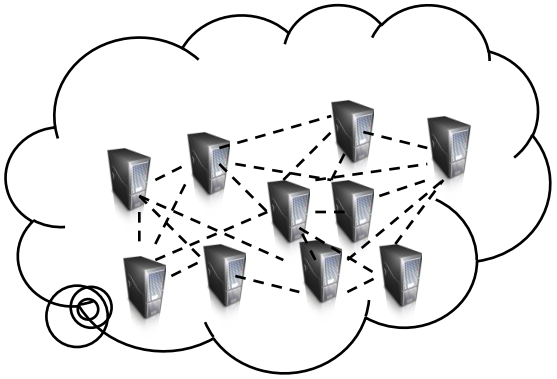


元数据管理与CWM标准

林子雨

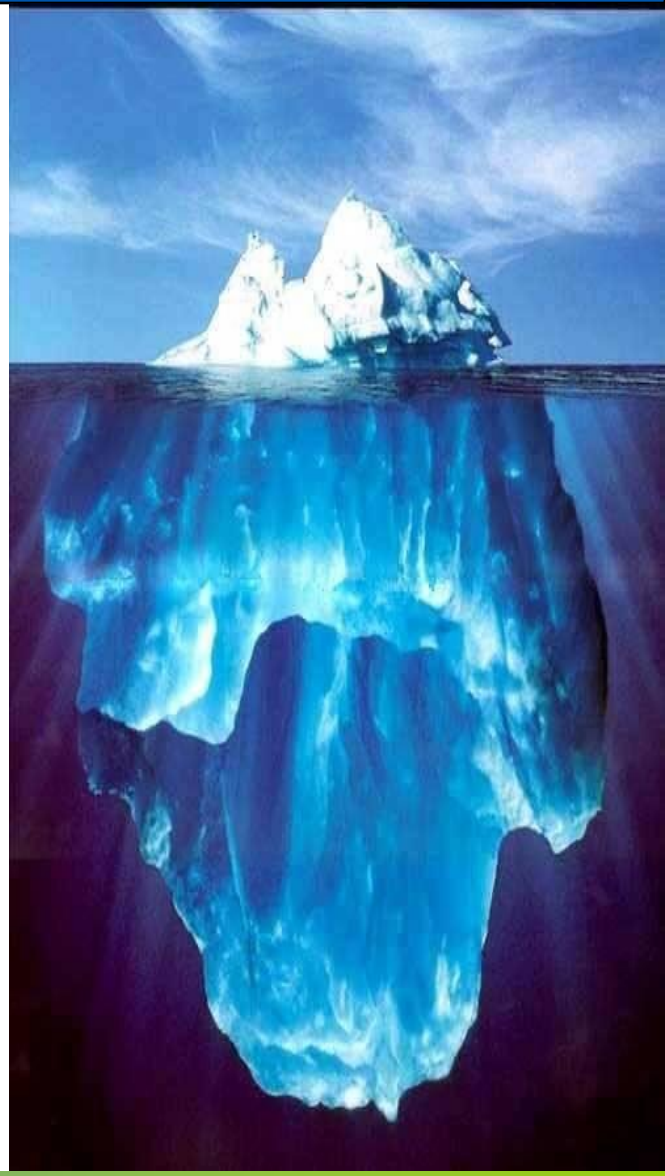
厦门大学计算机科学系

个人主页：http://www.cs.xmu.edu.cn/linziyu

E-mail: ziyulin@xmu.edu.cn

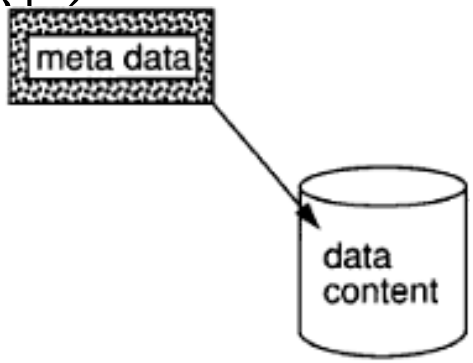
提纲

- 元数据管理基本概念
  - 元数据定义
  - 元数据管理
- CWM元数据标准
  - CWM标准概述
  - 对象模型层
  - 基础层
  - 资源层
  - 分析层
  - 管理层

