

面向对象石油管理公共数据模型建立研究^{*}

胡长军^{1,2}, 仝兆岐¹, 童云海¹

¹(石油大学 计算机系, 山东 东营 257062);

²(北京大学 计算机科学与技术系, 北京 100871)

E-mail: huchangjun@ailab.pku.edu.cn

http://www.hdpu.edu.cn

摘要: 给出了一个石油管理领域应用的面向对象公共数据模型 PDMM (petroleum data model for management), 论述了元模型的建立与表达、类结构组织和实体定义、属性及约束规则等模型构筑技术, 强调了通过引入“活动”、“特性”等超类使模型具有强大描述能力的特色, 给出了模型的 EXPRESS 语言描述方法。

关键词: 数据模型; 面向对象; 领域数据工程; 数据库

中图法分类号: TP311 **文献标识码:** A

数据模型是对数据结构、数据之间联系和数据操作约束的描述, 建立领域公共数据模型是实现领域软件数据共享和领域软件集成平台的基础。石油计算机应用是数据密集型的应用, 石油勘探开发的生产管理数据类型复杂, 联系多样, 而且数据量极大, 虽然经过多年的努力, 有了一批用于勘探开发生产的软件系统, 但这些软件一般都带有自己的数据模型和数据管理系统, 软件之间的数据共享十分困难, 这就造成了在对同一区块的数据用不同的数据进行处理时, 需要反复进行数据加载和格式转换, 因而造成了人力和财力的浪费。在生产管理领域, 由于缺乏公共数据模型, 造成了信息交换的困难以及各种统计报表数据自下而上传递的麻烦, 也造成自上而下信息查询的不方便。有人统计, 全球每年由于数据不能共享而造成的数据管理上的浪费达数十亿美元。彻底解决这一问题的关键是, 构造领域公共数据模型作为数据描述的标准, 同时设计标准的数据存取规范和应用程序通信规范, 并在此基础上, 设计领域软件集成平台。石油管理公共数据模型 (petroleum data model for management, 简称 PDMM) 就是我们利用面向对象技术, 根据生产管理实际需求构筑的一个公共数据模型。目前它已和国际石油开放软件组织发布的石油公共数据模型 EPICENTRE^[1]一起, 成为国内石油软件集成平台的基础。

1 模型的构筑方法

PDMM 是一个面向对象数据模型, 由类、实体、属性、约束规则、标准参考值和各种数据类型等组成, 模型采用国际标准建模语言 EXPRESS (ISO 10303 11) 加以描述。EXPRESS 是具有面向对象特征的数据描述语言, 它定义的一些主要概念有: 实体、属性、函数、实例、模式等, 这些与通常的面向对象分析和设计方法中的定义区别不大^[2]。它使用的关键字也不多, 所以十分易用, 且易于理

收稿日期: 1999-10-10; **修改日期:** 2000-01-07

基金项目: 国家 863 高科技发展计划资助项目 (863-306-ZT01-02-3); 中国石油天然气集团公司“九五”重点攻关计划资助项目 (961101-10)

作者简介: 胡长军 (1963-), 男, 河北沧州人, 博士生, 副教授, 主要研究领域为数据工程及知识处理, 领域软件集成技术, 并行计算; 仝兆岐 (1940-), 男, 山西汾阳人, 教授, 博士生导师, 主要研究领域为计算机网络, 数据库系统; 童云海 (1972-), 男, 浙江宁波人, 讲师, 主要研究领域为空间数据库, 信息系统。

解.用 EXPRESS 构筑模型的方法是,首先构筑元模型,然后利用元模型构筑整个模型.

1.1 PDMM 的主要概念

PDMM 是一个面向对象的数据模型,它涉及的主要概念有:

(1) 实体(entity):具有实际意义的可识别的对象.与石油勘探开发生产管理有关的、我们感兴趣的一切东西都可用实体来定义.

(2) 类和子类(supertype,subtype):具有相关主题的实体集合.在我们的模型中首先抽象了与生产管理相关的超类,然后层层向下展开.超类是对生产管理过程的高度抽象,不同层次的子类是对超类行为的具体化.

