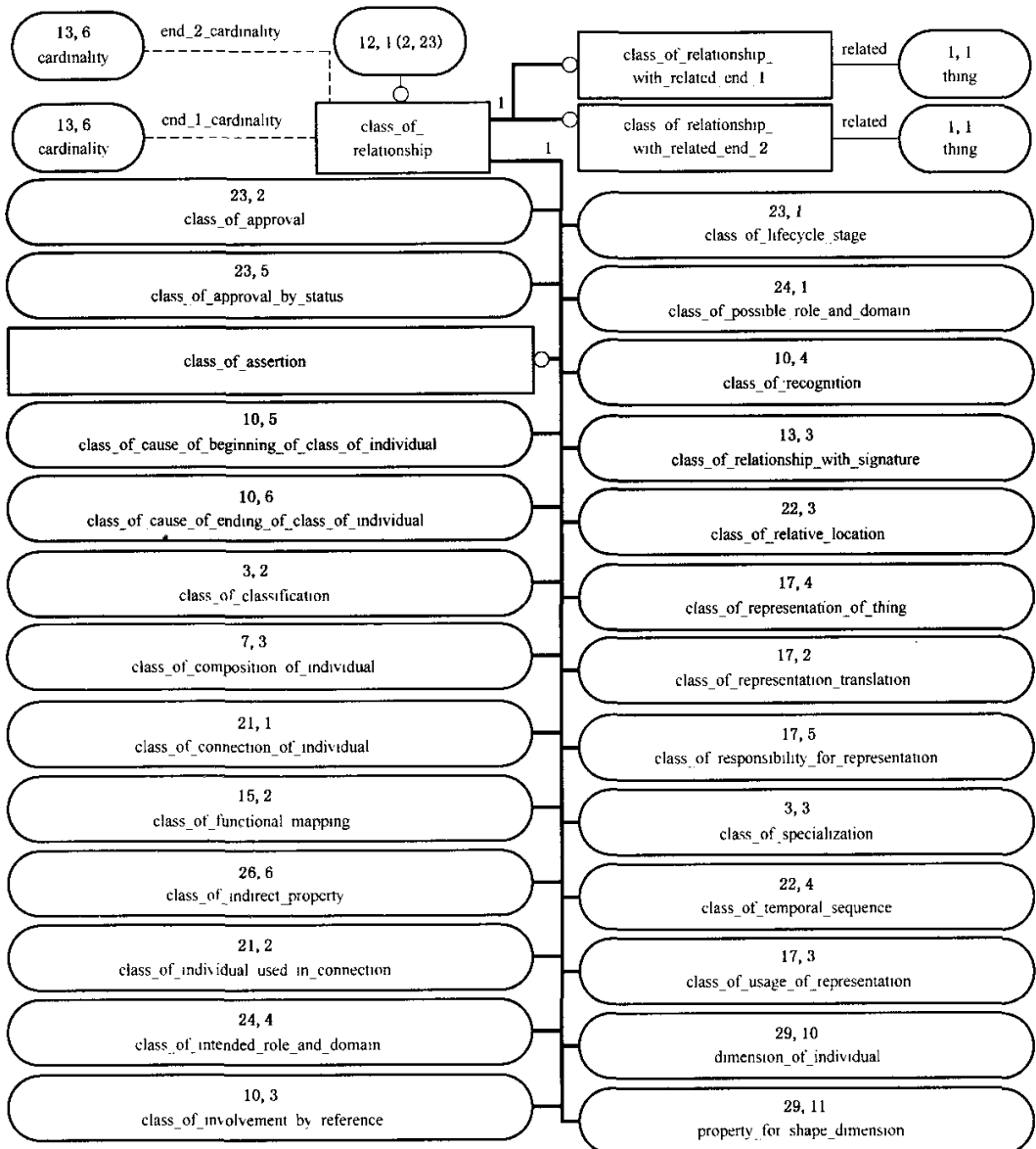


图 188 生命周期集成模式的 EXPRESS-G 图 (29 个图中的第 12 个)

5.2.12.1 断言类

class_of_assertion(断言类)是描述成员关系断定本质的 class_of_relationship。

示例：可以用 class_of_assertion 的实例表达断言、否认和可能。

EXPRESS 描述：

```
* )
ENTITY class_of_assertion
    SUBTYPE OF(class_of_relationship);
END_ENTITY;
( * .
```

5.2.12.2 关系类

class_of_relationship(关系类)是 class_of_abstract_object,其成员是 relationship 的成员。

EXPRESS 描述：

```
* )
ENTITY class_of_relationship
    SUPERTYPE OF (ONEOF(
        class_of_approval,
        class_of_approval_by_status,
        class_of_cause_of_beginning_of_class_of_individual,
        class_of_cause_of_ending_of_class_of_individual,
        class_of_classification,
        class_of_composition_of_individual,
        class_of_connection_of_individual,
        class_of_functional_mapping,
        class_of_indirect_property,
        class_of_individual_used_in_connection,
        class_of_intended_role_and_domain,
        class_of_involvement_by_reference,
        class_of_lifecycle_stage, class_of_assertion,
        class_of_possible_role_and_domain,
        class_of_recognition,
        class_of_relationship_with_signature,
        class_of_relative_location,
        class_of_representation_of_thing,
        class_of_representation_translation,
        class_of_responsibility_for_representation,
        class_of_specialization,
        class_of_temporal_sequence,
        class_of_usage_of_representation,
        dimension_of_individual,
        property_for_shape_dimension
    )
    ANDOR
    ONEOF(class_of_relationship_with_related_end_1,
```

```

class_of_relationship_with_related_end_2))
SUBTYPE OF(class_of_abstract_object);
end_1_cardinality : OPTIONAL cardinality;
end_2_cardinality : OPTIONAL cardinality;
END_ENTITY;

```

(*

属性定义:

end_1_cardinality: class_of_relationship 第一个属性的最大和最小基数。如果没有规定基数,那么对基数没有约束。

end_2_cardinality: class_of_relationship 第二个属性的最大和最小基数。如果没有规定基数,那么对基数没有约束。

5.2.12.3 带相关端 1 的关系类

class_of_relationship_with_related_end_1(带相关端 1 的关系类)是 class_of_relationship,在那里关联到 class_of_relationship 中的特殊 thing,而不是 class 的成员。相关的 thing 扮演 class_of_end_1 指示的 role_and_domain。

示例:用 Bloggs & Co 制造的产品是相关 thing 指向 Bloggs & Co 的 class_of_relationship。

EXPRESS 描述:

(*

```

ENTITY class_of_relationship_with_related_end_1
SUBTYPE OF(class_of_relationship);
related : thing;
END_ENTITY;

```

(*

属性定义:

related:相关的特殊 thing,并且不是其可能引用类的某些成员。

5.2.12.4 带相关端 2 的关系类

class_of_relationship_with_related_end_2(带相关端 2 的关系类)是 class_of_relationship,在那里关联到 class_of_relationship 中的特殊 thing,而不是 class 的成员。相关的 thing 扮演 class_of_end_2 指示的 role_and_domain。

示例:普通人拥有的焊接技术是 class_of_relationship_with_related_end_2 的一个例子,普通人是相关的事物。

EXPRESS 描述:

(*

```

ENTITY class_of_relationship_with_related_end_2
SUBTYPE OF(class_of_relationship);
related : thing;
END_ENTITY;

```

(*

属性定义:

related:相关的特殊 thing,并且不是其可能引用类的某些成员。

5.2.13 角色和域

本条声明了表达角色和域的实体数据类型。

注:图 189 是本条定义的实体数据类型图(见 4.8.4.8)。

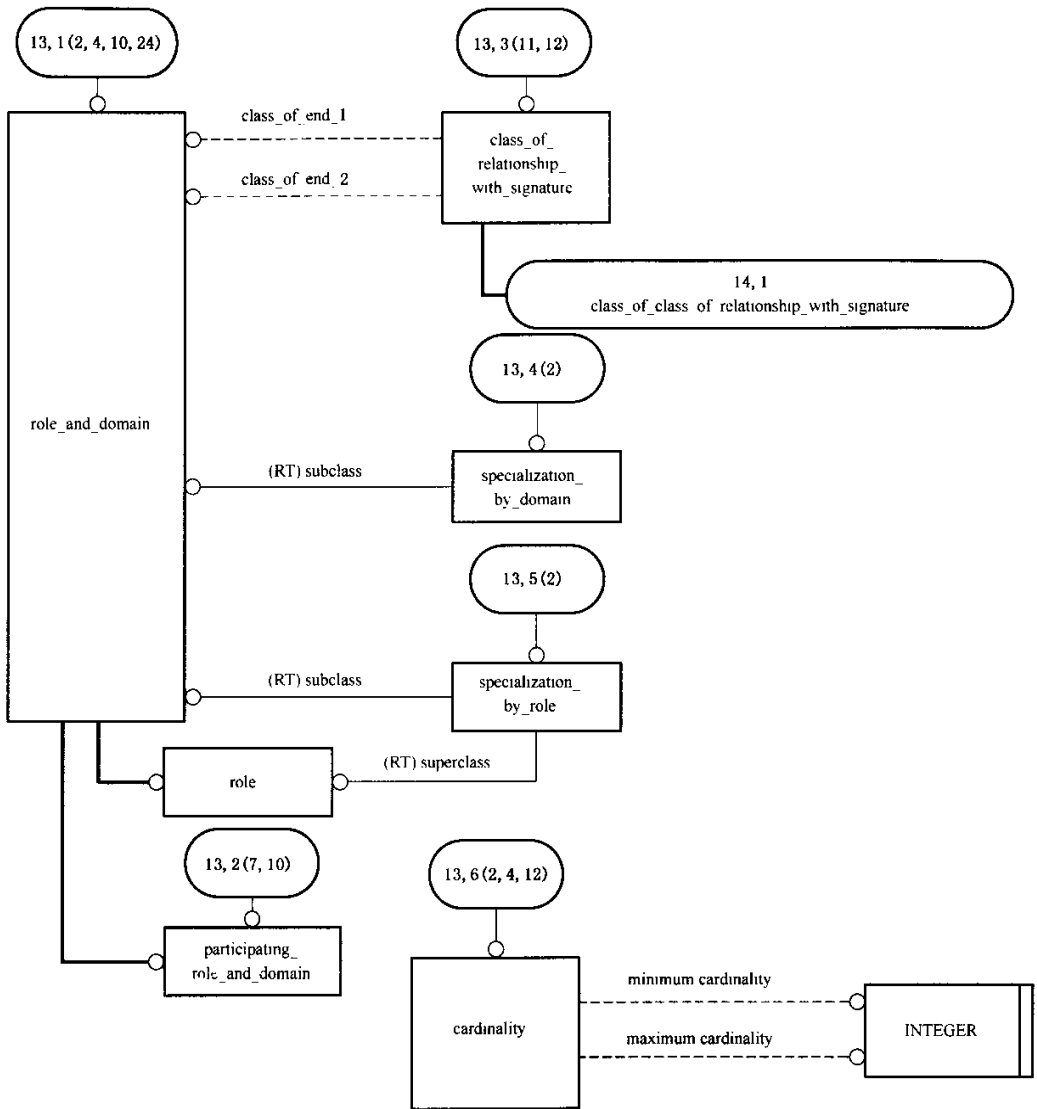


图 189 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 13 个)

5.2.13.1 基数

cardinality(基数)是一个事物可以在 class_of_relationship 或 class_of_multidimensional_object 中扮演一个特殊角色的最大和/或最小时间的 class。

示例：1 的最小值和 1 的最大值意味着每个对象的这种类型恰好有一个 relationship 或 multidimensional_object。

EXPRESS 描述：

*)

ENTITY cardinality

SUBTYPE OF(class);

maximum_cardinality : OPTIONAL INTEGER;

minimum_cardinality : OPTIONAL INTEGER;

END_ENTITY;

(*

属性定义:

maximum_cardinality:域成员可以参与规定角色的最大时间值。如果没有规定 maximum_cardinality,那么没有最大值约束。

注 1: maximum_cardinality 的通用值是 1 和许多,许多是没有规定值的结果。

minimum_cardinality:minimum_cardinality 是域成员可以参与规定角色的最小时间值。如果没有规定 minimum_cardinality,该值应该取零。

注 2: minimum_cardinality 的通用值是零和一。

5.2.13.2 class_of_relationship_with_signature

class_of_relationship_with_signature 是可能有 role_and_domain(为每个结束指定的)的 class_of_relationship。

注: class_of_relationship_with_signature 类似于简单的 EXPRESS 属性和其反属性。可以用 multidimensional_object 和 class_of_multidimensional_object 对更复杂的对象进行建模。

示例:“已婚”是一个 class_of_relationship,其中 class_of_end_1 是 role_and_domain “丈夫”,且 class_of_end_2 是 role_and_domain “妻子”。

EXPRESS 描述:

*)

```
ENTITY class_of_relationship_with_signature
  SUBTYPE OF(class_of_relationship, relationship);
  class_of_end_1 : OPTIONAL role_and_domain;
  class_of_end_2 : OPTIONAL role_and_domain;
END_ENTITY;
```

(*

属性定义:

class_of_end_1:class_of_relationship 成员的 end_1 属性的规定。

class_of_end_2:class_of_relationship 成员的 end_2 属性的规定。

5.2.13.3 participating_role_and_domain

participating_role_and_domain 是 role_and_domain,它还是显示 activity 参与角色的 class_of_individual。

示例:“表演者”和“抽水机”是 participating_role_and_domain 的例子。

EXPRESS 描述:

*)

```
ENTITY participating_role_and_domain
  SUBTYPE OF(role_and_domain, class_of_individual);
END_ENTITY;
```

(*

5.2.13.4 角色

role 是显示那些事物必须处理 activity、relationship 或 multidimensional_object 的 role_and_domain。

示例 1: 职工是 role,它显示人的那些临时部分必须处理职业关系。

示例 2: 抽水机是 role,它显示泵的那些临时部分必须处理抽水活动。

EXPRESS 描述:

*)


```
ENTITY role
    SUBTYPE OF(role_and_domain);
END_ENTITY;
( *
```

5.2.13.5 角色和域

role_and_domain(角色和域)是为 class_of_relationship 或 class_of_multidimensional_object 的结束规定域和角色的 class。

注: role_and_domain 类似规定了一个 EXPRESS 属性或其反属性。

示例:“丈夫和男人”和“妻子和女人”是 role_and_domain 的例子。

EXPRESS 描述:

```
* )
ENTITY role_and_domain
    SUBTYPE OF(class);
END_ENTITY;
( *
```

5.2.13.6 域规范

specialization_by_domain(域规范)是 specialization,它显示 role_and_domain 的成员是域 class 的 specialization。

示例:“制造公司”是“公司”域的 specialization。

EXPRESS 描述:

```
* )
ENTITY specialization_by_domain
    SUBTYPE OF(specialization);
    SELF\specialization.subclass : role_and_domain;
END_ENTITY;
( *
```

属性定义:

subclass:是 class 子类的 role_and_domain。

5.2.13.7 角色规范

specialization_by_role(角色规范)是 specialization,它显示 role_and_domain 是超类指定的 role。

示例:制造公司是制造商的角色规范。

EXPRESS 描述:

```
* )
ENTITY specialization_by_role
    SUBTYPE OF(specialization);
    SELF\specialization.subclass : role_and_domain;
    SELF\specialization.superclass : role;
END_ENTITY;
( *
```

属性定义:

subclass:在 specialization_by_role 中是子类的 role_and_domain。

superclass:在 specialization_by_role 中是超类的 role。

5.2.14 关系类类型

本条声明了表达关系类类型的实体数据类型。

注：图 190 是本条定义的实体数据类型图。

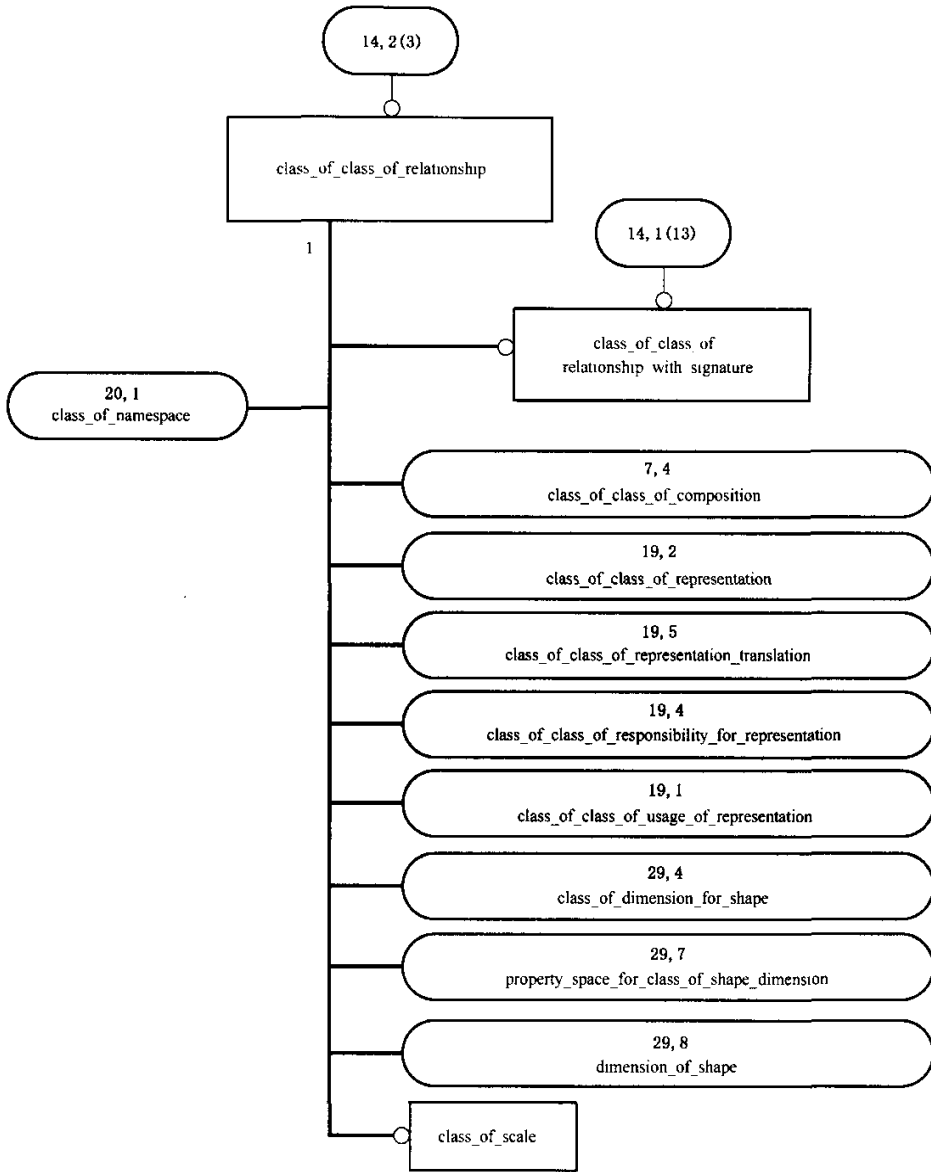


图 190 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 14 个)

5.2.14.1 关系类的类

class_of_class_of_relationship(关系类的类)是 class_of_class,其成员是 class_of_relationship 的实例。

示例：“反身代词”是 class_of_class_of_relationship 的一个例子。“反身代词”class_of_relationship 可能有扮演两个角色(其中其本身可以连接某些事物)的相同 thing,例如连接。

EXPRESS 描述:

*)

```

ENTITY class_of_class_of_relationship
  SUPERTYPE OF (ONEOF(
    class_of_class_of_composition,
    class_of_class_of_relationship_with_signature,
    class_of_class_of_representation,
    class_of_class_of_representation_translation,
    class_of_class_of_responsibility_for_representation,
    class_of_class_of_usage_of_representation,
    class_of_dimension_for_shape,
    class_of_namespace,
    class_of_scale,
    dimension_of_shape,
    property_space_for_class_of_shape_dimension
  ))
  SUBTYPE OF(class_of_class);
END_ENTITY;
( *