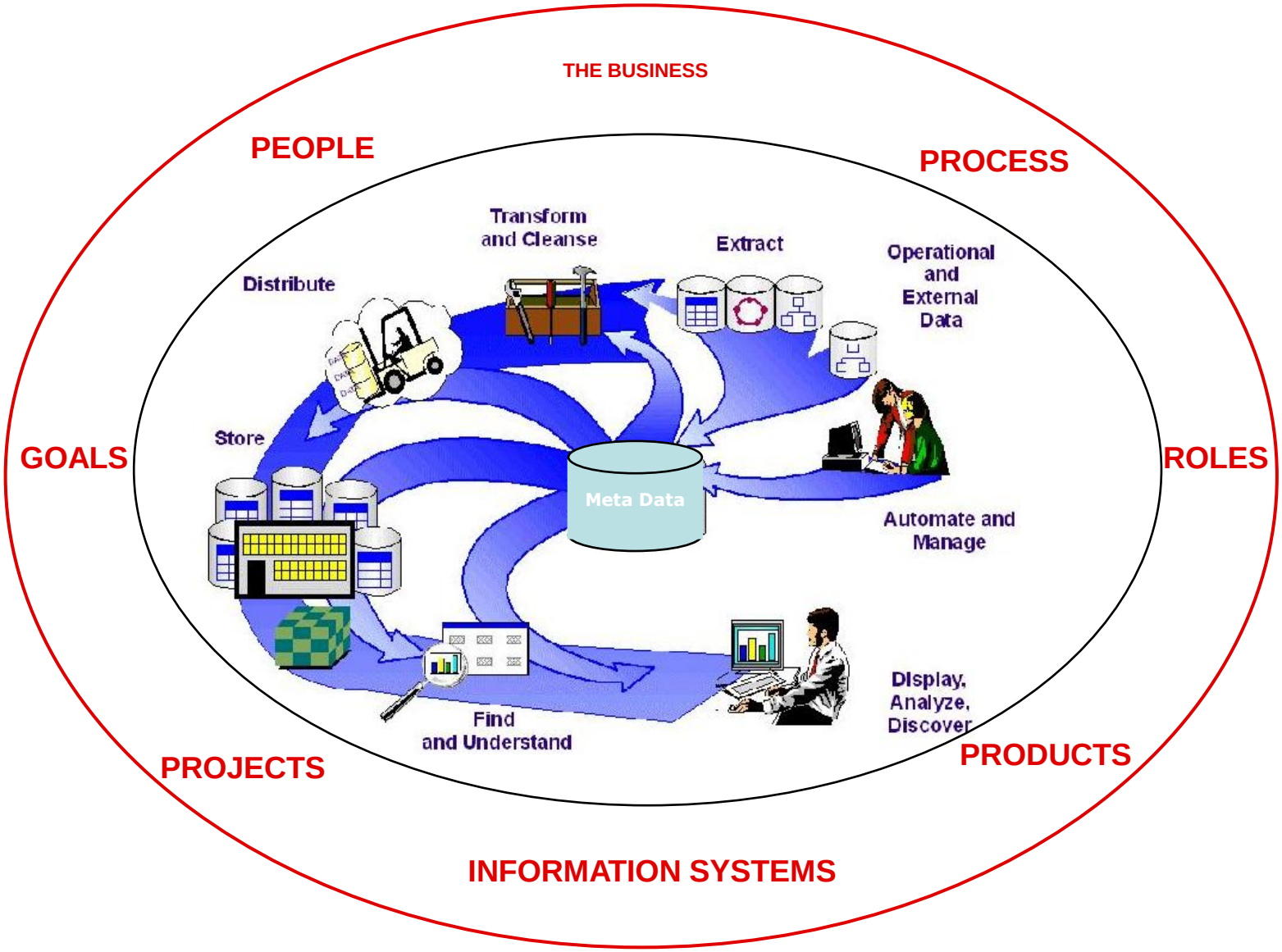


元数据定义

- “关于数据的数据”
- 比一般意义的数据范畴更加广泛
  - 不仅表示数据的类型、名称、值等信息
  - 提供数据的上下文描述信息（比如数据的所属区域、取值范围、数据间的关系、业务规则、数据来源等等）