(3) 属性:对实体的详细描述.在 PDMM 中有两类属性,一类是显式属性(explicit attribute),用来描述实体;另一类是逆属性(inverse attribute),用来建立实体之间的联系.

(4) 数据类型:包括基类型和自定义数据类型.模型根据生产管理实际,定义了许多数据类型,如曲线类型、周期类型等.这些类型是在 EXPRESS 的基本类型和聚集类型的基础上定义的.

(5) 约束规则:是对实体、属性以及联系的约束.模型中定义了两种类型约束,条件约束和惟一性约束,约束规则用 EXPRESS 表达式表示.从规则的约束对象来看,又可分为对一个实体内属性的约束和一个模式内所有实体之间联系的约束,前者叫局部约束,后者叫全局约束.

1.2 元模型定义

元模型是用来定义模型的模型,包括构筑模型的概念、规则、结构、数据类型等.元模型告诉我们如何去定义模型实体,怎样描述属性及约束规则等,所以元模型是模型的基础.它包括如下内容:

(1) 参考实体的定义:所谓参考实体是实体的实例值已被确定了的实体.生产实际中有许多实体的实例值是不经常变化的,例如,国家和油田等的名称、计量单位、一些技术标准参数等,这些实例值在建模时提前给定,可方便用户.

(2) 模型的表达:整个模型用 EXPRESS 的模式进行定义,模式中包含了 PDMM 的所有实体、属性和规则.模式用 EXPRESS 定义如下:

```
ENTITY schema_definition;
  name: STRING;
  types: SET OF named_type;
  global_rules: SET OF global_rule;
UNIQUE
  URL: name;
END-ENTITY;
```

这里,属性 name 为模式的名称,它在模式范围内取值应惟一,这里取值为模型名称“PDMM”,types 为模型定义的全体实体集合,global_rules 是模式中定义的全局规则集合.

(3) 命名数据类型:生产实践中常用到一些由基本数据类型组合的数据类型,如“周期”就是两个时间类型组成的时间段,命名数据类型就是为方便应用而定义的,是对自定义类型的命名.命名数据类型的作用域为定义它的模式.命名数据类型的 EXPRESS 定义如下:

```
ENTITY named_type
  ABSTRACT SUPERTYPE OF /* 命名类型是 entity_definition, defined_type 的超类 */
  (ONEOF (entity_definition, defined_type));
  name: STRING; /* 实体名称或自定义的类型名称 */
```

INVERSE

parent_schema; schema_definition FOR types; (* parent_schema 是 entity_definition 和 defined_type 所在的模式 *)

UNIQUE

URL; name, parent_schema;

END ENTITY;

(4) 自定义数据类型:是在 EXPRESS 数据类型的基础上定义的,主要是为了方便用户的应用,如针对勘探开发实际,定义了样品类型、空间数据类型、曲线类型等。数据类型的定义描述如下:

ENTITY defined_type

SUBTYPE OF (named_type, underlying_type);

domain; underlying_type;

where_rules; SET OF type_where_rule;

END ENTITY;

(5) 支撑类型 underlying_type 是自定义类型的超类,定义如下:

ENTITY underlying_type

ABSTRACT SUPERTYPE OF (ONEOF

simple_type, /* EXPRESS 的简单类型 */

aggregate_type, /* EXPRESS 的聚集类型,包括 list,set,bag,array */

PDMM_data_type, /* 模型中定义的扩充数据类型 */

defined_type, /* 自定义数据类型 */

enumeration_type)); /* 枚举类型 */

END ENTITY;

(6) 实体定义:描述了实体的子类、超类、属性、约束规则等。

PDMM 模型的实体定义是用 EXPRESS 的实体声明实现的,其形式如下:

ENTITY entity_definition

SUBTYPE OF (named_type, base_type, base_element_type);

supertypes; LIST OF UNIQUE entity_definition; /* 按在模式中声明的顺序列出该实体的超类 */

attributes; LIST OF UNIQUE attribute; /* 列出该实体的所有属性 */

where_rules; SET OF where_rule; /* 有关该实体的条件约束规则 */

uniqueness_rules; SET OF uniqueness_rule; /* 列出所有唯一性约束规则 */

instantiable; BOOLEAN; /* 是否可实例化标志,抽象实体不可实例化 */

independent; BOOLEAN; /* 该实体是否可独立实例化标志,如果它没有引用参考实体,是可以单独实例化的 */

supertype_rule; OPTIONAL STRING; /* 对实体超类的约束说明 */

END ENTITY

(7) 属性定义:描述属性名称、类型、性质等。模型中定义了两种类型的属性,一种是显式(explicit)属性,另一种是逆(inverse)属性,属性定义形式如下:

ENTITY attribute

ABSTRACT SUPERTYPE OF

(ONEOF (explicit_attribute, inverse_attribute)); /* 属性有两个子类,一个是显式属性,一个是逆属性 */

name; STRING; /* 属性名称 */

```

domain; base_type; /* 属性类型的值域 */
INVERSE
parent_entity; entity_definition FOR attributes; /* 逆属性指向的联系实体 */
UNIQUE
URI; name, parent_entity;
END_ENTITY;
ENTITY base_type /* 基类型定义(即属性取值域定义) */
ABSTRACT SUPERTYPE OF (ONEOF (simple_type,
aggregate_type,
PDMM data_type,
defined_type,
entity_definition));
END_ENTITY;

```

(8) 显式属性和逆属性定义:显式属性是实体固有的不能从其他属性导出的属性,是对实体的刻画;逆属性用来表示实体间联系,它表明逆属性所在实体被逆属性所指的实体所操纵,其定义如下:

```

ENTITY inverse_attribute
SUBTYPE OF (attribute);
SELF attribute.domain; entity_definition; /* 定义属性值域 */
inverted_attr; explicit_attribute; /* 定义与该实体相关的实体中指向本实体的属性 */

min_cardinality; INTEGER; /* 引用该实体实例的原实体实例的最小个数 */
max_cardinality; INTEGER; /* 引用该实体实例的原实体实例最大个数 */
duplicates; BOOLEAN; /* 标识该实体实例是否被同一实例多次引用 */
WHERE
role_invertible; inverted_attr IN domain.attributes;
uniqueness_correct; (max_cardinality>1) OR NOT duplicates; /* 只有属性值为多值时 duplicates 值才
有可能取真值 */
END_ENTITY;

```

(9) 惟一性约束规则:定义能够惟一标识实体实例的属性或属性的组合.

```

ENTITY uniqueness_rule;
label; OPTIONAL STRING; /* 惟一性准则标号 */
attributes; LIST [1;?] OF attribute; /* 规则涉及的所有属性 */
secondary_identifier; BOOLEAN; /* 当规则作为实体实例的第2标识符时取为“真” */

INVERSE
parent_entity; entity_definition FOR uniqueness_rules; /* 定义该规则的实体 */
UNIQUE
URI; label, parent_entity; /* 标号在定义的实体内应惟一 */
END_ENTITY;

```

(10) 条件约束规则:是对实体的每一个实例属性间关系的约束.

```

ENTITY where_rule;

```

```

label: OPTIONAL STRING;      /* 条件约束规则名称 */
rule-text: STRING;          /* 规则内容,用 EXPRESS 表达式表示 */
INVERSE
parent-entity: entity-definition FOR where-rules; /* 定义该规则的实体名 */
UNIQUE
URL: label, parent-entity;   /* 标号在定义的实体范围内惟一 */
END-ENTITY;

```

(11) 类型约束规则:用来实现对自定义数据类型的约束.

```

ENTITY type-where-rule;
label: OPTIONAL STRING;     /* 约束规则名 */
rule-text: STRING;         /* 规则内容: EXPRESS 表达式 */
INVERSE
parent-type: defined-type FOR where-rules; /* 定义规则的类型实体 */
UNIQUE
URL: label, parent-type;
END-ENTITY;

```

(12) 全局规则:定义对一个实体的多个实例或多个实体之间的约束规则.

```

ENTITY global-rule;
label: STRING;             /* 规则名称 */
rule-text: STRING;        /* 规则内容 */
entities: SET[1:?] OF entity-definition; /* 受本规则约束的实体 */
INVERSE
parent-schema: schema-definition FOR global-rules; /* 定义该规则的模式 */
UNIQUE
URL: label, parent-schema; /* 规则名在定义的模式内惟一 */
END-ENTITY;

```

2 PDMM 的主要内容

PDMM 主要由类、实体、属性、约束规则等组成.我们首先对石油生产管理活动进行高度的抽象,得到6个超类,然后对每一个超类进行进一步的具体化,层层向下分解,直到定义出每一个实体为止.