```

5.2.14.2 带签名的关系类的类

class_of_class_of_relationship_with_signature(带签名的关系类的类)是 class_of_class_of_relationship 和 class_of_relationship_with_signature。class_of_class_of_relationship_with_signature 的目的是允许把其他类型的关系类(在本部分中没有显示为实体数据类型)定义为引用数据。

示例: 继承(具有显示继承方向的角色)是一个 class_of_class_of_relationship_with_signature。class_of_relationship 是可以继承的,如果 A 关联 B,且 B 关联 C,那么 A 关联 C,所有的都是这样。

EXPRESS 描述:

```

( * )
ENTITY class_of_class_of_relationship_with_signature
  SUBTYPE OF(class_of_class_of_relationship,
    class_of_relationship_with_signature);
END_ENTITY;
( *

```

5.2.14.3 尺度(全文检索)类

class_of_scale(刻度类)是 class_of_class_of_relationship,其成员是 scale 的实例。

示例: 国际单位是 class_of_scale 的一个例子。

EXPRESS 描述:

```

( * )
ENTITY class_of_scale
  SUBTYPE OF(class_of_class_of_relationship);
END_ENTITY;
( *

```

5.2.15 函数

本条声明了表达函数的实体数据类型。

注: 图 191 是本条定义的实体数据类型图(见 4.9)。

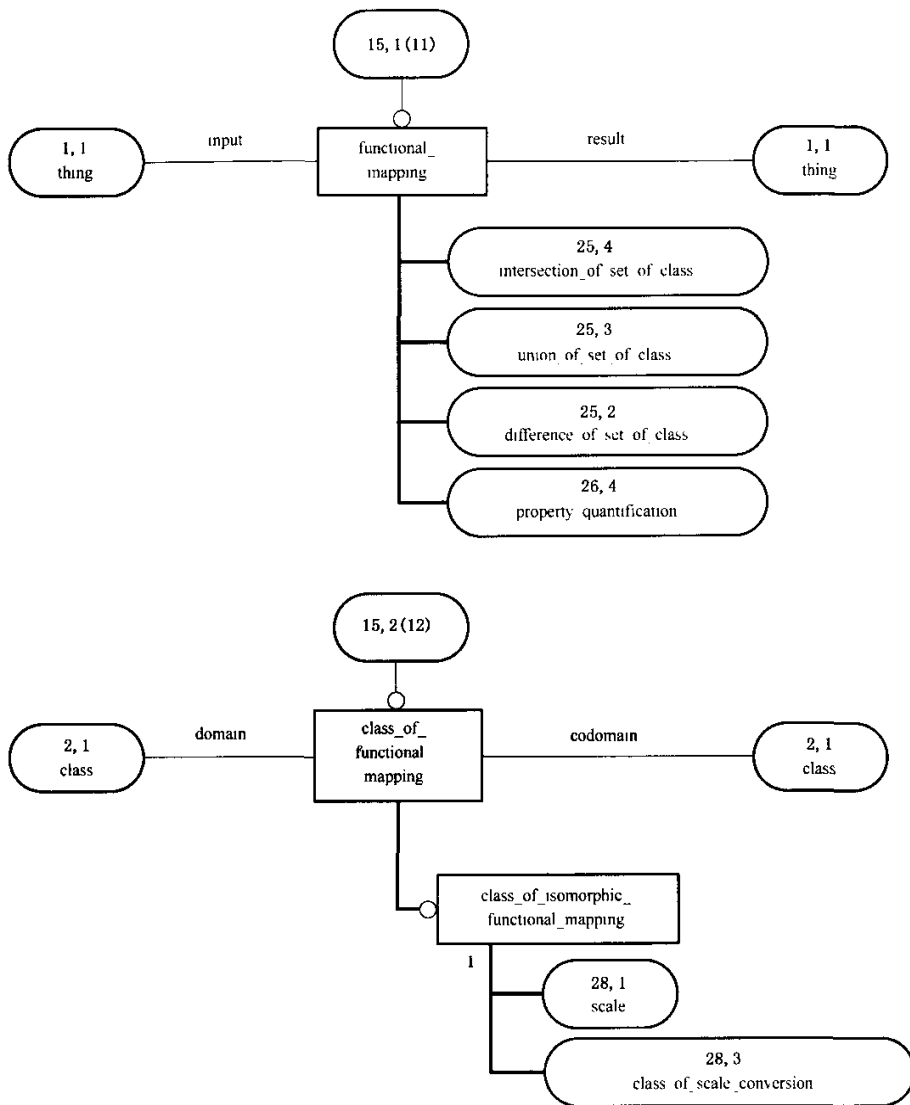


图 191 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 15 个)

5.2.15.1 函数映射类

class_of_functional_mapping(函数映射类)是多对一映射的 class_of_relationship。class_of_functional_mapping 是一个函数。

注 1: 这种实体类型自然具有函数名,但是这是 EXPRESS 的保留单词。

注 2: 多对一映射函数的意义是通常获得相同的答案。例如:五减三通常是二。具有其他两个参数的减函数可以得到二。

注 3: 如果函数具有多个参数,那么用 multidimensional_object 表达这些参数。

示例: 减函数是 class_of_functional_mapping 的一个例子。

EXPRESS 描述:

*)

```
ENTITY class_of_functional_mapping
    SUBTYPE OF(class_of_relationship);
    codomain ; class;
    domain ; class;
END_ENTITY;
```

(*

属性定义:

codomain: 把函数用于域的结果。

domain: 函数应用的事物集。

5.2.15.2 同构函数映射类

class_of_isomorphic_functional_mapping(同构函数映射类)是同构的 class_of_functional_mapping。

示例: 自然对数函数是 class_of_isomorphic_functional_mapping。

EXPRESS 描述:

(*

```
ENTITY class_of_isomorphic_functional_mapping
    SUPERTYPE OF (ONEOF(scale, class_of_scale_conversion))
    SUBTYPE OF(class_of_functional_mapping);
END_ENTITY;
```

(*

5.2.15.3 函数映射

functional_mapping(函数映射)是一种 relationship。它显示输入按 class_of_functional_mapping 分类的计算给出结果。

示例: [5, 3]映射到 2(通过减函数分类)是 functional_mapping 的一个例子。

EXPRESS 描述:

(*

```
ENTITY functional_mapping
    SUBTYPE OF(relationship);
    input ; thing;
    result ; thing;
END_ENTITY;
```

(*

属性定义:

input: 映射的输入。

result: 给定输入后函数的结果。

5.2.16 事物表达

本条声明了表达事物表达的实体数据类型。

注: 图 192 是本条定义的实体数据类型图(见 4.8.4.2)。

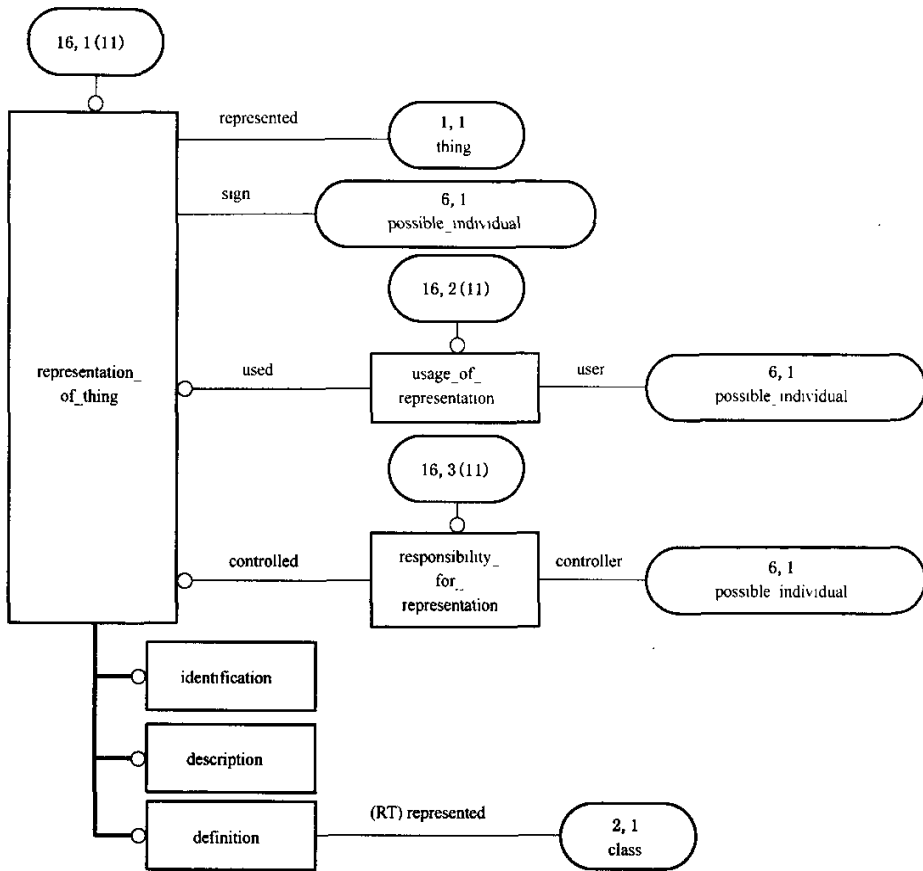


图 192 生命周期集成模式的 EXPRESS-G 图 (29 个图中的第 16 个)

5.2.16.1 定义

definition(定义)是一种 representation_of_thing,它显示标记 possible_individual 定义的 class。

示例: 上述语句的这个副本和其之前的标题之间的关系是一个 definition。

EXPRESS 描述:

*)

ENTITY definition

SUBTYPE OF(representation_of_thing);

SELF\representation_of_thing, represented ; class;

END_ENTITY;

(*

属性定义:

represented; definition 中定义的 class。

5.2.16.2 描述

description(描述)是一种 representation_of_thing,它显示 possible_individual 描述的 thing。

示例: 某精炼厂 1 号原油蒸馏设备的配管仪表与工厂具有 description 关系。

EXPRESS 描述:

*)

ENTITY description

```
    SUBTYPE OF(representation_of_thing);  
    END_ENTITY;  
    ( *
```

5.2.16.3 标识

identification(标识)是一种 representation_of_thing,它显示 possible_individual 是被标识 thing 的标识符。

示例1: 泵数据表的打印副本上的文本“P101”与可用 functional_physical_object 之间的关系是 identification 的一个例子。

示例2: 名字标签和佩戴它的员工之间的关系是 identification 的一个例子。

EXPRESS 描述:

```
    * )  
    ENTITY identification  
        SUBTYPE OF(representation_of_thing);  
    END_ENTITY;  
    ( *
```

5.2.16.4 事物表达

representation_of_thing(事物表达)是一种 relationship,它显示 possible_individual 是 thing 的标记。

示例: 带系列号和其他数据的标识牌与特殊压力容器(materialized_physical_object)之间的关系是 representation_of_thing(是一个 identification)的一个例子。

注: 通常,它将是关心的 class_of_representation_of_thing,而不是每个 representation_of_thing。然而,管理和控制单个的文档副本时,需要关心 representation_of_thing。

EXPRESS 描述:

```
    * ) .  
    ENTITY representation_of_thing  
        SUBTYPE OF(relationship);  
        represented : thing;  
        sign : possible_individual;  
    END_ENTITY;  
    ( *  
    属性定义:  
    represented: representation_of_thing 中表达的 thing。  
    sign: 在 representation_of_thing 中是标记的 possible_individual。
```

5.2.16.5 职责表达

responsibility_for_representation(职责表达)是一种 relationship,它显示控制者 possible_individual 管理被控制的 representation_of_thing。

示例: 本部分的管理职责归 ISO。

EXPRESS 描述:

```
    * )  
    ENTITY responsibility_for_representation  
        SUBTYPE OF(relationship);  
        controlled : representation_of_thing;  
        controller : possible_individual;
```

END_ENTITY;

(*

属性定义:

controlled: responsibility_for_representation 中被控制的 representation_of_thing。

controller: 在 responsibility_for_representation 中是控制者的 possible_individual。

5.2.16.6 表达应用

usage_of_representation(表达应用)是一种 relationship,它显示通过 possible_individual 使用 representation_of_thing。使用并不意味着职责。

示例: XYZ 公司在设计中使用标记“P101”表达特殊泵。

EXPRESS 描述:

*)

ENTITY usage_of_representation

SUBTYPE OF(relationship);

used : representation_of_thing;

user : possible_individual;

END_ENTITY;

(*

属性定义:

used: 某些用户或用户群使用的 representation_of_thing。

user: 是用户或用户群(使用 representation_of_thing)的 possible_individual。

5.2.17 表达类型

本条含实体数据类型的说明,并对表达类进行表达。

注: 图 193 是本条定义的实体数据类型图(见 4.8.4.2)。

5.2.17.1 定义类

class_of_definition(定义类)是一种 class_of_representation_of_thing,它显示的模式是已表达 class 的定义。

示例: 可以用 class_of_definition 的实例表达模式“移动液体的某物”与用英文名“泵”标识的 class 之间的连接。

EXPRESS 描述:

*)

ENTITY class_of_definition

SUBTYPE OF(class_of_representation_of_thing);

SELF\class_of_representation_of_thing, represented : class;

END_ENTITY;

(*

属性定义:

represented: 被引用 class_of_information_representation 成员定义的 class。

5.2.17.2 描述类

class_of_description(描述类)是一种 class_of_representation_of_thing,它显示的模式是被表达事物的描述。

示例: 可以用 class_of_description 的实例表达模式“这是老式喷射泵”与特殊泵之间的连接。

EXPRESS 描述:

*)

ENTITY class_of_description


```

SUBTYPE OF(class_of_representation_of_thing);
END_ENTITY;
( *

```

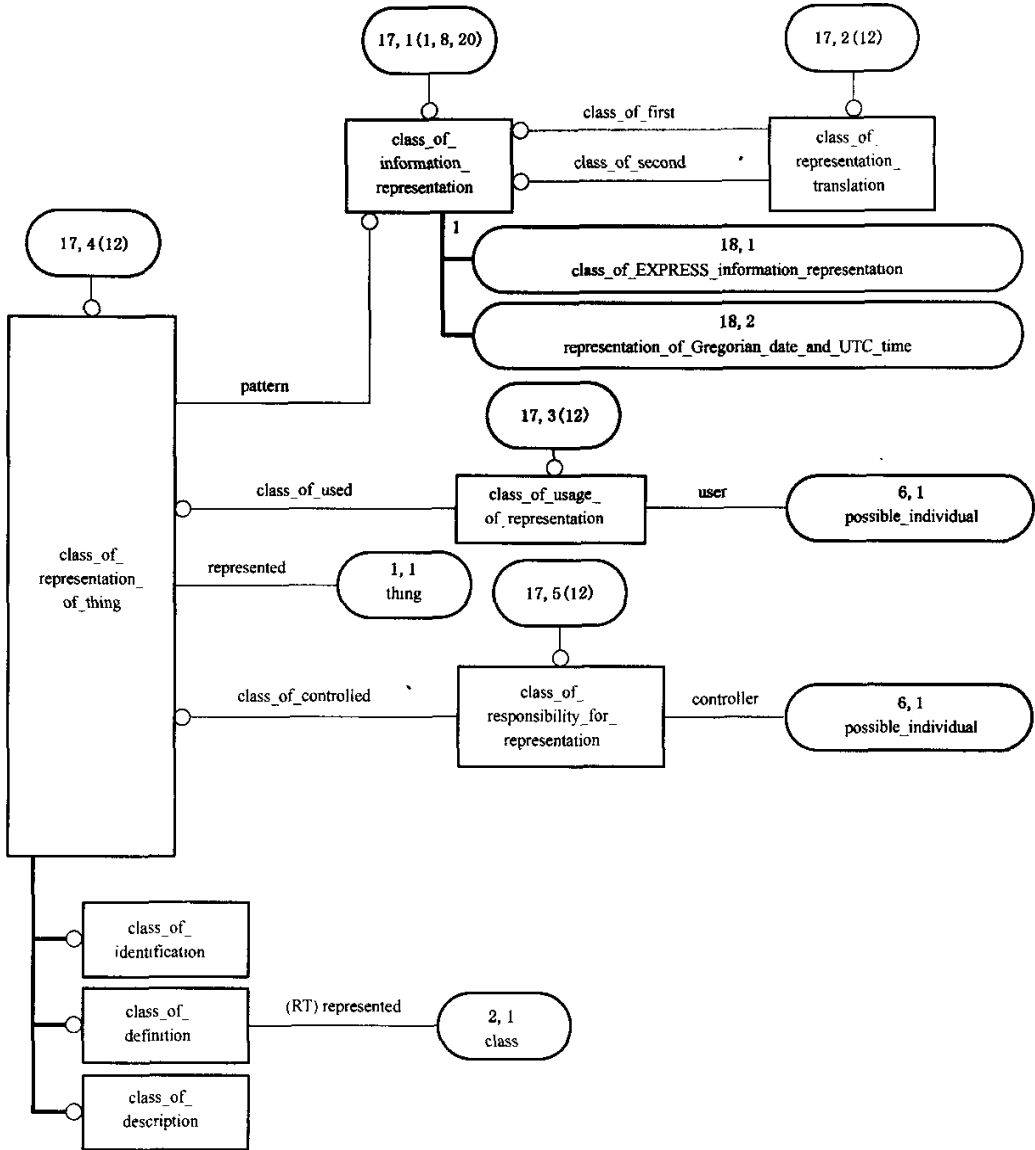


图 193 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 17 个)

5.2.17.3 标识类

`class_of_identification`(标识类)是一种 `class_of_representation_of_thing`,它显示用于引用被表达事物的模式。

示例: 可以用 `class_of_identification` 的实例表达模型“AC-1234”与特殊泵之间的连接(显示“AC-1234”的成员用于引用泵)。

EXPRESS 描述:

*)

```

ENTITY class_of_identification
    SUBTYPE OF(class_of_representation_of_thing);
END_ENTITY;
( *

```

5.2.17.4 信息表达类

class_of_information_representation(信息表达类)是定义表达信息模式的 class_of_arranged_individual。

示例：字符“s”连接字符“u”再连接字符“n”模式形成的文本是“sun” class_of_information_representation 的成员。

EXPRESS 描述：

```

* )
ENTITY class_of_information_representation
    SUPERTYPE OF (ONEOF(class_of_EXPRESS_information_representation,
                        representation_of_Gregorian_date_and_UTC_time))
    SUBTYPE OF(class_of_arranged_individual);
END_ENTITY;
( *

```

5.2.17.5 事物表达类

class_of_representation_of_thing(事物表达类)是一种 class_of_relationship,它显示表达 thing 的所有模型 class_of_information_representation 成员。

示例：可以用 class_of_information_representation 的实例表达,它指出用“伦敦”表示发生模型表达了英国首都的概念。

EXPRESS 描述：

```

* )
ENTITY class_of_representation_of_thing
    SUBTYPE OF(class_of_relationship);
    pattern : class_of_information_representation;
    represented : thing;
END_ENTITY;
( *

```

属性定义：

pattern: 其成员表达被引用 thing 的 class_of_information_representation。

represented: 用被引用 class_of_information_representation 成员表达的 thing。

5.2.17.6 翻译表达类

class_of_representation_translation(翻译表达类)是显示了 class_of_information_representation 两个实例翻译的 class_of_relationship。

示例：可以用 class_of_representation_translation 的实例表达显示了表达“F”和“15”是等效(分别是十六进制和八进制中十五的概念)的连接。

EXPRESS 描述：

```

* )
ENTITY class_of_representation_translation
    SUBTYPE OF(class_of_relationship);
    class_of_first : class_of_information_representation;
    class_of_second : class_of_information_representation;

```

END_ENTITY;

(*

属性定义:

class_of_first: 翻译中 class_of_information_representation 的第一个实例。

class_of_second: 翻译中 class_of_information_representation 的第二个实例。

5.2.17.7 职责表达类

class_of_responsibility_for_representation(职责表达类)是一种 class_of_relationship,其成员显示 possible_individual(通常是组织)认为模式成员可以用作被表达事物的表达。

示例: 可以用 class_of_responsibility_for_representation 的实例表达泵 # 1234 的标识与 XYZ 公司之间的连接(显示 XYZ 公司控制这个标识)。