元数据定义


THE BUSINESS
PEOPLE
PROCESS
GOALS
ROLES
PROJECTS
PRODUCTS
INFORMATION SYSTEMS
Transform and Cleanse
Extract
Operational and External Data
Distribute
Store
Meta Data
Automate and Manage
Find and Understand
Display, Analyze, Discover

分析系统关心的元数据

- 业务元数据
  - 业务名称、定义、描述和别名来表示数据仓库和业务系统中的各种属性，直接供业务分析人员使用
  - 业务元数据使分析系统使用人员能够更好理解、使用数据仓库，成为分析系统使用人员在数据仓库中的业务向导

分析系统关心的元数据

- 技术元数据包含关于分析系统数据技术层面的信息
  - 数据源元数据
  - ETL元数据
  - 数据仓库元数据
  - 数据集市元数据
  - OLAP SERVER元数据
  - 前端展现元数据
  - 其它类型元数据（挖掘模型，数据质量分析结果等）

分析系统关心的元数据

- 管理元数据主要是指日常建设过程中，涉及开发、运维等管理流程的基本信息。

提纲

- 元数据管理基本概念
  - 元数据定义
  - 元数据管理
- CWM元数据标准
  - CWM标准概述
  - 对象模型层
  - 基础层
  - 资源层
  - 分析层
  - 管理层

元数据管理

- 管理商业智能系统的元数据
- 贯穿商业智能系统的各个环节
- 系统的各个处理单元由元数据驱动

管理元数据的意义

- 有哪些数据？
- 它们在哪里用？
- 它的业务定义是什么？
- 这个数据还叫什么？
- 它与其他数据有什么关系？
- 谁用这个数据？
- 为什么我们要用它？
- 最近修改是什么时候？
- 这些数据准确、可靠吗？

管理元数据的意义

- 理解企业内部的信息资源
- 动态的数据字典
- 数据的浏览和归纳
- 数据在企业内部横向与纵向传递
- 保持整个企业的标准（保证企业内部统一的商业定义和商业规则）
- 数据生命周期的管理