该模型目前定义了7个超类,包括勘探开发管理数据类(M-EP-DATA)、管理活动类(M-ACTIVITY)、管理数据收集类(M-DATA-COLLECTION)、管理参考数据类(M-REF-DATA)、管理组织类(M-ORGANAZATION)、管理对象特性类(M-PROPERTY)和实体联系超类(M-ASSOCIATION).

(1) M-EP-DATA 超类:定义与勘探开发管理有关的一切实体,它将多年来不同部门为了管理需要设计的一些关系数据模型规范实体包括进来,形成了完整的生产管理实体集合.它的严格定义如下:

```

ENTITY M-EP-DATA
ABSTRACT SUPERTYPE (ONE OF
(BUSINESS-OBJECT,      /* 事务对象,包括合同、契约、制度等 */

```

```

EXPLORATION_OBJECT, /* 勘探管理对象,包含多年积累的关系模型实体 */
PRODUCTION_OBJECT, /* 开发管理对象 */
DRILLING_OBJECT, /* 钻井管理对象 */
REFINERY_OBJECT)); /* 炼化管理对象 */
SUBTYPE OF(PDMM-V1.0);

```

END_ENTITY

(2) M_ACTIVITY 超类:活动超类包括了管理过程中的一切行为的抽象,活动的属性包括活动性质、对象、时间等.它的定义如下:

```

ENTITY M_ACTIVITY

```

```

  ABSTRACT SUPERTYPE (ONE OF

```

```

    (DAILY_REPORT, MONTH_REPORT, YEAR_REPORT, WELL_LOCATION_DISCUSS, PRODUCTION_PLANNING, EXPLORATION_WELL_EVALUATION...));

```

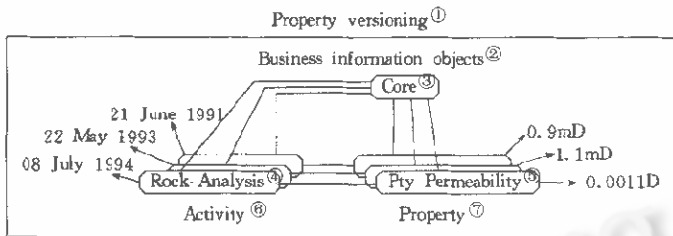
```

  SUBTYPE OF(PDMM-V1.0);

```

END_ENTITY

活动的引入是模型的特色之一,在石油勘探开发过程中,对同一对象进行反复认识得到不同的结果是常有的事,引入活动可以很好地描述这一情况.如对于一块岩芯,在不同时间用不同的仪器测同一特性的不同值,这些值对不同应用具有不同的意义.如果用关系方法描述这一问题较为麻烦,但若用我们的模型来描述,就十分自然,如图1所示.



①特性的版本,②事务信息对象,③岩芯,④岩性分析,⑤渗透率,⑥活动,⑦特性.

Fig. 1 Object, property and activity
图1 对象、特性和活动

(3) M_DATA_COLLECTION 超类:是对一个项目或一个事件相关数据的收集,它由活动而创建,在物理实现时,可考虑将一个收集的数据集中存放,以提高数据存取效率.

(4) M_REF_DATA 超类:是对实际中常用而不经常变化的实例值的抽象,如单位名称、国家名称、计量单位等等.模型允许各单位根据自己的实际定义参考值.但对每一个自定义的值,要按规定给全属性.定义参考属性不仅可以使属性的组织更为合理,而且数据的一致性也能得到保证,特别是将一些公共属性组织成专门的类,可以充分利用面向对象的继承机制,大大提高数据管理的质量.这是模型的第2个特色.