EXPRESS 描述:

*)

ENTITY class_of_responsibility_for_representation

SUBTYPE OF(class_of_relationship);

class_of_controlled ; class_of_representation_of_thing;

controller ; possible_individual;

END_ENTITY;

(*

属性定义:

class_of_controlled: 被引用 possible_individual 控制的 class_of_representation_of_thing。

controller:; 控制被引用 class_of_representation_of_thing 的 possible_individual。

5.2.17.8 表达应用类

class_of_usage_of_representation(表达应用类)是一种 class_of_relationship,其成员显示 possible_individual(通常是组织)把被表达事物的表达作为模式成员读或使用。

示例: 可以用 class_of_usage_of_representation 的类表达泵 # 1234 的标识与承包人 ABC 有限公司之间的连接(显示 ABC 有限公司使用这个标识符)。

EXPRESS 描述:

*)

ENTITY class_of_usage_of_representation

SUBTYPE OF(class_of_relationship);

class_of_used ; class_of_representation_of_thing;

user ; possible_individual;

END_ENTITY;

(*

属性定义:

class_of_used: 被引用 possible_individual 使用的 class_of_representation_of_thing。

user: 使用被引用 class_of_representation_of_thing 的 possible_individual。

5.2.18 EXPRESS 和 UTC 的表达

本条声明了表达 EXPRESS 和 UTC 表达的实体数据类型。

注: 图 194 是本条定义的实体数据类型图。

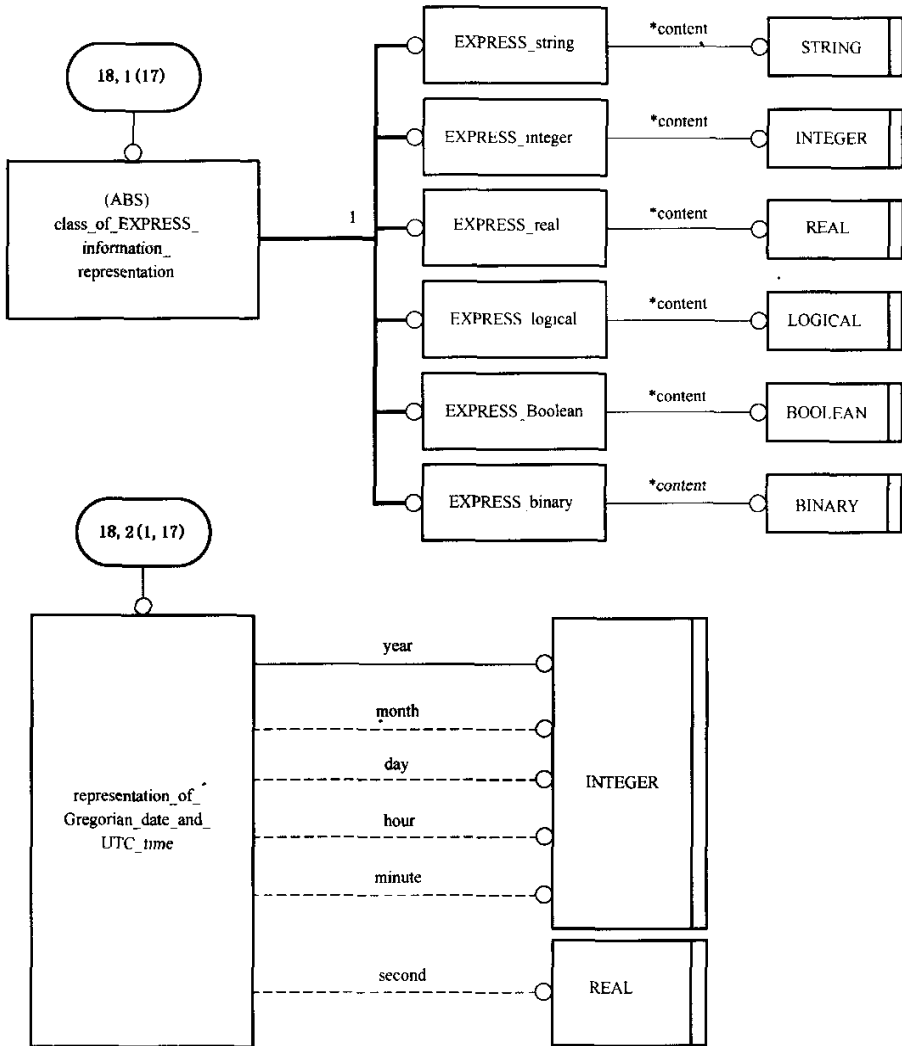


图 194 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 18 个)

5.2.18.1 EXPRESS 布尔值

EXPRESS_Boolean(EXPRESS 布尔值)是表达了 GB/T 16656.11—1996 中 8.1.5 定义的布尔值的 class_of_EXPRESS_information_representation。

注：唯一性规则保证任何值只拥有一次。

EXPRESS 描述：

(*)

ENTITY EXPRESS_Boolean

SUBTYPE OF(class_of_EXPRESS_information_representation);

content ; BOOLEAN;

UNIQUE

rule_1 ; content;

END_ENTITY;

(*)

属性定义:

content: EXPRESS_Boolean 的值。

正式声明:

rule_1: 系统中的内容应该是唯一的。

5.2.18.2 EXPRESS 二进制值

EXPRESS_binary(EXPRESS 二进制值)是表达了 GB/T 16656.11—1996 中 8.1.7 定义的二进制值的 class_of_EXPRESS_information_representation。

注: 唯一性规则保证任何值只拥有一次。

EXPRESS 描述:

*)

ENTITY EXPRESS_binary

SUBTYPE OF(class_of_EXPRESS_information_representation);

content ; BINARY;

UNIQUE

rule_1 ; content;

END_ENTITY;

(*

属性定义:

content: EXPRESS_binary 的值。

正式声明:

rule_1: 系统中的内容应该是唯一的。

5.2.18.3 EXPRESS 整数

EXPRESS_integer(EXPRESS 整数)是表达了 GB/T 16656.11—1996 中 8.1.3 定义的整数值的 class_of_EXPRESS_information_representation。

注: 唯一性规则保证任何值只拥有一次。

EXPRESS 描述:

*)

ENTITY EXPRESS_integer

SUBTYPE OF(class_of_EXPRESS_information_representation);

content ; INTEGER;

UNIQUE

rule_1 ; content;

END_ENTITY;

(*

属性定义:

content: EXPRESS_integer 的值。

正式声明:

rule_1: 系统中的内容应该是唯一的。

5.2.18.4 EXPRESS 逻辑值

EXPRESS_logical(EXPRESS 逻辑值)是表达了 GB/T 16656.11—1996 中 8.1.4 定义的逻辑值的 class_of_EXPRESS_information_representation。

注: 唯一性规则保证任何值只拥有一次。

EXPRESS 描述:

*)

```
ENTITY EXPRESS_logical
  SUBTYPE OF(class_of_EXPRESS_information_representation);
  content : LOGICAL;
UNIQUE
  rule_1 : content;
END_ENTITY;
```

(*

属性定义:

content: EXPRESS_logical 的值。

正式声明:

rule_1: 系统中的内容应该是唯一的。

5.2.18.5 EXPRESS 实数

EXPRESS_real(EXPRESS 实数)是表达了 GB/T 16656.11—1996 中 8.1.2 定义的实数值的 class_of_EXPRESS_information_representation。

注: 唯一性规则保证任何值只拥有一次。

EXPRESS 描述:

*)

```
ENTITY EXPRESS_real
  SUBTYPE OF(class_of_EXPRESS_information_representation);
  content : REAL;
UNIQUE
  rule_1 : content;
END_ENTITY;
```

(*

属性定义:

content: EXPRESS_real 的值。

正式声明:

rule_1: 系统中的内容应该是唯一的。

5.2.18.6 EXPRESS 字符串

EXPRESS_string(EXPRESS 字符串)是表达了 GB/T 16656.11—1996 中 8.1.6 定义的字符串的 class_of_EXPRESS_information_representation。

注: 唯一性规则保证任何值只拥有一次。

EXPRESS 描述:

*)

```
ENTITY EXPRESS_string
  SUBTYPE OF(class_of_EXPRESS_information_representation);
  content : STRING;
UNIQUE
  rule_1 : content;
END_ENTITY;
```

(*

属性定义:

content;EXPRESS_string 的值。

正式声明:

rule_1:系统中的内容应该是唯一的。

5.2.18.7 EXPRESS 信息表达类

class_of_EXPRESS_information_representation(EXPRESS 信息表达类)是 GB/T 16656.11 定义的 class_of_information_representation。

EXPRESS 描述:

*)

```
ENTITY class_of_EXPRESS_information_representation
    ABSTRACT SUPERTYPE OF (ONEOF(EXPRESS_string, EXPRESS_integer,
        EXPRESS_real, EXPRESS_logical, EXPRESS_Boolean,
        EXPRESS_binary))
    SUBTYPE OF(class_of_information_representation);
END_ENTITY;
(*
```

5.2.18.8 乔治日期和 UTC 时间的表达

representation_of_Gregorian_date_and_UTC_time(乔治日期和 UTC 时间的表达)是一种 class_of_information_representation,其成员是使用 GB/T 7408—2005 和格林日期表达系统规定的 UTC 时间标识系统的时间表达式。

应该是 UTC 时间表达式表达所有时间。日期应该遵守格林日语法。

注 1: 世界标准时间(UTC)是世界法定时间的基础,并严格遵守 TAI(见下文),秒的整数除外。为了确保平均年,根据国际地球自转服务(IERS)(<http://hpiers.obspm.fr>)的建议插入这些闰秒,太阳在格林子午线 12:00:00 UTC 的 0.9 s 内是在正顶。因此,UTC 是格林尼治标准时间(GMT)的继承者,当时间单位是平均太阳日时使用它。用来自 200 多个原子钟(位于全世界 30 多个国家的度量衡研究所和天文台)的 BIPM 计算国际原子时间(TAI)。每个月在 BIPM 圆形 T(<ftp://62.161.69.5/pub/tai/publication>)中计算有效的 TAI。相对于想象中的时钟来说,估计 TAI 每年不会失去或获得超过十分之一微秒(0.000 000 1 s)。

注 2: 虽然 GB/T 7408 允许午夜时间有 0 和 24 两种表达,但是本部分限制表示为 0。

注 3: 秒值上升,但不包括 61.0,其允许闰秒。这意味着地球自转决定了太阳时间。通常在年中或年末根据需要增加或减少闰秒,尽管地球的自转发生变化,通过多一秒的方法保证法定时间与非正式的太阳时间一致。

EXPRESS 描述:

*)

```
ENTITY representation_of_Gregorian_date_and_UTC_time
    SUBTYPE OF(class_of_information_representation);
    year : INTEGER;
    month : OPTIONAL INTEGER;
    day : OPTIONAL INTEGER;
    hour : OPTIONAL INTEGER;
    minute : OPTIONAL INTEGER;
    second : OPTIONAL REAL;
```

WHERE

```
valid_month : {1 <= month <= 12};
valid_day : {1 <= day <= 31};
valid_hour : {0 <= hour <= 23};
```

```

    valid_minute : {0 <= minute <= 59};
    valid_second : {0.0 <= second < 61.0};
END_ENTITY;

```

(*

属性定义:

year: 阳历中定义的年, 应该使用与明确传播世纪和世纪中的年一样的许多必要数字定义年。不应该使用截短的年数。

month: 指定月的位置, 如 GB/T 7408—2005 中 5.2.1 定义的一样。

day: 天的值, 如 GB/T 7408—2005 中 5.2.1 定义的一样。

hour: 24 小时钟上指定时间的小时元素。应该用零值表达午夜。

minute: 指定时间的分元素。

second: 指定时间的秒元素。

正式声明:

valid_month: 月应该是 1~12 之间的值。

valid_day: 天应该是 1~31 之间的值。

valid_hour: 小时应该是 0~23 之间的值。

valid_minute: 分应该是 0~59 之间的值。

valid_second: 秒应该是 0.0~61.0 的值, 但不包括 61.0。

5.2.19 表达类的所有类型

本条声明了表达表达类类型的实体数据类型。

注: 图 195 是本条定义的实体数据类型图。

5.2.19.1 定义类的类

class_of_class_of_definition(定义类的类)是一种 **class_of_class_of_representation**, 其成员是 **class_of_definition** 的成员。

示例: “标准”是一个 **class_of_class_of_definition**。

EXPRESS 描述:

(*

```

ENTITY class_of_class_of_definition
    SUBTYPE OF(class_of_class_of_representation);
END_ENTITY;

```

(*

5.2.19.2 描述类的类

class_of_class_of_description(描述类的类)是一种 **class_of_class_of_representation**, 其成员是 **class_of_description** 的成员。

示例: 服务描述是一个 **class_of_class_of_description**。

EXPRESS 描述:

(*

```

ENTITY class_of_class_of_description
    SUBTYPE OF(class_of_class_of_representation);
END_ENTITY;

```

(*

5.2.19.3 标识类的类

class_of_class_of_identification(定义类的类)是一种 **class_of_class_of_representation**, 其成员是

class_of_identification 的成员。

示例：可以用 class_of_class_of_identification 的实例表达类“制造零件族”与 representation_form “GB/T 17645 基本语义单元”之间的连接(显示可以使用 GB/T 17645 标识零件族)。

EXPRESS 描述：

*)

ENTITY class_of_class_of_identification

 SUBTYPE OF(class_of_class_of_representation);

END_ENTITY;

(*

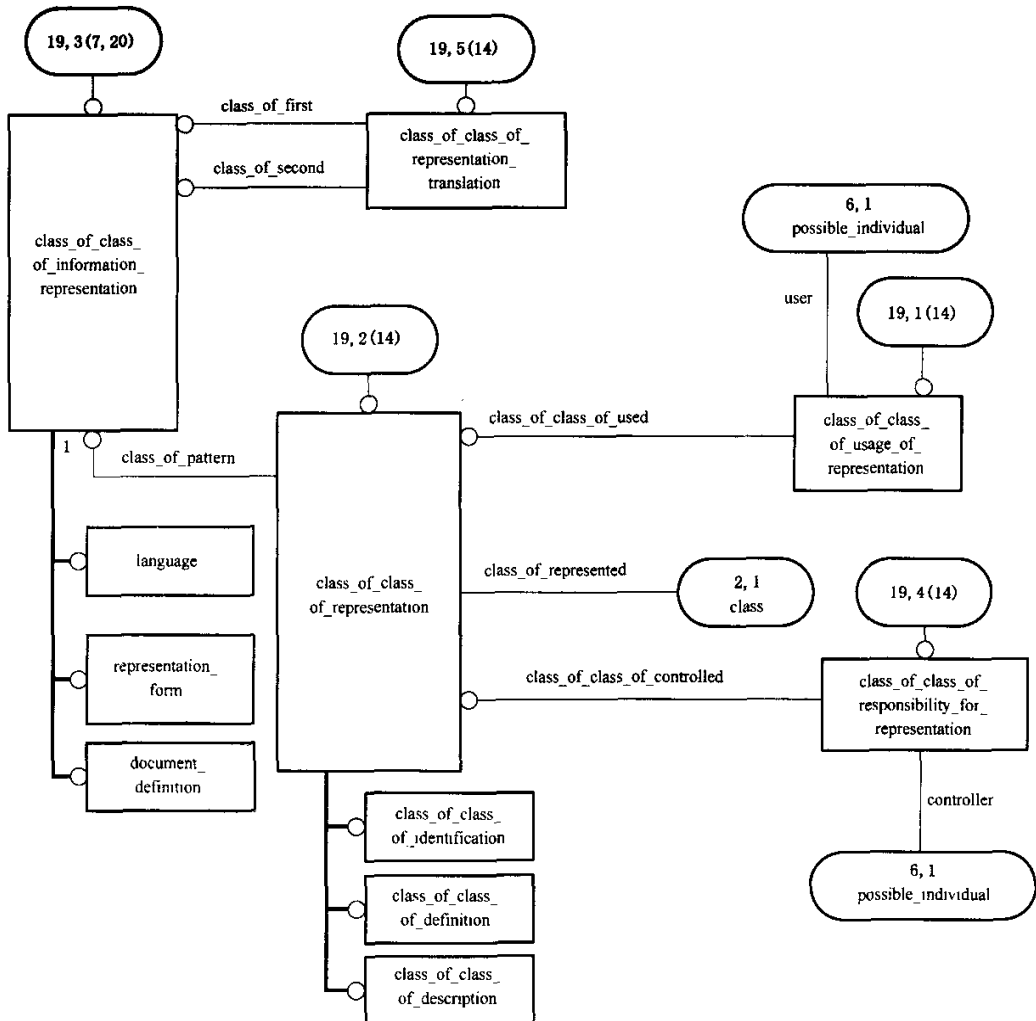


图 195 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 19 个)

5.2.19.4 信息表达类的类

class_of_class_of_information_representation(信息表达类的类)是一种对信息表达类进行分类的 class_of_class_of_individual。

示例：八进制整数是一个 class_of_class_of_representation,其成员是对应八进制格式整数的所有信息表达类。

EXPRESS 描述：

*)

```
ENTITY class_of_class_of_information_representation
    SUPERTYPE OF (ONEOF(representation_form, language,
                        document_definition))
    SUBTYPE OF(class_of_class_of_individual);
END_ENTITY;
```

(*)

5.2.19.5 表达类的类

class_of_class_of_representation(表达类的类)是一种 class_of_class_of_relationship,其成员是 class_of_representation_of_thing 的实例。

示例:显示可以用类“XML”模式表达类“文档”成员的连接是一个 class_of_class_of_representation。

EXPRESS 模式:

*)

```
ENTITY class_of_class_of_representation
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_pattern : class_of_class_of_information_representation;
    class_of_represented : class;
END_ENTITY;
```

(*)

属性定义:

class_of_pattern:其成员可以表达被引用类成员的 class_of_class_of_information_representation。

class_of_represented:可以用被引用 class_of_class_of_information_representation 成员表达其成员的 class。

5.2.19.6 表达翻译类的类

class_of_class_of_representation_translation 是一种 class_of_class_of_relationship,其成员是 class_of_representation_translation 的成员。

示例:其成员包括 ASCII 二进制成员与 ASCII 文本表达类之间所有翻译类的类 ASCII 是一个 class_of_class_of_representation_translation。

EXPRESS 描述:

*)

```
ENTITY class_of_class_of_representation_translation
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_first : class_of_class_of_information_representation;
    class_of_second : class_of_class_of_information_representation;
END_ENTITY;
```

(*)

属性定义:

class_of_first:为其定义翻译的第一个 class_of_class_of_information_representation。

class_of_second:为其定义翻译的第二个 class_of_class_of_information_representation。

5.2.19.7 职责表达类的类

class_of_class_of_responsibility_for_representation(职责表达类的类)是一种 class_of_class_of_relationship,其成员是连接控制者与表达集的 class_of_responsibility_for_representation 的成员。

示例:可以用 class_of_class_of_usage_of_representation 的实例表达韦氏与标识集(韦氏泵与韦氏系列号之间显示

韦氏定义的标识符的)之间的连接。

EXPRESS 描述:

*)

```
ENTITY class_of_class_of_responsibility_for_representation
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_class_of_controlled ; class_of_class_of_representation;
    controller ; possible_individual;
END_ENTITY;
```

(*

属性定义:

class_of_class_of_controlled:被引用 possible_individual 控制的 class_of_class_of_representation。
controller:控制被引用 class_of_class_of_representation 的 possible_individual。

5.2.19.8 表达应用类的类

class_of_class_of_usage_of_representation(表达应用类的类)是一种 class_of_class_of_relationship,其成员是连接用户和表达集的 class_of_usage_of_representation 的成员。

示例:可以用 class_of_class_of_usage_of_representation 的实例表达用户公司与标识集(韦氏泵与韦氏系列号之间显示用户使用韦氏标识符的)之间的连接。

EXPRESS 描述:

*)

```
ENTITY class_of_class_of_usage_of_representation
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_class_of_used ; class_of_class_of_representation;
    user ; possible_individual;
END_ENTITY;
```