元数据管理的几个概念

- 元模型（**meta model**）
- 元数据库（**metadata repository**）
- 元数据管理工具

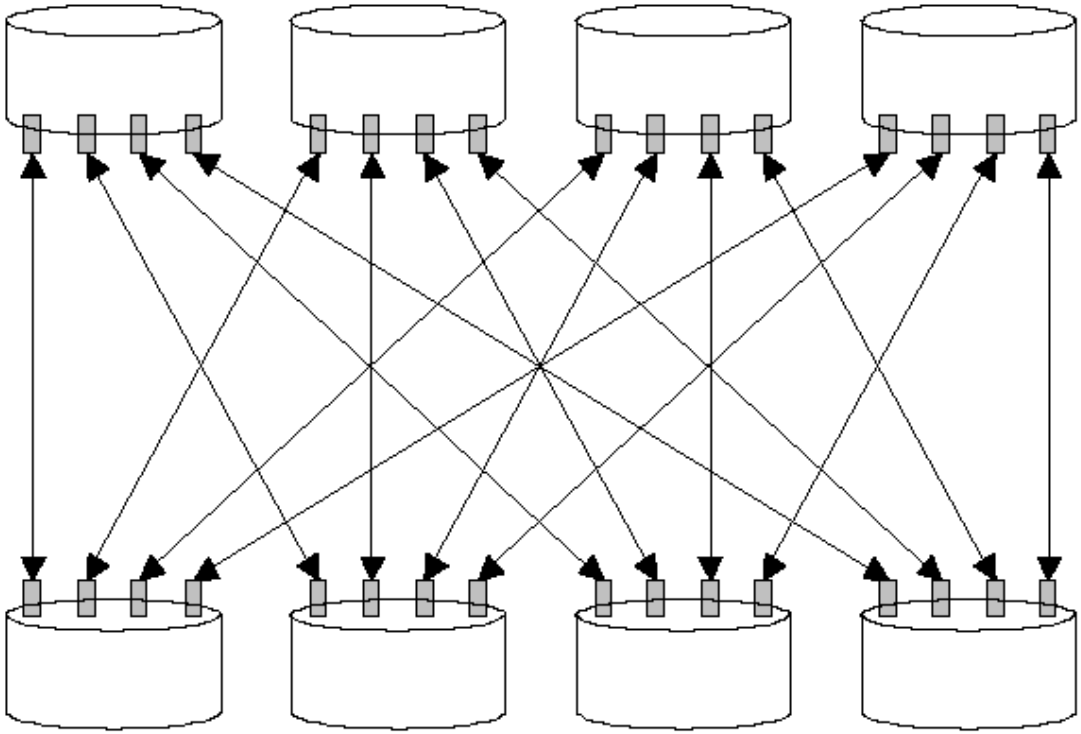
元模型

- 关于元数据的“元数据”
  - OIM模型
    - CA和微软的元数据标准
    - 利用PCAF文件交换
    - OIM组织已经解散
  - CWM模型
    - OMG组织制定的标准
    - 得到IBM，NCR，SAS，Hyperion等公司支持
    - 利用XMI文件进行交换