(5) M_ORGANAZATION 超类:由各管理单位组成,它按实际存在的管理层次定义.

(6) M_PROPERTY 超类:是对实体特性的抽象.将对象特性与对象及对象联系本身分离,是PDMM模型的第3个特色,可有效地解决同一对象特征的不同版本问题.在勘探开发过程中,对同一数据不同领域专家有不同的解释,得到同一对象特征的不同版本是常有的事,模型的这一特色可有效地描述这一现象.一组特性值只能描述一个对象或对象之间关系的实例,如果有多个实例值需要描述,就需要引入活动类的概念,因为不同版本的特性值是由不同的活动得到的.

(7) ASSOCIATION 超类:是对实体之间联系的刻画,它可以描述领导关系、成份关系、组成关系、集合关系、分类关系、后继关系、连接关系、约束关系这8种类型的关系,不仅如此,它还能描述实体之间的动态关系,为此,PDMM 专门定义了子类 Transient-association. Association 类的定义弥补了关系模型的对实体联系刻画不细的缺陷,是模型的第4大特色.

这些超类是对勘探开发管理活动的高度抽象.在这些超类下,我们进一步定义了不同层次的实体.在模型中一个实体可以属于多个超类,也允许多重继承.在整个模型中,一种信息只被说明一次,没有重复说明的信息,这样就保证了数据的一致性.另外,PDMM 是一个逻辑数据模型,它不依赖于任何具体计算机环境和数据库管理系统,通过投影工具可以把它投影到目前流行的任何数据库系统中,所以它具有良好的移植性.

3 PDMM 的应用

PDMM 已在我们设计的中国石油天然气集团公司生产经营数据仓库系统中获得应用.利用模型描述了一个原油和成品油的生产和经营活动,包括计划、调度、配产、营销等,并为它设计了一个投影工具,将用 PDMM 描述的领域数据投影到 ORACLE 数据库系统中存储.设计实现了基于模型的生产经营信息系统.应用实践表明:(1) 无论从数据描述能力还是数据的存取效率来看,该模型作为勘探开发管理数据平台的基础都是合适的;(2) 领域数据模型的标准化对领域的信息系统建设具有重要意义,只有领域数据模型的标准化,才有领域生产经营决策的信息化;(3) 任何公共数据模型都不可能涵盖所有的应用需求,所以允许用户根据自己的应用需求扩充和剪裁模型是十分必要的,前提是制定好扩充和剪裁的规范,否则模型标准就失去意义,这也正是我们今后的工作内容.

References:

- [1] <http://www.posc.org>.
- [2] Shao, Wei-zhong, Yang, Fu-qing. Object-Oriented Analysis. Beijing: Tsinghua University Press, 1998 (in Chinese).

附中文参考文献:

- [2] 邵维中,杨芙清.面向对象的分析.北京:清华大学出版社,1998.

Research on Constructing of Object-Oriented Petroleum Common Data Model^{*}

HU Chang-jun^{1,2}, TONG Zhao-qi¹, TONG Yun-hai²

¹(Department of Computer Science, University of Petroleum, Dongying 257062, China);

²(Department of Computer Science and Technology, Beijing University, Beijing 100871, China)

E-mail: huchangjun@ailab.pku.edu.cn

<http://www.hdpu.edu.cn>

Abstract: In this paper, an object-oriented common data model PDMM (petroleum data model for management) is presented, which is constructed for petroleum resource management domain. PDMM is characterizedly its strong capacities of describing data with its definition of superclasses "activity" and "property". The constructing techniques in PDMM include meta-model construction and its expressions, class organization, entity definition, as well as attributes and constraints. The representation of the model in EXPRESS language is also

discussed.

Key words: data model; object-oriented; domain data engineering; database

* Received October 10, 1999; accepted January 7, 2000

Supported by the National High Technology Development Program of China under Grant No. 863-306-ZT01-02-3; the Key Sci-Tech Project of the 'Ninth Five-Year-Plan' of the CNPC under Grant No. 961101-10