(*

属性定义:

class_of_class_of_used:被引用 possible_individual 使用的 class_of_class_of_representation。
user:使用被引用 class_of_class_of_representation 的 possible_individual。

5.2.19.9 文档定义

document_definition(文档定义)是定义文档内容和/或结构的 class_of_class_of_information_representation。

示例:XYZ公司的材料安全数据表是一个 document_definition。

EXPRESS 描述:

*)

```
ENTITY document_definition
    SUBTYPE OF(class_of_class_of_information_representation);
END_ENTITY;
```

(*

5.2.19.10 语言

Language(语言)是一种 class_of_class_of_information_representation,其成员是用语言创造的所有信息表达。

示例:可以用 language 的实例表达英语、法语、C++ 和 Java。

EXPRESS 描述:

*)

ENTITY language

SUBTYPE OF(class_of_class_of_information_representation);

END_ENTITY;

(*)

5.2.19.11 表达格式

representation_form(表达格式)是区分表达格式的 class_of_class_of_information_representation。

示例：都可以用 representation_form 的实例表达十六进制、文本、脚本、符号、图片、图表、旗语、莫尔斯电码、乐谱、乐器数字界面、文件格式和 XML。

EXPRESS 描述：

*)

ENTITY representation_form

SUBTYPE OF(class_of_class_of_information_representation);

END_ENTITY;

(*)

5.2.20 命名空间

本条声明了表达命名空间的实体数据类型。

注：图 196 是本条定义的实体数据类型图(见 4.8.4.2.5)。

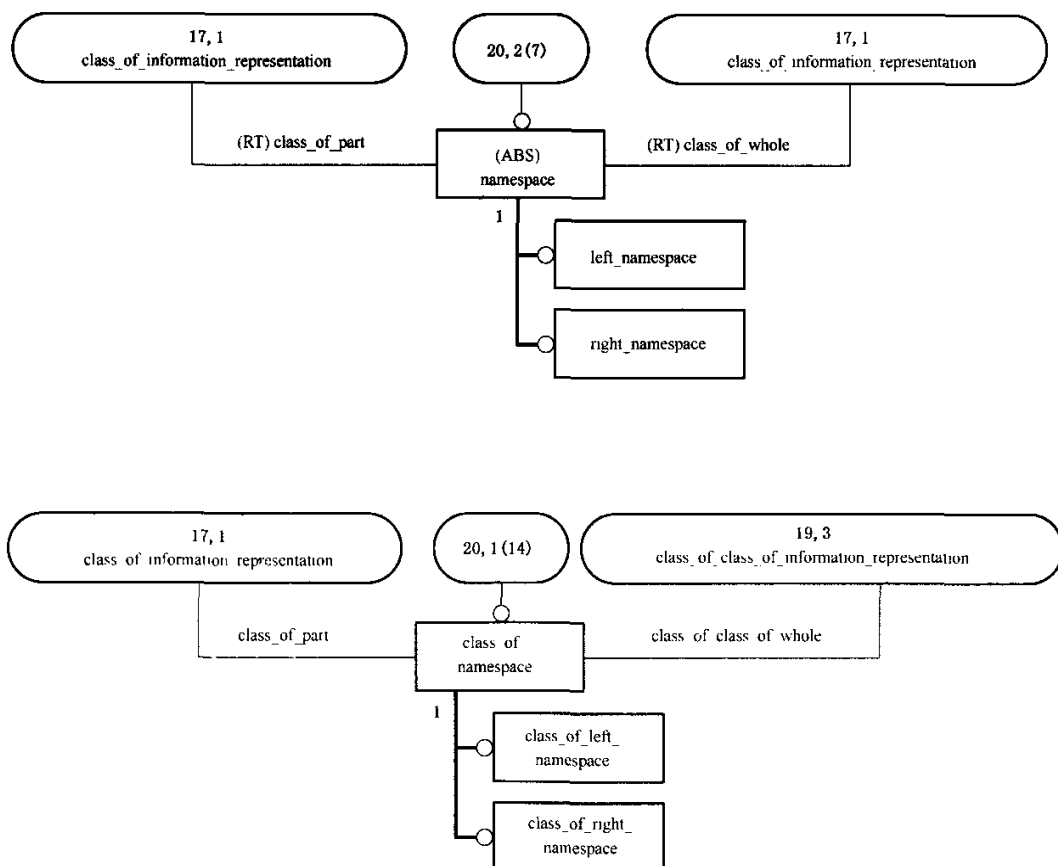


图 196 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 20 个)

5.2.20.1 左命名空间的类

class_of_left_namespace(左命名空间的类)是一种 class_of_namespace,它显示 class_of_part 是 class_of_class_of_whole 成员的 left_namespace。

示例: WC1:是水公司 1 的消费者站点标识符的 left_namespace。

EXPRESS 描述:

```
* )
ENTITY class_of_left_namespace
    SUBTYPE OF(class_of_namespace);
END_ENTITY;
( *
```

5.2.20.2 命名空间的类

class_of_namespace(命名空间的类)是一种 class_of_class_of_relationship,它显示 class_of_information_representation 是用作所有 class_of_class_of_information_representation(是 class_of_class_of_whole)成员的 class_of_part。

示例: WC1 用作水公司标识符集的命名空间。

EXPRESS 描述:

```
* )
ENTITY class_of_namespace
    SUPERTYPE OF (ONEOF(class_of_left_namespace,
                        class_of_right_namespace))
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_class_of_whole :
        class_of_class_of_information_representation;
    class_of_part : class_of_information_representation;
END_ENTITY;
( *
```

属性定义:

class_of_class_of_whole:其成员具有命名空间的 class_of_class_of_information_representation。

class_of_part:是命名空间的 class_of_information_representation。

5.2.20.3 右命名空间的类

class_of_right_namespace(右命名空间的类)是一种 class_of_namespace,其中 class_of_part 是 class_of_class_of_whole 成员的命名空间。

EXPRESS 描述:

```
* )
ENTITY class_of_right_namespace
    SUBTYPE OF(class_of_namespace);
END_ENTITY;
( *
```

5.2.20.4 左命名空间

left_namespace(左命名空间)是一种 namespace,其中 class_of_part 是 class_of_whole 的左部分。

示例: 在 WC1:1234 中,WC1:是命名空间,它是一个 left_namespace。

EXPRESS 描述:

```
* )
```

```

ENTITY left_namespace
  SUBTYPE OF(namespace);
END_ENTITY;
( *

```

5.2.20.5 命名空间

namespace(命名空间)是一种 class_of_arrangement_of_individual, 其中 class_of_information_representation 和部件的 class_of_whole 和 class_of_part 是整个命名空间中最有效的部分。

示例: 在标识符 WC1:1234 中, 字符串 WC1: 是一个 namespace。

EXPRESS 描述:

```

* )
ENTITY namespace
  ABSTRACT SUPERTYPE OF (ONEOF(right_namespace, left_namespace))
  SUBTYPE OF(class_of_arrangement_of_individual);
  SELF\class_of_composition_of_individual.class_of_part ;
                                     class_of_information_representation;
  SELF\class_of_arrangement_of_individual.class_of_whole ;
                                     class_of_information_representation;
END_ENTITY;
( *

```

属性定义:

class_of_part: 是命名空间的 class_of_information_representation。

class_of_whole: 具有命名空间 class_of_part 的 class_of_information_representation。

5.2.20.6 右命名空间

right_namespace(右命名空间)是一种 namespace, 它显示 class_of_part 是 class_of_whole 的最右部分。

示例: 当 ZH 是 5367ZH 中的命名空间时, 它们之间的关系用 right_namespace 显示。

EXPRESS 描述:

```

* )
ENTITY right_namespace
  SUBTYPE OF(namespace);
END_ENTITY;
( *

```

5.2.21 连接

本条声明了表达连接的实体数据类型。

注: 图 197 是本条定义的实体数据类型图(见 4.7.3)。

5.2.21.1 个体连接类

class_of_connection_of_individual(个体连接类)是一种 class_of_relationship, 其成员是 connection_of_individual 的成员。它显示 class_of_side_1 class_of_individual 的成员可以连接 class_of_side_2 class_of_individual 的成员。

注 1: 在该 class_of_connection_of_individual 成员的 connection_of_individual 中, class_of_side_1 和 class_of_side_2 显示分别是 side_1 和 side_2 的 class_of_individual。

注 2: 不能用 class_of_connection_of_individual 的实例表达柔度、刚度和焊缝, 它们是连接中被连接或使用的材料类。

示例: 电线之间的电连接是一个 class_of_connection_of_individual。

EXPRESS 描述:

*)

ENTITY class_of_connection_of_individual

ABSTRACT SUPERTYPE OF (ONEOF(class_of_direct_connection,
class_of_indirect_connection))

SUBTYPE OF(class_of_relationship);

class_of_side_1 : class_of_individual;

class_of_side_2 : class_of_individual;

END_ENTITY;

(*)

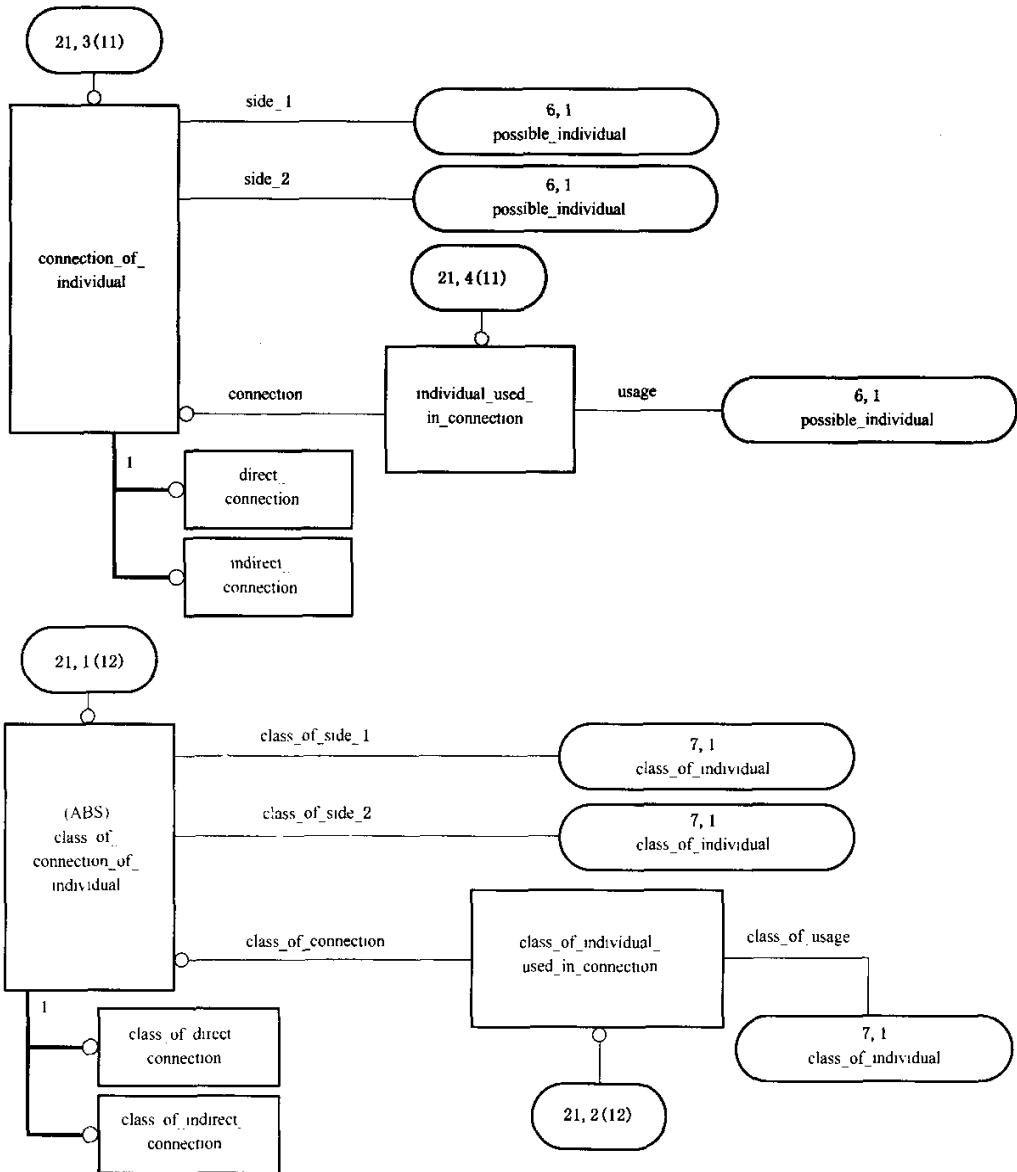


图 197 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 21 个)

属性定义:

class_of_side_1:其成员在 class_of_connection_of_individual 成员中扮演 side_1 角色的 class_of_individual。

class_of_side_2:其成员在 class_of_connection_of_individual 成员中扮演 side_2 角色的 class_of_individual。

5.2.21.2 直接连接类

class_of_direct_connection(直接连接类)是一种 class_of_connection_of_individual,其成员是 direct_connection 的成员。

示例:三腿电插头插入三空插座是 class_of_direct_connection 的一个例子。

EXPRESS 描述:

*)

ENTITY class_of_direct_connection

SUBTYPE OF(class_of_connection_of_individual);

END_ENTITY;

(*

5.2.21.3 间接连接类

class_of_indirect_connection(间接连接类)是一种 class_of_connection_of_individual,其成员是 indirect_connection 的成员。

示例:滴管间接连接排泄漏斗是 class_of_indirect_connection 的一个例子。

EXPRESS 描述:

*)

ENTITY class_of_indirect_connection

SUBTYPE OF(class_of_connection_of_individual);

END_ENTITY;

(*

5.2.21.4 连接中个体的类

class_of_individual_used_in_connection(连接中个体的类)是一种 class_of_relationship,其成员是 individual_used_in_connection 的成员。它显示在 class_of_connection_of_individual 中使用 class_of_individual 的成员。

示例:可以用 class_of_individual_involved_in_connection 的实例表达 class_of_connection_of_individual(它显示 B12 类型的线束连接到管支架)与 class_of_individual “直径 20 mm 的螺钉”(它显示管之间与 B12 类型线束之间的连接使用了四个直径 20 mm 的螺钉)之间的连接。

EXPRESS 描述:

*)

ENTITY class_of_individual_used_in_connection

SUBTYPE OF(class_of_relationship);

class_of_connection ; class_of_connection_of_individual;

class_of_usage ; class_of_individual;

END_ENTITY;

(*

属性定义:

class_of_connection:其成员在 class_of_individual_involved_in_connection 成员中是连接的 class_of_connection_of_individual。

class_of_usage: 在 class_of_individual_used_in_connection 成员中使用其成员的 class_of_individual。

5.2.21.5 个体连接

connection_of_individual(个体连接)是一种 relationship,它显示可以在直接或间接连接的 possible_individual 成员之间转移物质、能量或两者。对这两个相关的 possible_individual 实例进行排序没有重要意义。仅仅使用名字 side_1 和 side_2 区分属性。

EXPRESS 描述:

*)

```
ENTITY connection_of_individual
    SUPERTYPE OF (ONEOF(direct_connection, indirect_connection))
    SUBTYPE OF(relationship);
    side_1 : possible_individual;
    side_2 : possible_individual;
END_ENTITY;
```

(*

属性定义:

side_1: connection_of_individual 相关的第一个 possible_individual。

side_2: connection_of_individual 相关的第二个 possible_individual。

5.2.21.6 直接连接

direct_connection(直接连接)是一种 connection_of_individual,它显示 side_1 和 side_2 是通过通用空间边界直接连接的。

示例: 可以用 direct_connection 的实例表达一种关系,这种关系显示终结一系列通讯电缆的插头连接计算机设备上的插座。

EXPRESS 描述:

*)

```
ENTITY direct_connection
    SUBTYPE OF(connection_of_individual);
END_ENTITY;
```

(*

5.2.21.7 间接连接

indirect_connection(间接连接)是一种 connection_of_individual,它显示 side_1 和 side_2 通过其他个体连接。

示例: 可以用 indirect_connection 的实例表达显示伦敦和巴黎之间有一条铁路连接的关系。

EXPRESS 描述:

*)

```
ENTITY indirect_connection
    SUBTYPE OF(connection_of_individual);
END_ENTITY;
```

(*

5.2.21.8 连接中使用的个体

individual_used_in_connection(连接中使用的个体)是一种 relationship,它显示 connection_of_individual 中使用的 possible_individual。

示例: 可以用 individual_used_in_connection 的实例表达两根管线的法兰盘接头与螺钉、螺母、垫片和衬垫集临时部件(显示螺钉和衬垫集参与连接)之间的 relationship。

EXPRESS 描述:

*)

ENTITY individual_used_in_connection

SUBTYPE OF (relationship);

connection : connection_of_individual;

usage : possible_individual;

END_ENTITY;

(*

属性定义:

connection: 被引用 possible_individual participates 参与其中的 connection_of_individual。

usage: 参与被引用 connection_of_individual 的 possible_individual。

5.2.22 相对位置和顺序

本条声明了表达相对位置和顺序的实体数据类型。

注: 图 198 是本条定义实体数据类型图(见 4.7.4)。

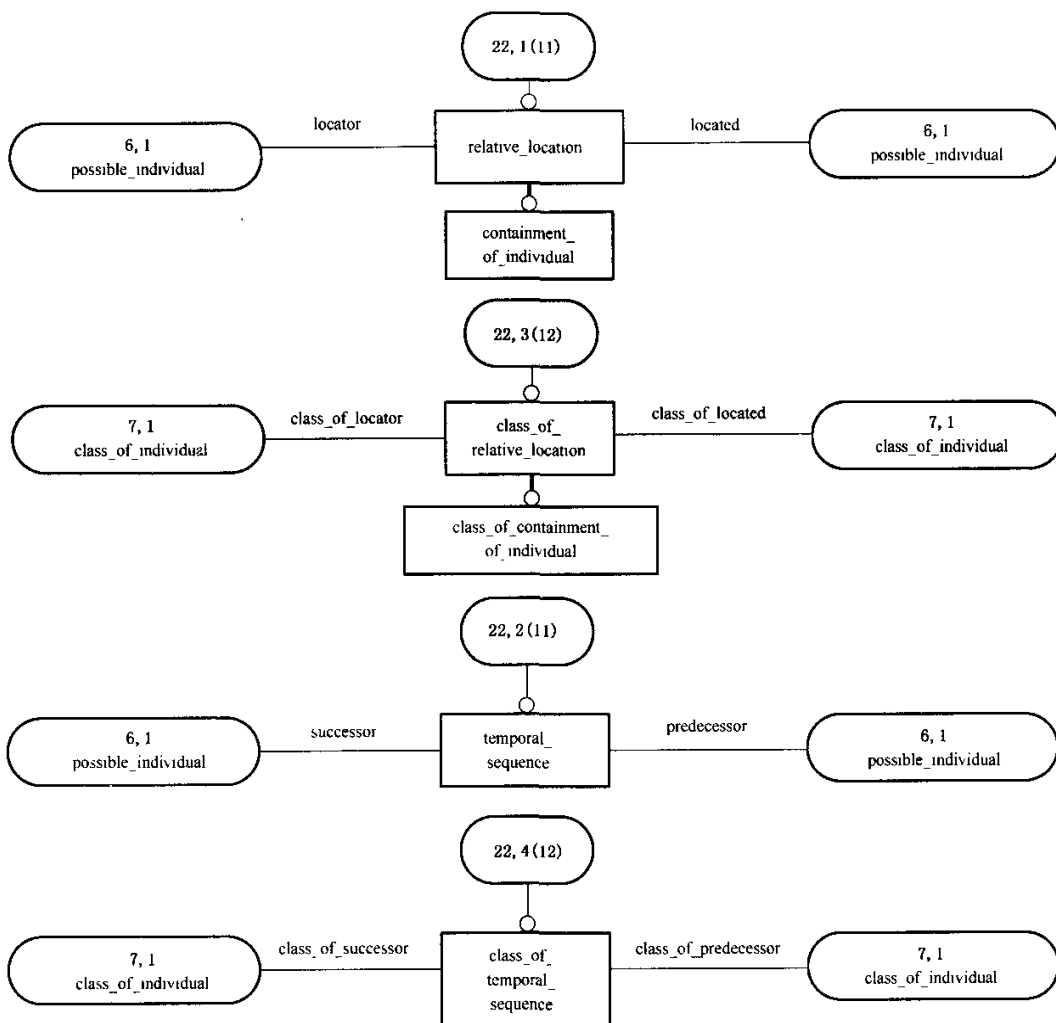


图 198 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 22 个)