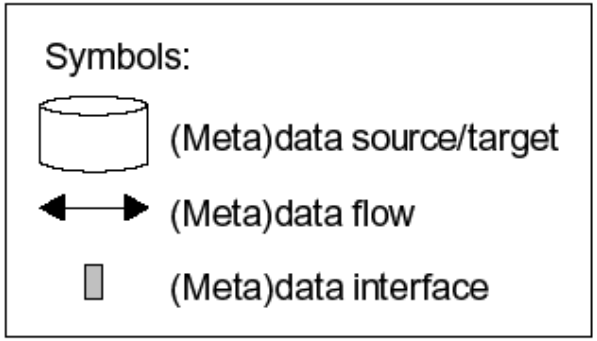
元数据库

- 元数据库就是一个逻辑上的统一存储元数据的地点
- 元数据存储常见的形式
  - 分散存储
  - 统一存储，提供不同接口
  - 统一存储，统一接口

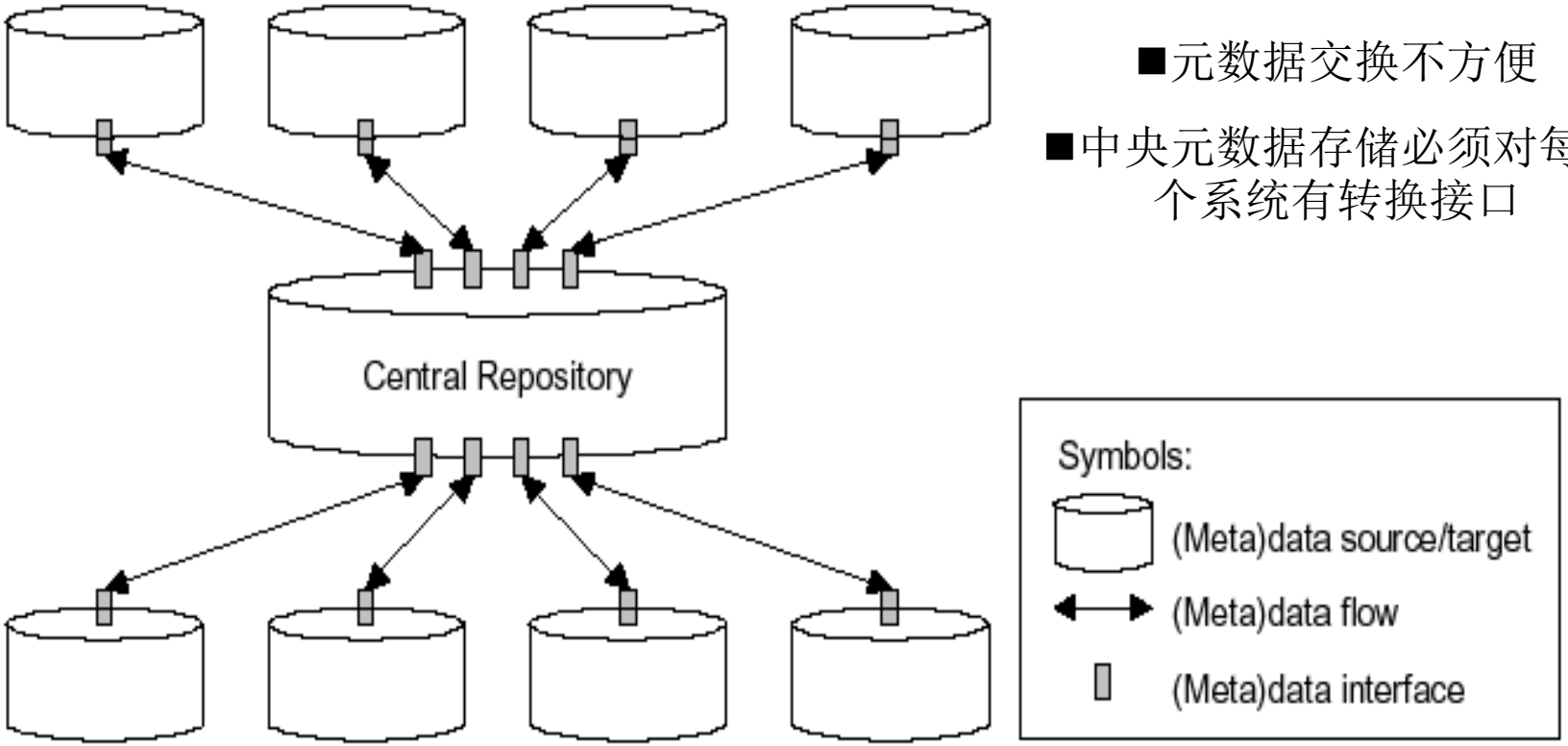
不同系统各自提供元数据接口



- 实现复杂
- 元数据不统一
- 易成为”蜘蛛网”