5.2.22.1 个体容积类

class_of_containment_of_individual(个体容积类)是一 class_of_relative_location 种,其成员是 containment_of_individual 的实例。它显示 class_of_locator class_of_individual 的成员可以包含 class_of_located class_of_individual 的成员。

示例:可以用“1 500 mL 有螺旋盖的塑料瓶”装那种“防冻液体”是一个 class_of_containment_of_individual。

EXPRESS 描述:

```
* )
ENTITY class_of_containment_of_individual
    SUBTYPE OF(class_of_relative_location);
END_ENTITY;
( *
```

5.2.22.2 相对位置类

class_of_relative_location(相对位置类)是一种 class_of_relationship,其成员是 relative_location 的实例。

示例:旁边、上面和下面是 class_of_relative_location 的例子。

EXPRESS 描述:

```
* )
ENTITY class_of_relative_location
    SUBTYPE OF(class_of_relationship);
    class_of_located ; class_of_individual;
    class_of_locator ; class_of_individual;
END_ENTITY;
( *
```

属性定义:

class_of_located:通过 class_of_locator class_of_individual 成员定位其成员的 class_of_individual。

class_of_locator:其成员担当 class_of_located class_of_individual 成员定位器的 class_of_individual。

5.2.22.3 临时顺序类

class_of_temporal_sequence(临时顺序类)是一种 class_of_relationship,其中顺序是临时自然状态。

示例 1:可以用 class_of_sequence 的实例表达显示成员“七月”跟随成员“六月”的连接。

示例 2:可以用 class_of_sequence 的实例表达显示排空油罐活动在清洗活动之前的连接。

EXPRESS 描述:

```
* )
ENTITY class_of_temporal_sequence
    SUBTYPE OF(class_of_relationship);
    class_of_predecessor ; class_of_individual;
    class_of_successor ; class_of_individual;
END_ENTITY;
( *
```

属性定义:

class_of_predecessor:在 class_of_sequence 成员中其成员是前任的 class_of_individual。

class_of_successor:在 class_of_sequence 成员中其成员是后续者的 class_of_individual。

5.2.22.4 个体容积

containment_of_individual(个体容积)是一种 relationship,其中定位器 possible_individual 包含被定位的 possible_individual,但不是它的某部分。

示例: 可以用 containment_of_individual 的实例表达容器内部的容器容量。

注: 容积与成分不一样,在成分中,整体由其所有的部分组成,与容积一致,所包含的不是容器的一部分。

EXPRESS 描述:

*)

```
ENTITY containment_of_individual
    SUBTYPE OF(relative_location);
```

```
END_ENTITY;
```

(*

5.2.22.5 相对位置

relative_location(相当位置)是一种 relationship,它显示一个 possible_individual 的位置是相对另一个 possible_individual 的位置的。

注: relative_location 的 classification 显示 relative_location 的本质,例如上面、下面和旁边。

示例: 在用 class_of_relative_location 上面分类的 relative_location 中,A(被定位者)相对 B(定位器)显示 A 在 B 的上面。

EXPRESS 描述:

*)

```
ENTITY relative_location
    SUBTYPE OF(relationship);
    located : possible_individual;
    locator : possible_individual;
```

```
END_ENTITY;
```

(*

属性定义:

located: 被定位的 possible_individual。

locator: 是被定位 possible_individual 的参考位置的 possible_individual。

5.2.22.6 临时顺序

temporal_sequence(临时顺序)是一种 relationship,在临时检测中,它显示一个 possible_individual 在另一个 possible_individual 之前。

示例 1: 可以用 temporal_sequence 的实例表达工厂建设阶段 possible_individual 在工厂试运行阶段 possible_individual 之前的 relationship。

示例 2: 可以用 temporal_sequence 的实例表达工业革命 period_in_time known 在信息革命 period_in_time 之前的 relationship。

EXPRESS 描述:

*)

```
ENTITY temporal_sequence
    SUBTYPE OF(relationship);
    predecessor : possible_individual;
    successor : possible_individual;
```

```
END_ENTITY;
```

(*

属性定义：

predecessor: 在顺序中是前任的 possible_individual。

successor: 在顺序中是后续者的 possible_individual。

5.2.23 生命周期阶段和批准

本条声明了表达生命周期阶段和批准的实体数据类型。

注：图 199 是本条定义的实体数据类型图(见 4.7.7 和 4.7.18)。

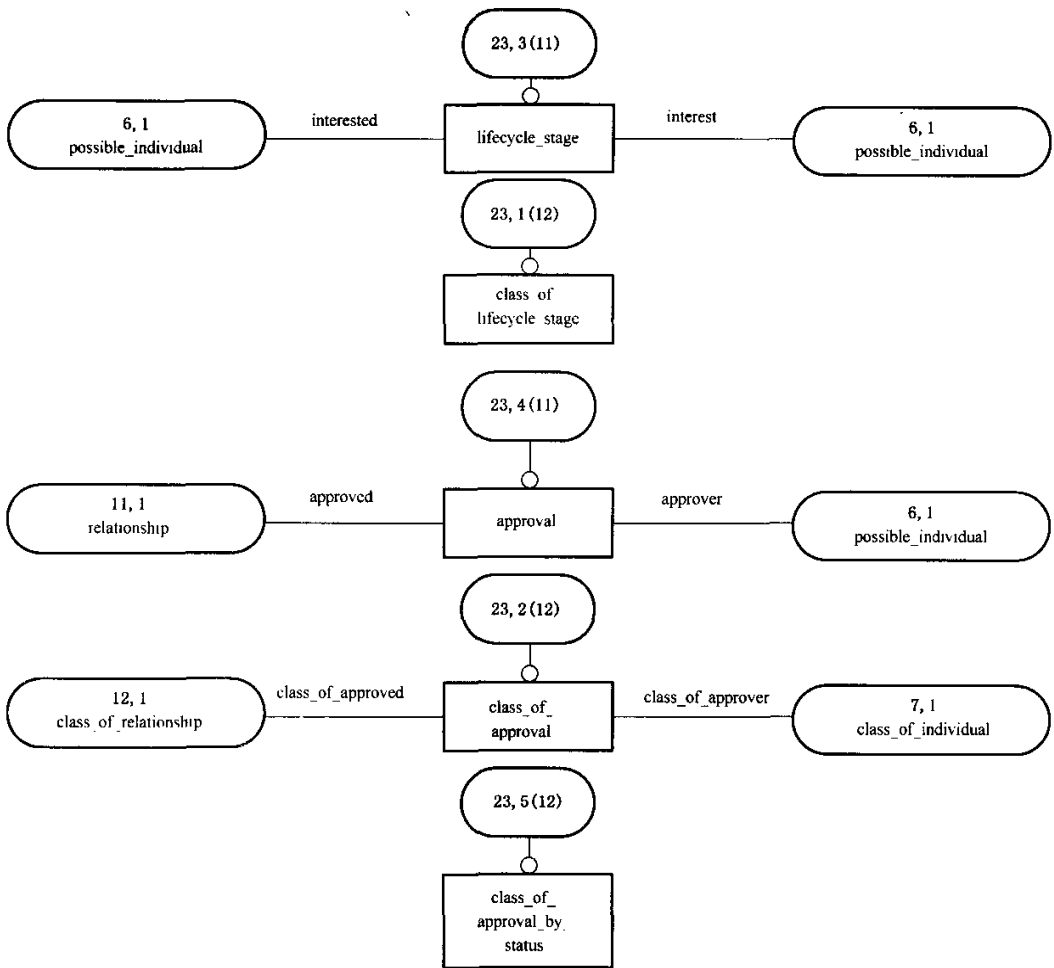


图 199 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 23 个)

5.2.23.1 批准

Approval(批准)是一种 relationship,它显示批准者 possible_individual 已经批准 relationship。

注：应该关心需要批准什么。有时候不是说一个泵已批准,而是特殊 activity 中泵的参与,或某些 class_of_activity 的成员。

示例：带建造活动的工厂设计 involvement_by_reference(现场经理批准的)是 approval 的一个例子。

EXPRESS 描述：

*)

ENTITY approval

SUBTYPE OF(representation);

```

    approved ; relationship;
    approver ; possible_individual;
END_ENTITY;

```

(*

属性定义:

approved: approval 中被批准的 relationship。

approver: approval 中是批准者的 possible_individual。

5.2.23.2 批准类

class_of_approval(批准类)是一种 class_of_relationship,其成员是 approval 的成员,它显示 class_of_individual 的成员在 approval 中对于被批准的类成员来说是批准者。

示例: 现场经理批准建筑(class_of_involvement_by_reference)的设计规范是 class_of_approval 的一个例子。

EXPRESS 描述:

(*

```

ENTITY class_of_approval
    SUBTYPE OF(class_of_relationship);
    class_of_approved ; class_of_relationship;
    class_of_approver ; class_of_individual;
END_ENTITY;

```

(*

属性定义:

class_of_approved: class_of_approver 成员批准其成员的 class_of_relationship。

class_of_approver: 其成员是 class_of_relationship 批准者的 class_of_individual。

5.2.23.3 状态批准类

class_of_approval_by_status(状态批准类)是一种 class_of_relationship,它显示批准状态独立于批准者批准的内容。

示例: 批准、批准但有意见、拒绝并有意见是 class_of_approval_by_status 的例子。

EXPRESS 描述:

(*

```

ENTITY class_of_approval_by_status
    SUBTYPE OF(class_of_relationship);
END_ENTITY;

```

(*

5.2.23.4 生命周期阶段类

class_of_lifecycle_stage(生命周期阶段类)是一种 class_of_relationship,其成员是 lifecycle_stage 的成员。

示例: 可以用 class_of_lifecycle_stage 的实例表达计划的、必须的、期望的和提议的。

EXPRESS 描述:

(*

```

ENTITY class_of_lifecycle_stage
    SUBTYPE OF(class_of_relationship);
END_ENTITY;

```

(*

5.2.23.5 生命周期阶段

lifecycle_stage(生命周期阶段)是一种 relationship,它显示 possible_individual 对某些 possible_individual 的影响。

示例: 可以用 lifecycle_stage 的实例表达把可能的建筑物连接到 XYZ 公司临时部分的关系。通过对可用 class_of_lifecycle_stage 的分类,可以表达 lifecycle_stage 的本质(例如:“计划的”)。

EXPRESS 描述:

*)

```
ENTITY lifecycle_stage
    SUBTYPE OF (relationship);
    interest : possible_individual;
    interested : possible_individual;
END_ENTITY;
```

(*

属性定义:

interest: 影响被引用 possible_individual 的 possible_individual。

interested: 在被引用 possible_individual 中,具有影响的 possible_individual。

5.2.24 可能角色和预期角色

本条声明了表达可能角色和预期角色的实体数据类型。

注: 图 200 是本条定义的实体数据类型图(见 4.8.4.8.1)。

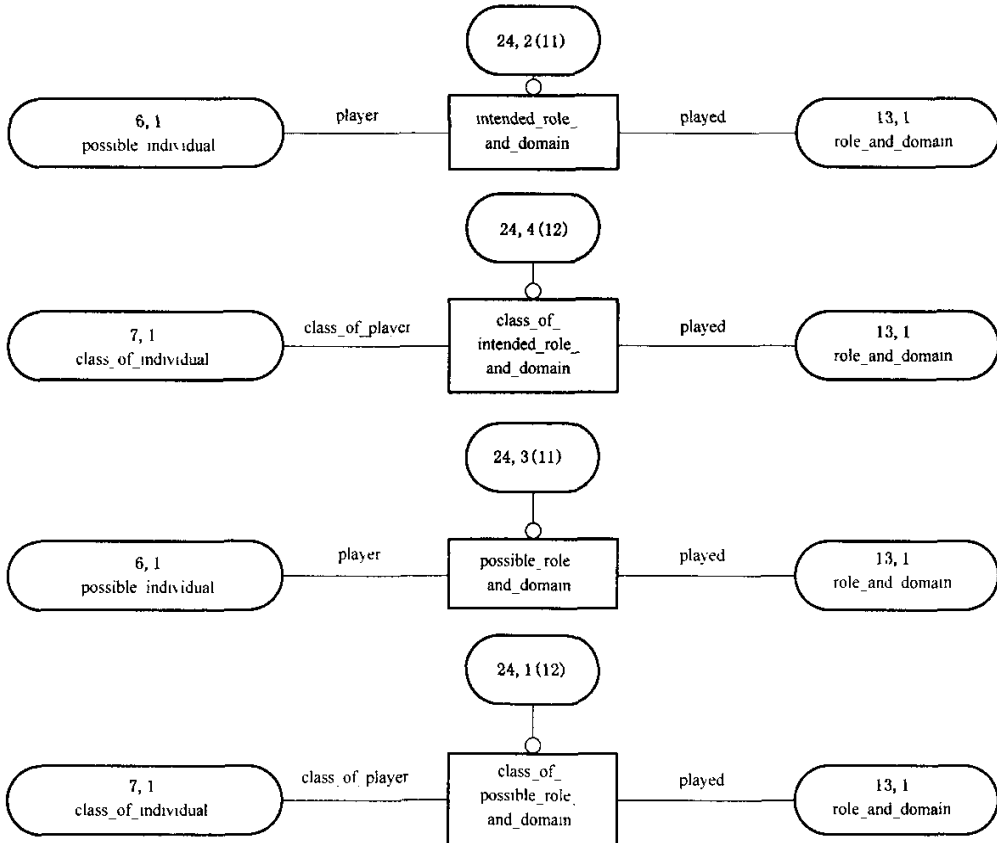


图 200 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 24 个)

5.2.24.1 预期角色和域类

class_of_intended_role_and_domain(预期角色和域类)是一种 class_of_relationship,它显示 class_of_individual 的成员预期将担当 role_and_domain 的成员。

示例:在某些抽吸活动中,泵预期将扮演执行者的 role_and_domain。

EXPRESS 描述:

*)

ENTITY class_of_intended_role_and_domain

SUBTYPE OF(class_of_relationship);

class_of_player ; class_of_individual;

played ; role_and_domain;

END_ENTITY;

(*

属性定义:

class_of_player:其成员可以扮演预期 role_and_domain 的 class_of_individual。

played:class_of_individual 成员预期将扮演的 role_and_domain。

5.2.24.2 可能角色和域类

class_of_possible_role_and_domain(可能角色和域类)是一种 class_of_relationship,它显示在某些 activity 中可以用 class_of_individual 的成员扮演 role_and_domain。

示例:泵可以扮演锚的角色(虽然它们预期不会这么做)。

EXPRESS 描述:

*)

ENTITY class_of_possible_role_and_domain

SUBTYPE OF(class_of_relationship);

class_of_player ; class_of_individual;

played ; role_and_domain;

END_ENTITY;

(*

属性定义:

class_of_player:其成员可以扮演被引用 role_and_domain 的 class_of_individual。

played:被引用 class_of_individual 成员可以扮演的 role_and_domain。

5.2.24.3 预期角色和域

intended_role_and_domain(预期角色和域)是一种 relationship,它显示 role_and_domain(possible_individual 的某些临时部分)预期获得相关的某些 activity。

示例:在某些抽吸活动中,某些分为泵类的 possible_individual 预期扮演执行者的 role_and_domain。

EXPRESS 描述:

*)

ENTITY intended_role_and_domain

SUBTYPE OF(relationship);

played ; role_and_domain;

player ; possible_individual;

END_ENTITY;

(*

属性定义:

played:被引用 possible_individual 预期扮演的 role_and_domain。

player: 预期扮演被引用 role_and_domain 的 possible_individual。

5.2.24.4 可能角色和域

possible_role_and_domain(可能角色和域)是一种 relationship,它显示扮演者 possible_individual 可能可以扮演被扮演者 role_and_domain。

示例: 扮演锚是泵 1234 的可能角色。

EXPRESS 描述:

```

* )
ENTITY possible_role_and_domain
  SUBTYPE OF(relationship);
  played ; role_and_domain;
  player ; possible_individual;
END_ENTITY;
    
```

(*

属性定义:

played: possible_individual 可以扮演的 role_and_domain。

player: 可以扮演 role_and_domain 的 possible_individual。

5.2.25 集运算

本条声明了表达集运算的实体数据类型。

注: 图 201 是本条定义的实体数据类型图(见 4.8.5.2.1 和 4.9)。

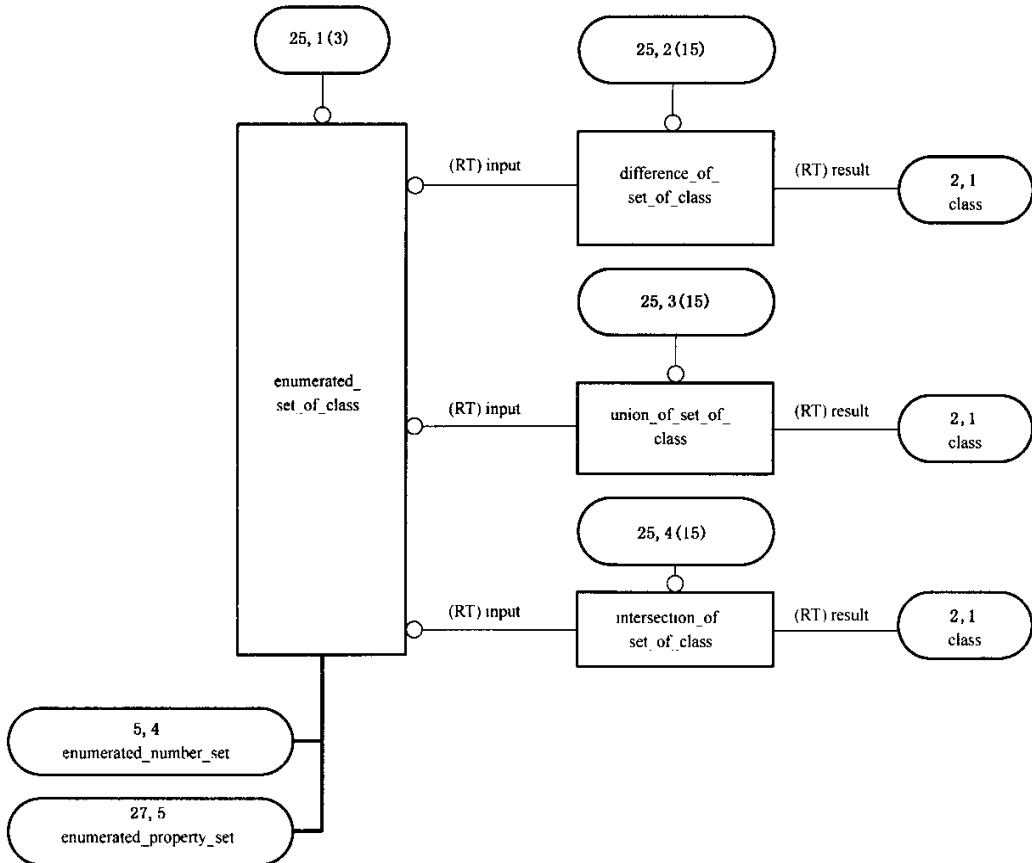


图 201 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 25 个)

5.2.25.1 类集的差运算

difference_of_set_of_class(类集的差运算)是一种 functional_mapping,它显示结果 class 的成员数是类并集成员数(是 enumerated_set_of_class 的成员)与其交集之间的差。

注:当 enumerated_set_of_class 是由一个类和另一个类(是第一个类的子类)组成时,那么差集是除该子类以外的其他类。

示例:enumerated_set_of_class{{A,B,C},{B,C,D},{C,D,E}}的差集是{A,B,D,E}。

EXPRESS 描述:

*)

```
ENTITY difference_of_set_of_class
  SUBTYPE OF(functional_mapping);
  SELF\functional_mapping.input : enumerated_set_of_class;
  SELF\functional_mapping.result : class;
END_ENTITY;
```

(*

属性定义:

input:是差函数域的 enumerated_set_of_class。

result:是差函数值域的 class。

5.2.25.2 枚举类集

enumerated_set_of_class(枚举类集)是一种枚举类实例集的 class_of_class。枚举意味着指定了整个成员集。

示例:{Plastic, 1.2kg, frame}是一个 enumerated_set_of_class。更通用的{{A,B,C},{B,C,D},{C,D,E}}是一个 enumerated_set_of_class。“1.1 版本的 ERDL 的电机工程类”也是一个 enumerated_set_of_class。

EXPRESS 描述:

*)

```
ENTITY enumerated_set_of_class
  SUBTYPE OF(class_of_class);
END_ENTITY;
```

(*

5.2.25.3 类集的交运算

intersection_of_set_of_class(类集的交运算)是一种 functional_mapping,它显示结果 class 由每个类都通用的 enumerated_set_of_class 类成员组成。

示例:enumerated_set_of_class{{A,B,C},{B,C,D},{C,D,E}}的交集是{C}。

EXPRESS 描述:

*)

```
ENTITY intersection_of_set_of_class
  SUBTYPE OF(functional_mapping);
  SELF\functional_mapping.input : enumerated_set_of_class;
  SELF\functional_mapping.result : class;
END_ENTITY;
```

(*

属性定义:

input:其成员将做交运算的 enumerated_set_of_class。

result:表达 enumerated_set_of_class 成员交集的 class。

5.2.25.4 类集的并运算

union_of_set_of_class(类集的并运算)是一种 functional_mapping,它显示结果 class 的成员数是 enumerated_set_of_class 类成员的和。

示例: enumerated_set_of_class {{A,B,C},{B,C,D},{C,D,E}}的并集是{A,B,C,D,E}。

EXPRESS 描述:

*)

```
ENTITY union_of_set_of_class
    SUBTYPE OF(functional_mapping);
    SELF\functional_mapping.input : enumerated_set_of_class;
    SELF\functional_mapping.result : class;
END_ENTITY;
```

(*

属性定义:

input:是并函数域的 enumerated_set_of_class。

result:是并函数值域的 class。

5.2.26 属性

本条声明了表达属性的实体数据类型。

注:图 202 是本条定义的实体数据类型图(见 4.8.4.3)。

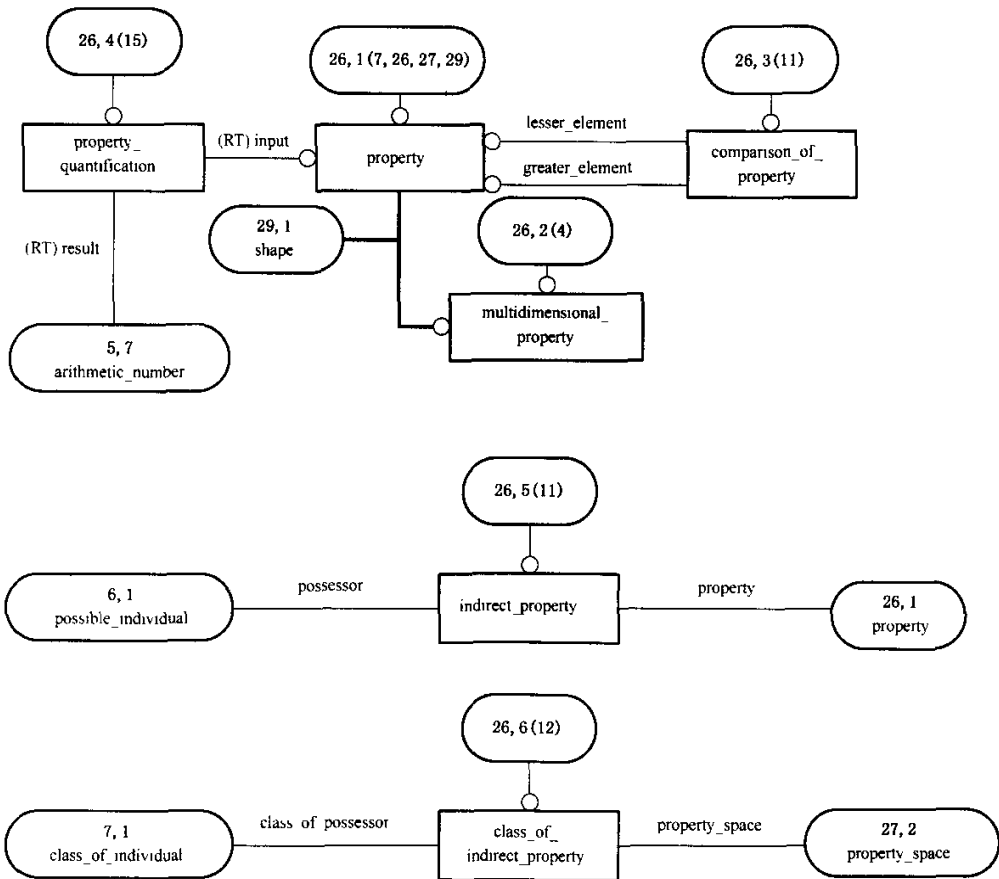


图 202 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 26 个)

5.2.26.1 间接属性类

class_of_indirect_property(间接属性类)是一种 class_of_relationship,它显示 class_of_individual 的成员可以把 class_of_property 的成员作为这种类型的 indirect_property 拥有。

示例:最大允许工作压力是压力显示的 class_of_indirect_property,压力容器可以拥有它。

EXPRESS 描述:

*)

```
ENTITY class_of_indirect_property
  SUBTYPE OF(class_of_relationship);
  class_of_possessor : class_of_individual;
  property_space : property_space;
END_ENTITY;
```

(*

属性定义:

class_of_possessor:其实例可以拥有一个 property_space 成员的 class_of_individual。

property_space:class_of_individual 成员可以拥有其成员的 property_space。

5.2.26.2 比较属性

comparison_of_property(比较属性)是一种 relationship,它显示一个属性的数量大于另一个属性的数量。

示例:可以用 comparison_of_property 的实例显示房间的温度低于熔炉中的温度。

EXPRESS 描述:

*)

```
ENTITY comparison_of_property
  SUBTYPE OF(relationship);
  greater_element : property;
  lesser_element : property;
END_ENTITY;
```

(*

属性定义:

greater_element:是 comparison_of_property 中较大元素的 property。

lesser_element:是 comparison_of_property 中较小元素的 property。

5.2.26.3 间接属性

indirect_property(间接属性)是 property 与 possible_individual 之间的 relationship。通过用 class_of_indirect_property 对其分类定义 indirect_property 的本质。

当属性不直接应用于其应用的 possible_individual,而是来源于某些方法时,属性是间接的。

注:因为属性不直接应用,所以它是间接的。某个时间点事物只能有一个温度,因此最大允许工作温度不是它的温度,而是一个来源于某些试验或为了确定其值而计算的间接属性(与其相对的是当前测量值)。这是使其间接的原因。

示例:通过 50 barA 压力与 V101 之间的 indirect_property 规定 V101 的最大允许工作压力 50 barA,通过 class_of_indirect_property 最大允许工作压力对它进行分类。

EXPRESS 描述:

*)

```
ENTITY indirect_property
  SUBTYPE OF(relationship);
```

```

    possessor : possible_individual;
    property : property;
END_ENTITY;

```

(*

属性定义:

possessor: 拥有 indirect_property 的 possible_individual。

property: possible_individual 间接拥有的 property。

5.2.26.4 多维属性

multidimensional_property(多维属性)是一种 property,也是一个 multidimensional_object。

示例: 泵的水头特性是一个 multidimensional_object。它由一对连续的 Q、H 属性组成,其中 Q 是流量,H 是水头差。每对属性 Q_i 和 H_i 是一个 multidimensional_property[Q_i, H_i],其中 Q_i 是特殊流量,H_i 是特殊水头。

EXPRESS 描述:

*)

```

ENTITY multidimensional_property
    SUBTYPE OF(property, multidimensional_object);
END_ENTITY;

```

(*

5.2.26.5 属性

property(属性)是一种 class_of_individual,它是连续 class_of_property 的成员。通过按一定比例映射到数字上可以量化 property。

注 1: property 的成员是一种 possible_individual,它与其他成员一样在 property 表达的质量或特性方面具有相同的维度或数量。

注 2: 特性或质量的类型是 class_of_property 的实例,例如温度或密度。

注 3: 不应该在相同的数据仓库中生成属性副本(例如按相同比例映射到相同的数)。

EXPRESS 描述:

```

ENTITY property
    SUBTYPE OF(class_of_individual);
END_ENTITY;

```

(*

5.2.26.6 属性量化

property_quantification(属性量化)是一种 functional_mapping,其成员把 property 映射到 arithmetic_number。

示例: 可以用 property_quantification 的实例表达把具体质量映射到数字 4.2 的连接。

注 1: 通过 class_of_representation_of_thing 连接 arithmetic_number 和 class_of_EXPRESS_information_representation 的连接完成数字的实际表达。

注 2: 通过对依据比例的 property_quantification 进行分类,给定量化的单位或比例。

EXPRESS 描述:

*)

```

ENTITY property_quantification
    SUBTYPE OF(functional_mapping);
    SELF\functional_mapping.input : property;
    SELF\functional_mapping.result : arithmetic_number;
END_ENTITY;

```

(*)

属性定义:

input: 被引用 arithmetic_number 所量化的 property。

result: 量化被引用 property 的 arithmetic_number。

5.2.27 属性类型

本条声明了表达属性类型的实体数据类型。

注: 图 203 是本条定义的实体数据类型图(见 4.8.4.3)。

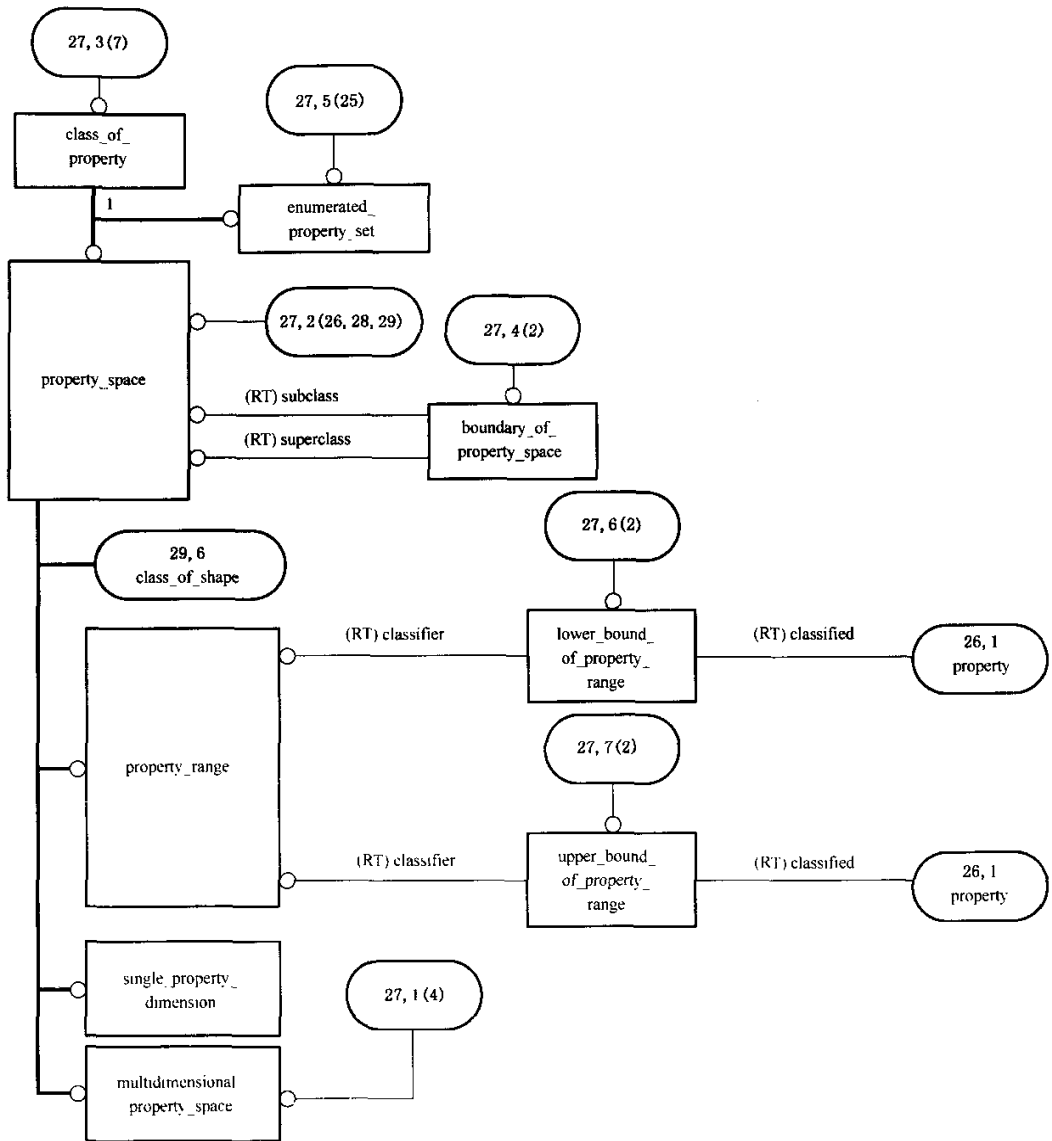


图 203 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 27 个)

5.2.27.1 属性空间边界

boundary_of_property_space(属性空间边界)是一种显示子类成员构成超类边界的 specialization。

示例: 符合最大速度水头曲线的 property_space 是符合泵作业范围 property_space 的一个边界。

EXPRESS 描述:

*)

```
ENTITY boundary_of_property_space
    SUBTYPE OF(specialization);
    SELF\specialization. subclass : property_space;
END_ENTITY;
```

(*

属性定义:

subclass: 其成员构成超类属性所引用 property_space 的边界的 property_space。

superclass: 子类属性引用的 property_space 成员所界限的 property_space。

5.2.27.2 属性类

class_of_property(属性类)是一种 class_of_class_of_individual, 其成员是 property 的实例。

示例: “温度”是 class_of_property 的一个例子。

EXPRESS 描述:

*)

```
SUPERTYPE OF (ONEOF(property_space, enumerated_property_set))
    SUBTYPE OF(class_of_class_of_individual);
END_ENTITY;
```

(*

5.2.27.3 枚举属性集

enumerated_property_set(枚举属性集)是一种 class_of_property 和 enumerated_set_of_class, 其成员是相同 single_property_dimension 或 multidimensional_property_space 的枚举属性集。

EXPRESS 描述:

```
ENTITY enumerated_property_set
    SUBTYPE OF(class_of_property, enumerated_set_of_class);
END_ENTITY;
```

(*

5.2.27.4 属性范围下界

lower_bound_of_property_range(属性范围下界)是一种显示属性是 property_range 下界的 class_of_classification。

示例: $-10\text{ }^{\circ}\text{C}$ 是范围 $-10\text{ }^{\circ}\text{C}\sim 20\text{ }^{\circ}\text{C}$ 的下界。

EXPRESS 描述:

*)

```
ENTITY lower_bound_of_property_range
    SUBTYPE OF(classification);
    SELF\classification. classified : property;
    SELF\classification. classifier : property_range;
END_ENTITY;
```

(*

属性定义:

classified: 作为被分类对象, 是 lower_bound_of_property_range 下界的 property。

classifier: 作为分类者, 在 lower_bound_of_property_range 中被界限的 property_range。

5.2.27.5 多维属性空间

multidimensional_property_space(多维属性空间)是一种 property_space 和 multidimensional_object, 其成员是每个可以映射到多个数字的属性。每个属性将由具有相同属性维数的元素组成。

示例: 泵流量和水头差的特性曲线是一个 multidimensional_property_space。

EXPRESS 描述:

*)

ENTITY multidimensional_property_space

SUBTYPE OF(property_space, multidimensional_object);

END_ENTITY;

(*

5.2.27.6 属性范围

property_range(属性范围)是一种是 single_property_dimension 连续性子集的 property_space。

示例: $-10\text{ }^{\circ}\text{C} \sim +20\text{ }^{\circ}\text{C}$ 是温度的一个 property_range。

EXPRESS 描述:

*)

ENTITY property_range

SUBTYPE OF(property_space);

END_ENTITY;

5.2.27.2 属性空间

property_space(属性空间)是一种 class_of_property, 其成员是具有一致连续性的 property。

示例 1: 温度属性集是温度的一个 property_space。

示例 2: 特殊泵曲线上的压力和流量 class_of_property 成员是一个 property_space。

EXPRESS 描述:

*)

ENTITY property_space

SUBTYPE OF(class_of_property);

END_ENTITY;

(*

5.2.27.8 一维属性

single_property_dimension(一维属性)是一种 property_space, 它是每个属性映射到单个数字的单一并完全连续的属性。

示例: 温度、压力粘性和长度是 single_property_dimension 的例子。

EXPRESS 描述:

*)

ENTITY single_property_dimension

SUBTYPE OF(property_space);

END_ENTITY;

5.2.27.9 属性范围上界

upper_bound_of_property_range(属性范围上界)是一种显示 property 是 property_range 上界的 class_of_classification。

示例: $+20\text{ }^{\circ}\text{C}$ 是范围 $-10\text{ }^{\circ}\text{C} \sim +20\text{ }^{\circ}\text{C}$ 的上界。

EXPRESS 描述:

*)


```

ENTITY upper_bound_of_property_range
  SUBTYPE OF(classification);
  SELF\classification.classified : property;
  SELF\classification.classifier : property_range;
END_ENTITY;

```

(*

属性定义:

classified: 作为被分类对象, 是 upper_bound_of_property_range 规定的上界的 property_range。

classifier: 作为分类者, 具有 upper_bound_of_property_range 规定的上界的 property_range。

5.2.28 转换比例

本条声明了表达转换比例的实体数据类型。

注: 图 204 是本条定义的实体数据类型图(见 4.8.4.3.2)。

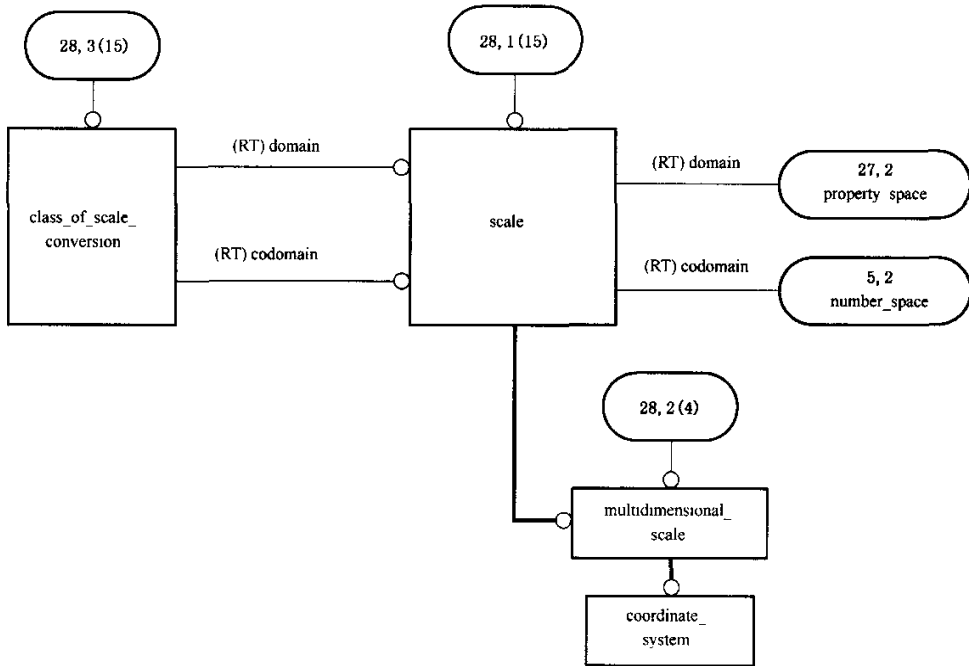


图 204 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 28 个)

5.2.28.1 转换比例类

class_of_scale_conversion(转换比例类)是一种 class_of_isomorphic_functional_mapping, 它定义了用于属性量化的两种不同单位比例之间的转换。

示例: 都可以用 scale 的实例表达温度的华氏比例和温度的摄氏比例。可以用 class_of_scale_conversion 的实例表达这两个比例之间的转换。

EXPRESS 描述:

(*

```

ENTITY class_of_scale_conversion
  SUBTYPE OF(class_of_isomorphic_functional_mapping);
  SELF\class_of_functional_mapping.codomain : scale;
  SELF\class_of_functional_mapping.domain : scale;
END_ENTITY;

```

(*

属性定义:

codomain: 声称要转换的第二个 scale。

domain: 声称要转换的第一个 scale。

5.2.28.2 坐标系

coordinate_system(坐标系)是在 n 维空间(在其中任意的几何转换都是有效的)中定位和关联 possible_individual 的 multidimensional_scale。

示例: XYZ 位置的坐标系是 coordinate_system 的一个例子。

EXPRESS 描述:

*)

```
ENTITY coordinate_system
    SUBTYPE OF(multidimensional_scale);
END_ENTITY;
```

(*

5.2.28.3 多维比例

multidimensional_scale(多维比例)是一种也是 multidimensional_object 的 scale。

示例: [摄氏度, 秒]比例是一个可以在其上绘出所有时间-温度变化的 multidimensional_scale。

EXPRESS 描述:

*)

```
ENTITY multidimensional_scale
    SUBTYPE OF(scale, multidimensional_object);
END_ENTITY;
```

(*

5.2.28.4 比例

scale 是一种 class_of_isomorphic_functional_mapping, 其成员是 property_quantification 的成员。在询问中, 它显示一个 property_space 映射到 scale 上的 number_space。

示例: 可以用 scale 的实例表达 class_of_number[-273, inf]与 class_of_property 温度之间的摄氏度比例连接。

EXPRESS 描述:

*)

```
ENTITY scale
    SUBTYPE OF(class_of_isomorphic_functional_mapping);
    SELF\class_of_functional_mapping, codomain : number_space;
    SELF\class_of_functional_mapping, domain : property_space;
END_ENTITY;
```

(*

属性定义:

codomain: 其成员可以量化被引用 property_space 成员的 number_space。

domain: 被引用 class_of_number 成员可以量化其成员的 class_of_property。

5.2.29 形状

本条声明了表达形状实体数据类型的。

注: 图 205 是本条定义的实体数据类型图(见 4.8.4.5)。

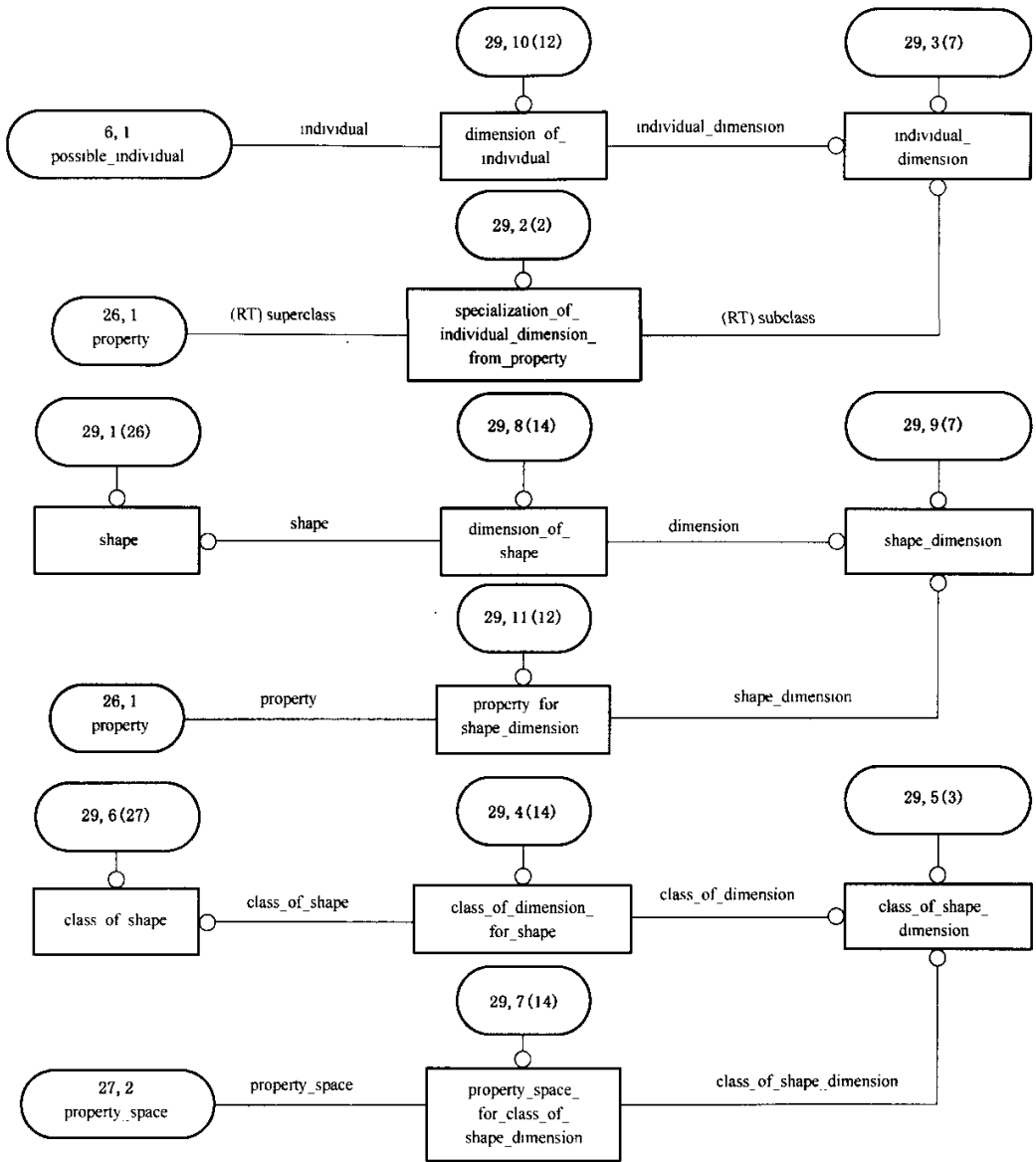


图 205 生命周期集成模式的 EXPRESS-G 图(29 个图中的第 29 个)

5.2.29.1 形状维度类

class_of_dimension_for_shape(形状维度类)是一种 class_of_class_of_relationship,它显示 class_of_shape 的成员具有 class_of_dimension 成员的维度。

示例：指定“圆类”的成员有“直径类”的成员是 class_of_dimension_for_shape 的一个实例。

EXPRESS 描述：

*)

```

ENTITY class_of_dimension_for_shape
  SUBTYPE OF(class_of_class_of_relationship);
  class_of_dimension ; class_of_shape_dimension;
  class_of_shape ; class_of_shape;
  
```

END_ENTITY;

(*

属性定义:

class_of_dimension; class_of_dimension_for_shape 中的 class_of_shape_dimension。

class_of_shape; class_of_dimension_for_shape 中的 class_of_shape。

5.2.29.2 形状类

class_of_shape(形状类)是一种 property_space,其成员是 shape 的实例。

EXPRESS 描述:

ENTITY class_of_shape

SUBTYPE OF(property_space);

END_ENTITY;

(*

5.2.29.3 形状维度类

class_of_shape_dimension(形状维度类)是一种是 class_of_shape 维度的 class_of_class。

示例:直径、高度和宽度(通常多于具体的一个)是 class_of_shape_dimension 的例子。

EXPRESS 描述:

ENTITY class_of_shape_dimension

SUBTYPE OF(class_of_class);

END_ENTITY;

(*

5.2.29.4 个体的维度

dimension_of_individual(个体的维度)是一种 class_of_relationship,它显示 individual_dimension 的每个线集成员是 possible_individual 的维度。

示例:通过具体圆的圆心并在该圆的圆周上结束的所有线的集是该圆的维度(直径)。通过对 dimension_of_individual 分类的 dimension_of_shape 显示具体的维度。

EXPRESS 描述:

ENTITY dimension_of_individual

SUBTYPE OF(class_of_relationship);

individual ; possible_individual;

individual_dimension ; individual_dimension;

END_ENTITY;

(*

属性定义:

individual;赋给 dimension_of_individual 中 individual_dimension 的 possible_individual。

individual_dimension;dimension_of_individual 中 possible_individual 的 individual_dimension。

5.2.29.5 形状的维度

dimension_of_shape(形状的维度)是一种 class_of_class_of_relationship,它显示 shape_dimension 的成员是 shape 成员的维度。

示例:是 10 m 圆的直径的 10 m 线集是 dimension_of_shape 的一个例子。

EXPRESS 描述:

*)

ENTITY dimension_of_shape

SUBTYPE OF(class_of_class_of_relationship);

```
dimension : shape_dimension;  
shape : shape;  
END_ENTITY;
```

(*

属性定义:

dimension: shape 的 shape_dimension。
shape: 拥有 shape_dimension 的 shape。

5.2.29.6 个体维度

individual_dimension(个体维度)是一种 class_of_individual,其成员特性化一个特殊 possible_individual。

示例: 线集是特殊圆的所有直径。

EXPRESS 描述:

```
* )  
ENTITY individual_dimension  
    SUBTYPE OF(class_of_individual);  
END_ENTITY;  
( *
```

5.2.29.7 形状维度的属性

property_for_shape_dimension(形状维度的属性)是一种 class_of_class_of_relationship,它显示 shape_dimension 的成员是 property。

示例: 10 m 直径是 10 m 长。

EXPRESS 描述:

```
* )  
ENTITY property_for_shape_dimension  
    SUBTYPE OF(class_of_relationship);  
    property : property;  
    shape_dimension : shape_dimension;  
END_ENTITY;  
( *
```

属性定义:

property: property_for_shape_dimension 的 property。: the for the shape_dimension;其成员是 property_for_shape_dimension 中 property 具体化的 shape_dimension。

5.2.29.8 形状维度类的属性空间

property_space_for_class_of_shape_dimension(形状维度类的属性空间)是一种显示 property_space(class_of_shape_dimension 来自它)的 class_of_class_of_relationship。

示例: 直径是一个长度维度。

EXPRESS 描述:

```
* )  
ENTITY property_space_for_class_of_shape_dimension  
    SUBTYPE OF(class_of_class_of_relationship);  
    class_of_shape_dimension : class_of_shape_dimension;  
    property_space : property_space;  
END_ENTITY;
```

(*

属性定义:

class_of_shape_dimension: 其成员在被引用 property_space 中具有一个 property 的 class_of_shape_dimension。

property_space: 是 class_of_shape_dimension 源头的 property_space。

5.2.29.9 形状

shape(形状)是一种依赖于位置的不变关系,并依赖于所有点(构成了它的轮廓或其外表面)之间成比例的距离的 property。

示例 1: 20 mm 直径的圆和 10 mm~20 mm 直径的圆是 shape 的例子。

示例 2: 可以用 shape 的实例表达不规则形状,例如:泵模型的外部封套。

EXPRESS 描述

*)

ENTITY shape

 SUBTYPE OF(property);

END_ENTITY;

(*

5.2.29.10 形状维度

shape_dimension(形状维度)是一种定义形状方面 individual_dimension 集的 class_of_class_of_individual。

示例: 5 m 的直径、3 mm 的高度和 10 cm 的宽度是 shape_dimension 的成员。

EXPRESS 描述:

*)

ENTITY shape_dimension

 SUBTYPE OF(class_of_class_of_individual);

END_ENTITY;

(*

5.2.29.11 属性个体维度的专门化

specialization_of_individual_dimension_from_property(属性个体维度的专门化)是一种显示维度成员是属性成员的 specialization。

示例: 10 m 的直径是 10 m 长。

EXPRESS 描述:

*)

ENTITY specialization_of_individual_dimension_from_property

 SUBTYPE OF(specialization);

 SELF\specialization. subclass : individual_dimension;

 SELF\specialization. superclass : property;

END_ENTITY;

(*

属性定义:

subclass: 是特殊化的 individual_dimension。

superclass: 是通用化的 property。

*)

END_SCHEMA;

附录 A
(规范性附录)
信息对象注册

A.1 文档标识

为了给开放系统中信息对象提供明确标识,对象标识符

{iso standard 15926 part{2} version {1}}

表示 GB/T 18975 的本部分。GB/T 16262.1 定义了值的含义,并在 GB/T 18975.1 中进行了描述。

注:它是本部分的版本对象标识符。

A.2 模式标识

为了给开放信息系统中提供 process_plant_lifecycle_schema 的明确标识符,对象标识符

{ iso standard 15926 part(2) version(1) object(1) lifecycle-integration-schema(1) }

赋值给 lifecycle_integration_schema (见第 5 章)。GB/T 16262.1 定义了值的含义,并在 GB/T 16656.1 中进行了描述。

注:它是该模式的版本对象标识符。

附录 B
(资料性附录)
计算机可解释的列表

本附录引用了本部分描述的 EXPRESS 模式列表,没有说明或其他的解释性文本。该列表是可以由计算机解释的表格,并可在下面的网址中找到:

<http://www.tc184-sc4.org/EXPRESS/>

如果不能访问这些站点,请直接使用信箱. sc4sec@tc184-sc4.org 与 ISO 中心秘书处联或 ISO TC 184/SC4 秘书处联系。

注:上面 URL 中,计算机可解释的表格所提供的信息是资料性的。本部分包含的信息是规范性的。

附录 C
(资料性附录)

GB/T 16656.11 EXPRESS 应用

在第 5 章中定义的数据模型由 GB/T 16656.11 中定义的 EXPRESS 语言来描述,然而,并不是这种语言的所有的特征都可用。下列 EXPRESS 结构就不在数据模型规范范围内。

- 结构数据类型;
- 概括性数据类型;
- 选择性数据类型;
- 参数数据类型——排列;
- 集合类型;
- 继承属性;
- 反面属性;
- 域规则;
- 全局规则;
- 运算法则;
- 常数。

实体数据类型和属性名称遵照 GB/T 16656.11,没有附加的限制。

附录 D (资料性附录)

GB/T 18975 中集合理论的一些注

D.1 引言

该附录解释了 GB/T18975 本部分所支持的集合理论的一些原理。

D.1.1 什么是集合

一个集合就是拥有成员的一种事务,且由所有成员定义(空集是没有成员的集合)。也就是说,如果两个集合有同样的成员,它们就是同样的集合。如果有不同的成员,就是不同的集合。同时,一个集合由它的成员定义,不是所有的成员都是可知的。

D.1.2 集合与 GB/T 18975

GB/T 18975 中的时间-空间范例意味着对象是不变的,变更通过 possible_individual 来处理,possible_individual 是所表达状态的 whole_life_individual 的一个临时部分。它给出了基于时间之外的状态模型,而不是当前的状态,也就意味着所有的事务都是不变的。

例如:一辆汽车有时是红色的,有时是蓝色的(意味着汽车的集合元素已经改变),它有两种状态,一个是红色,一个是蓝色。这些集合元素并不取决于你在过去、现在和将来所看见的。甚至当你从开始就关注,它将来的状态也可能是蓝色的,而且非常有可能你并不知道这些。

D.2 一些不同类别的集合理论

一个集合就是拥有成员的一种事务,且由成员数定义(空集是没有成员的集合)。也就是说,如果两个集合有同样的成员,它们就是同样的集合。如果有不同的成员,就是不同的集合。同时,一个集合由他的成员定义,不是所有的成员都是可知的。

D.2.1 单层集合

单层集合是一个有集合和集合元素的系统,但是集合本身不能成为集合的成员。这种情况符合实体关系模型,实体类型不能是其他实体类型的成员,见图 D.1。

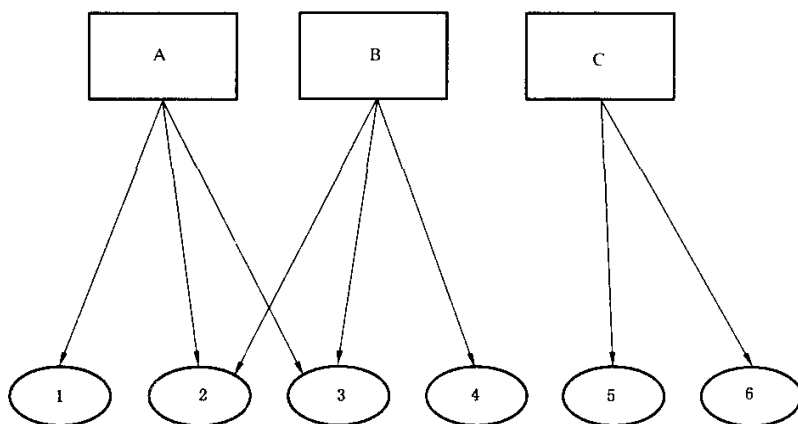


图 D.1 单层集合

某种情况下,甚至不允许元素成为多个集合的元素。

D.2.2 多级集合

多级集合是一级集合可以作为上级集合的成员,但没有层级的交叉,见图 D.2

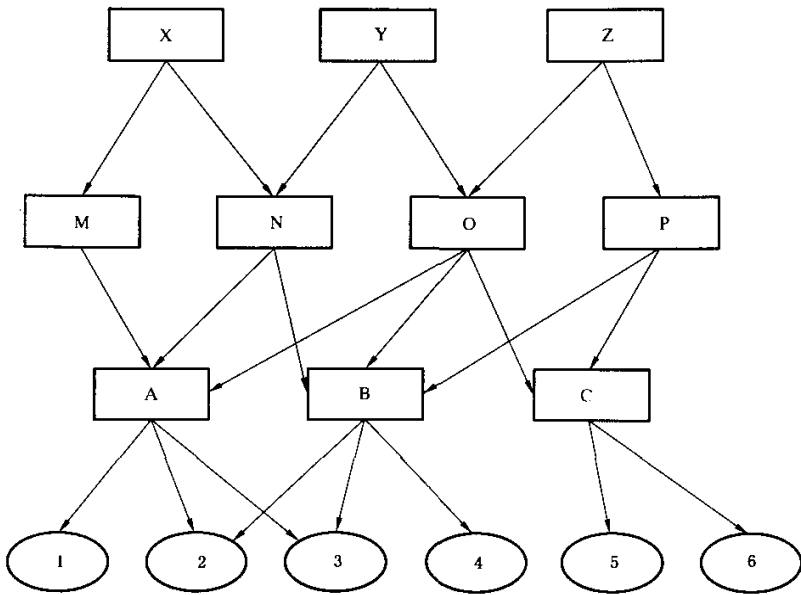


图 D.2 层次集的示例

多集集合的一个应用的例子是在数据模型、meta-model, meta-meta-model 方法多级集合中,在本部分中通过实体类型 individual, class_of_individual 及 class_of_class_of_individual 可发现。

D.2.3 Well-founded 集合

Well-founded 集合是标准集合理论的集合,例如 Zermelo-Fraenkel (ZF)集合理论,或可在标准文本中可找到的 von Neuman, Bernays, Goedel (VNBG)集合理论。Well-founded 集合可从它们下层的集合中获得元素,但是不可以循环(即:一个集合作为它自己的元素)。图 D.3 表明了一个 well-founded 集合系统的例子。

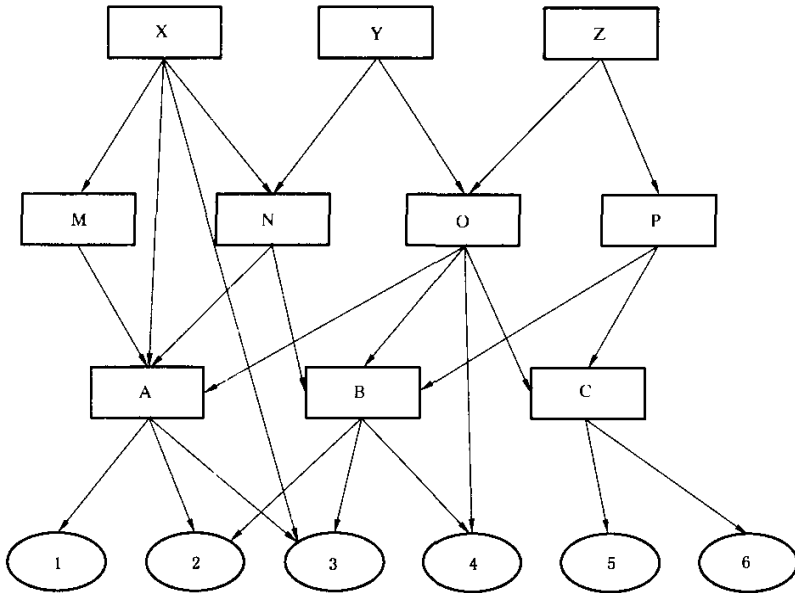


图 D.3 well-founded 集合的示例

集合理论的这种形式在处理 Russell's 的矛盾时被广泛地应用,如果集合是可以作为本身的成员,那么在某些情况下就可能出现矛盾,例如所有不包含本身的集合的集合不存在。

D.2.4 Non-well-founded 集合理论

non-well-founded 集合理论的本质是允许集合成为本身的成员,结构图见图 D.4。

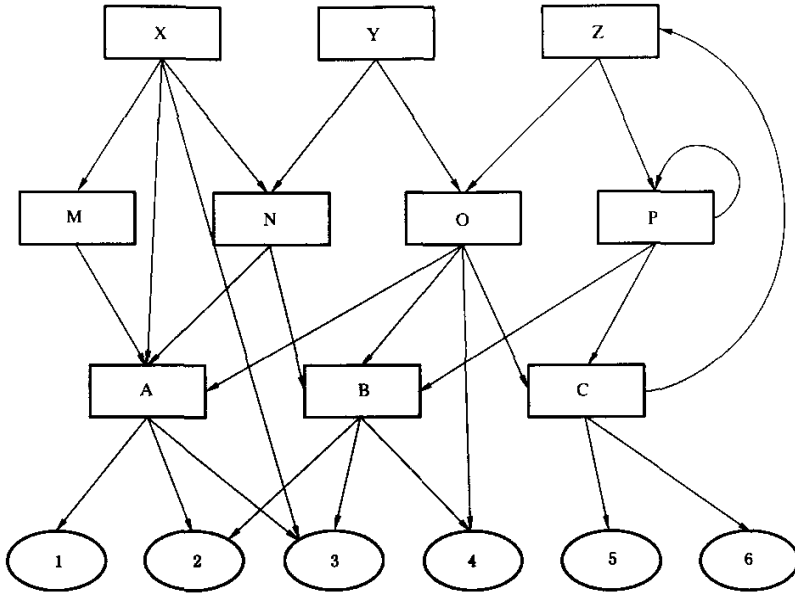


图 D.4 non-well-founded 集合的示例

通过强调集合的构造避免了 Russell's 矛盾,承认 well-founded 集合是有用的事务防止了“类是一个类”,“事务是一个类”“类是一个事务”情况的出现。

D.3 注

应该注明集合的每种类型都是下列集合类型的子集,由于 GB/T 18975 承认 non-well-founded 集合,它也支持这里提及的所有其他的集合类型。

附录 E
(资料性附录)

分析 association 的应用和含义

E.1 引言

传统上许多数据模型都采用快照观点,这就意味着当发生改变的时候,历史就会被覆盖而不存在了。EPISTLE 与数据模型相联系,该数据模型可以管理工厂整个生命周期流程过程的信息。这就意味着 EPISTLE 可以管理过去、现在和将来的信息。

ECM 已经历了许多版本,在早期的版本中,association 的应用支持 history。association 是一种掌握历史信息的方式,这种信息可能会随时间而改变,这种关系可由包括属性(有效期的开始和结束日期)的实体类型所代替,这种实体类型被称为 association,这种方法在[5]有所描述。

近来,EPISTLE Core 模型处理变更的方法本身也发生了改变。目前管理变更所采用的方法是通过识别个体事务在有效期内的不同状态以及在不同状态之间的永恒的关系。[6]中描述了这种方法的概要。第 5 条的概念模型中采用了此方法。变更的理由超出了附录的范围。然而,本附录没有考虑这两种方法如何互相关联。而且对于 association 如何转化为状态和永恒的关系,我们已经建立了四个基本的方式,一种方法用于分析 association 如何被表达。

注:本文第 5 章或 EPISTLE Core Model 中的实体类型的名字不是必须的。

E.2 从映射关系到 association

图 E.1 表达了物理对象 ownership 的两种建模方式。第一个模型 ownership 作为组织和物理对象之间的一种关系。当 ownership 变更时可能会出现一些问题。或者不允许变更,或者物理对象中存在的 ownership 属性被覆盖。意味着前面的 ownership 历史记录已经丢失。



图 E.1 从一个映射模型移动到使用了 associations 的模型

第二个模型 ownership 作为一个 association 实体类型。一个开始日期和一个结束日期作为其属性,且引用了它自己和所有者的属性。现在当 ownership 变更时,前面 ownership 关系的结束日期得到了一个结束日期,一个新的 ownership 记录产生了。结果是 ownership 历史能被保存下来。

E.3 从 association 到状态和类之间的关系

应用 association 似乎是解决了问题,但是通过更深层次的分析发现一些信息被隐藏了。为了揭开这个事实我们需要考虑时间空间范围的个体事务以及这些时间-空间范围内临时部分,它们之间的一些关系是真实的。为了帮助说明我们用空间-时间图表示,见图 E.2。

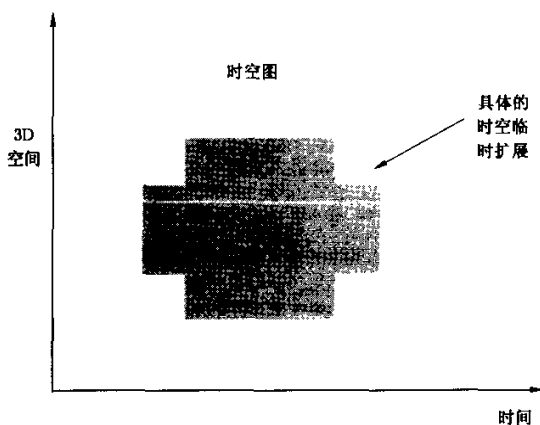


图 E. 2 时空图

已经发现五种不同的方式在空间-时间范围内用于表达不同种类的 association。这些将在下面的子部分中说明。

E. 3.1 方式 1: 个体和类状态之间的关系

图 E. 3 表示分类方式。给出了个体状态的例子, 汽车 1 从 1/1/2001 到 4/3/2001 时间内根据红色来分类。

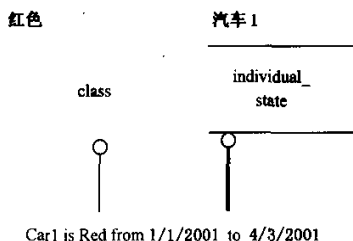


图 E. 3 分类 association 的示例

如果检查这里的空间-时间图的情况(见图 E. 4), 可以看出汽车 1 的状态是根据其颜色是红色时分类的(阴影部分)。

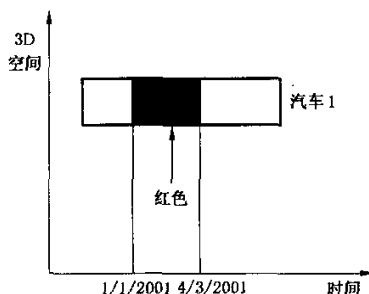


图 E. 4 表示了空间-时间的数据模型

图 E. 5 表明了使用状态的分类。

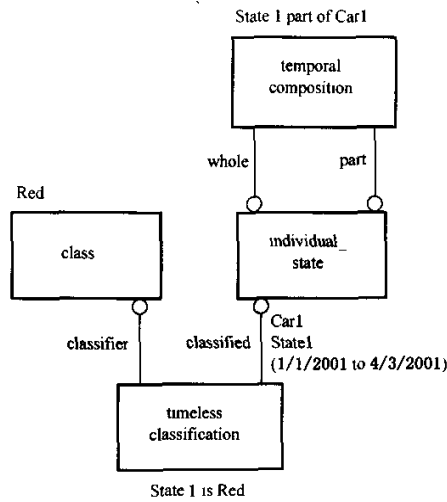


图 E.5 使用了状态的分类

这里汽车的状态被明确的模型化,不是作为分类 association 的部分,而且表明了作为整个汽车的一个临时部分,根据它是红色的来分类。当前这种分类关系是没有时间限制的,因为在分类的状态中作为红色的时间是明确的。

E.3.2 方式 2:一个个体两种状态之间的关系

图 E.6 表明了用一个 association 来表达个体事务两种状态之间的关系的情况,为了说明该模型,下面给出了一个轮子 1 如何成为汽车 1 的一部分的例子,日期为 1/1/2001 到 5/4/2001。

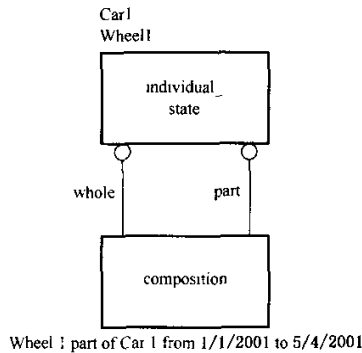


图 E.6 两个个体之间的关系

图 E.7 是一个空间-时间图形,表达了汽车和轮子不同的状态,以及汽车和轮子的整个生命状态。

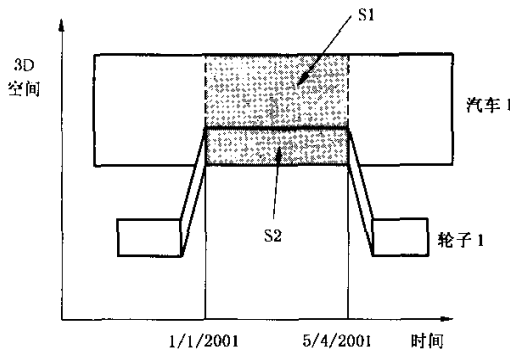


图 E.7 成分的时空图

该图表示汽车 1 的一个状态, S1; 轮子 1 的一个状态, S2, 它们有着同样的状态和结束日期, 而 S2 是 S1 的一部分。当这个空间-时间图被模型化时, 结果就如图 E. 8 所示。

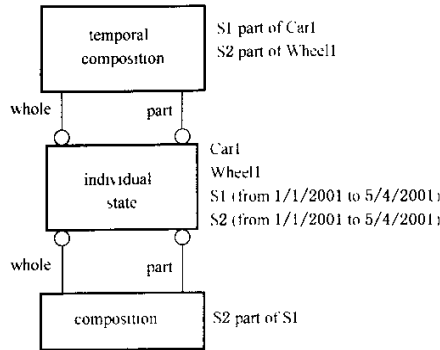


图 E. 8 使用了状态的成分

这里的 individual_states S1 和 S2 被模型化。S1 作为汽车 1 的临时部分, S2 作为轮子 1 的临时部分, 而 S2 是 S1 的一部分。

E. 3.3 方式 3: 一致的个体

图 E. 9 表示一个特定的泵如何被安排来完成特定的任务, TAG P101, 时间为 3/1/2000 到 5/8/2001。在这之后, 该泵被移走, 并且被另一个可以完成同样任务的泵所代替。

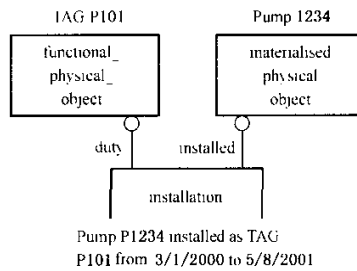


图 E. 9 一致个体

图 E. 10 是一个空间-时间图。可以看出任务由 TAG P101 表达, 泵 1234 在安排的时间内是一致的, 即泵 1234 作为 TAG P 101 被安排的的状态 S1 实际上也是 TAG P 101 的一个状态。确切地说, TAG P 101 是由在此处安排的泵的状态组成的。

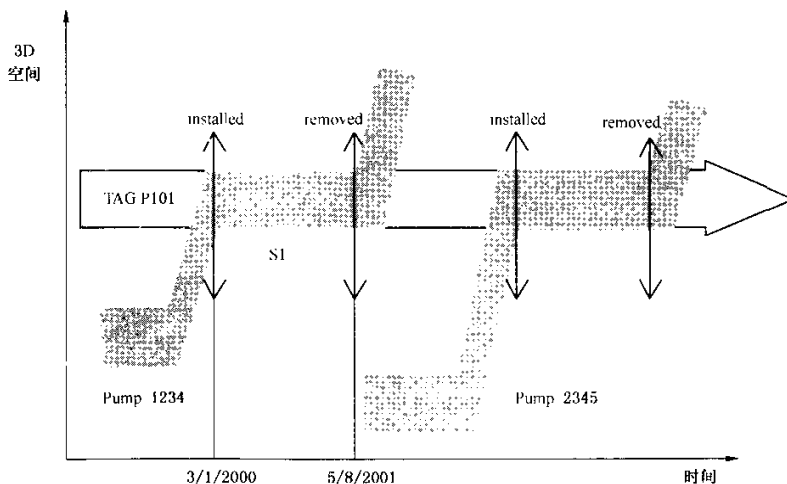


图 E. 10 一致个体的时空图

当它被模型化而不是作为一个联合体时,结果如图 E. 11 所示。这里的 S1 被看作为 TAG P 101 和泵 1234 的一个临时部分。

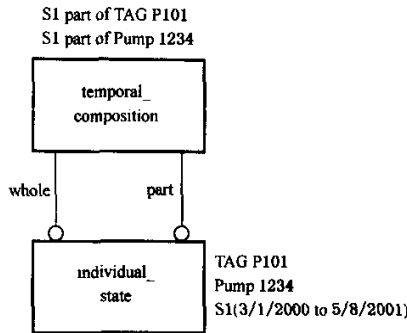


图 E. 11 使用了状态的一致个体

E. 3. 4 模式 4: 两个类之间的关系

图 E. 12 是一个特殊的联合体,例子中的离心泵是一个特殊泵。Association 自动给予这个特殊泵开始和结束的日期,但实际上类是永恒的,什么时候离心泵不是特殊泵并没有时间限制。

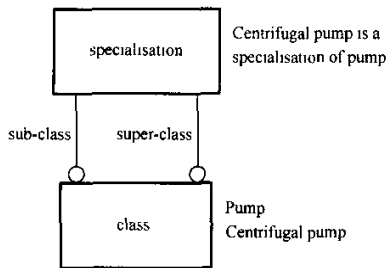


图 E. 12 两个类之间的关系

此种情况没有空间-时间图,由于所有的对象都与时间无关。然而,可以检查特殊 association 的意义,而且在该情况下可以确定特殊 association 表明的是子类的每个成员也是超类的一个成员。因此可把特例作为关系的一个子类,如图 E. 13 表示的那样。

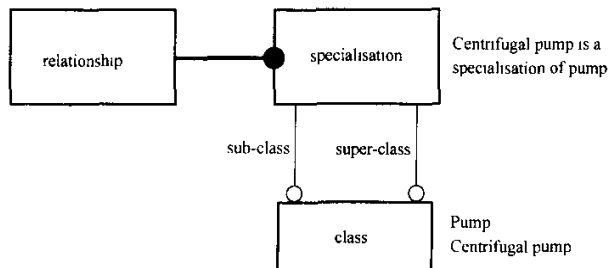


图 E. 13 分析两个类之间的关系

E. 3. 5 两个类之间的类关系

图 E. 14 表示了以不同方式出现的两个类之间的 association。Association 是 composition_according_to_class,用来说明整个类的一个成员有一个部分类的成员作为其一部分。给出的例子是一个离心泵有一个叶轮作为它的一部分。尽管 association 有起始和中止时间,但是在这里没有应用,但声明还是必要的。

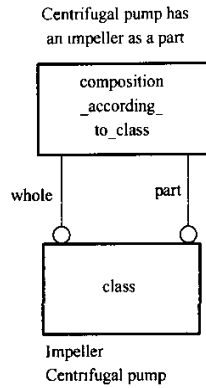


图 E. 14 关系类

该情况也没有空间-时间图,但是分析表明与前面的例子有所不同。Association 表达的也是一种规则。该情况下,整个类的每个成员都可以有分类的成员作为一个部分。因此每个离心泵都可以有一个叶轮作为它的一部分。而且该情况下特殊的叶轮和特殊的离心泵之间特殊的关系是该规则的一个事例。因此,这种 association 表达了一个 class_of_relationship。

在图 E. 15 中,增加了组合关系和分类关系用以表达规则如何应用于特殊事例。一个叶轮,3456 S1 被作为叶轮分类,一个泵,泵 1234 S2 被作为一个泵分类;一个组合关系 #1 显示叶轮 3456 S1 是泵 1234 S2 的一部分;最后,一个关系分类显示 #1 是一个叶轮作为一个泵的一部分的一种情况。

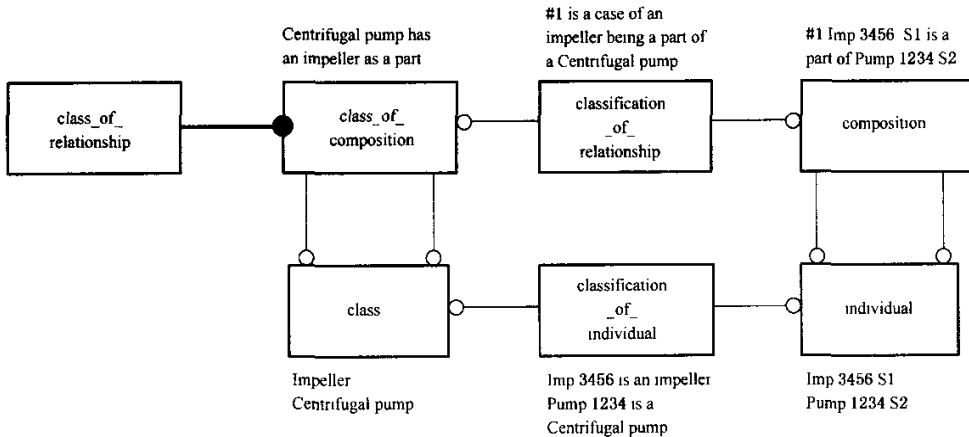


图 E. 15 分析关系类

E. 4 结论

相对于当前的域状态,在管理历史记录上 Association 已经成为一种重要的手段,然而,本文中的分析表明大多数的细节被 association 结构所隐藏。本文显示的 association 的空间-时间分析方法允许建立一种更加精确的模型使得隐藏的细节更加明朗。已经确定了五种方式来表达 Association 如何用空间-时间图表示。

参 考 文 献

- [1] GB/T 16656.21 工业自动化系统与集成 产品数据表达与交换 第21部分:实现方法:交换结构的纯正文编码.
- [2] ISO/TR 9007:1987 Information processing systems—Concepts and terminology for the conceptual schema and the information base.
- [3] ACZEL Peter. Non-Well-Founded Sets, Center for the Study of Language and Information, Stanford, California, 1988.
- [4] ITÔ K. Encyclopedic Dictionary of Mathematics, Mathematical Society of Japan, Edition 2, Cambridge, Massachusetts, MIT Press, 1993.
- [5] WEST Matthew, FOWLER Julian. Developing High Quality Data Models. Version 3.0. EPISTLE, 1996-08-27 [cited 2001-03-11]. Available from the World Wide Web: <<http://www.stepcom.ncl.ac.uk/epistle/data/mdlgdocs.htm>>.
- [6] WEST Matthew. Some Notes on the Nature of Things. ISO TC184/SC4/WG10 N307, 2000-06-09 [cited 2001-03-11]. Available from the World Wide Web: <http://www.nist.gov/sc4/wg_qc/wg10/current/n307/wg10n307.htm>.
-