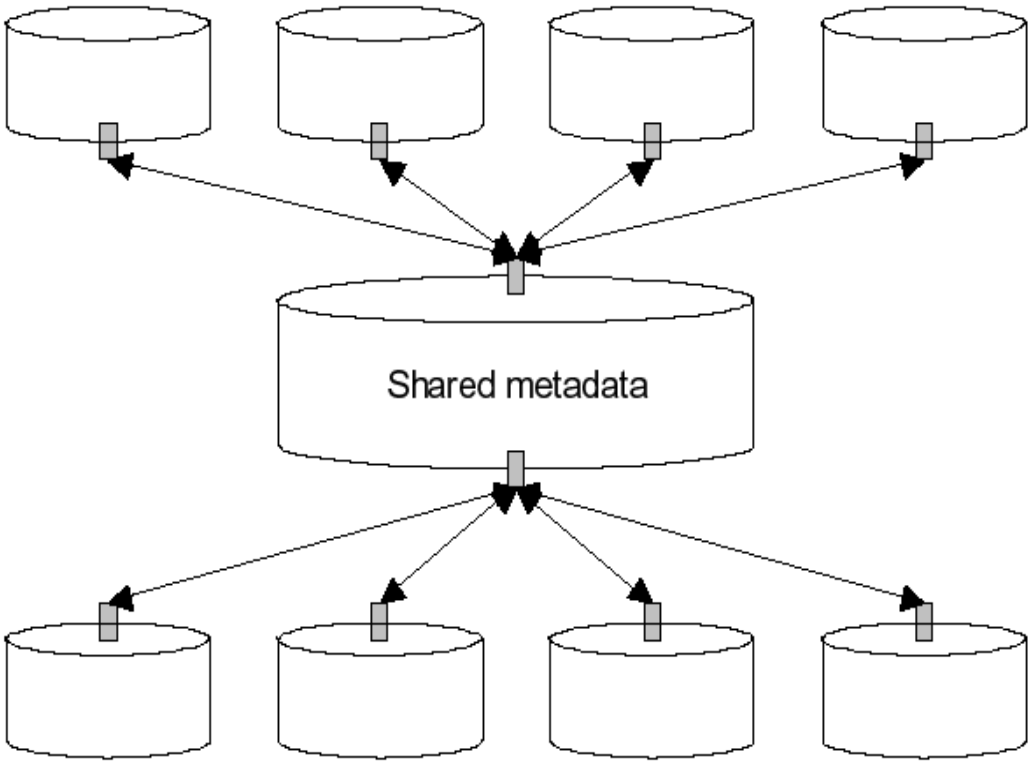


中央元数据存储

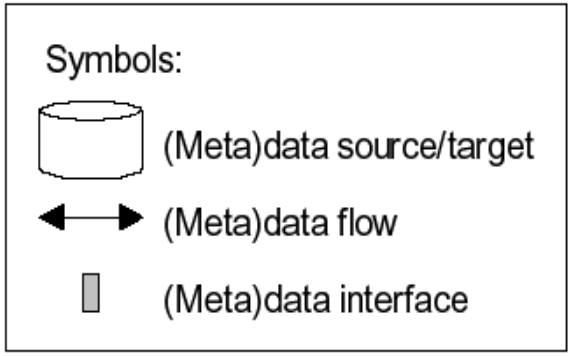


- 所有存取必须通过中央存储
- 元数据交换不方便
- 中央元数据存储必须对每一个系统有转换接口

基于标准的中央元数据管理



- 有利于元数据的交换
- 屏蔽系统内部变化
- 中央元数据只需要统一接口



元数据管理工具

- 元数据浏览、展示和管理的平台
- 知名的元数据管理工具包括：
  - Meta Center
  - Meta Matrix
  - Meta Integration
  - DB2, Teradata,Oracle等数据仓库中的元数据管理模块
  - …

提纲

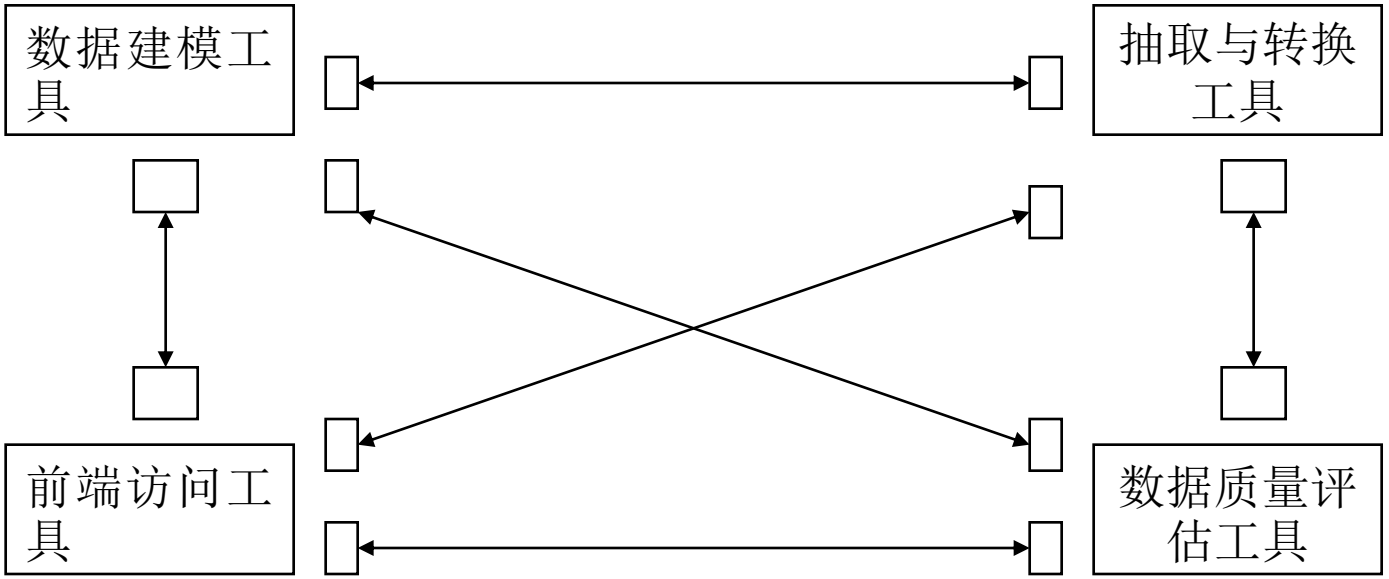
- 元数据管理基本概念
  - 元数据定义
  - 元数据管理
- CWM元数据标准
  - CWM标准概述
  - 对象模型层
  - 基础层
  - 资源层
  - 分析层
  - 管理层

CWM标准背景

- OMG是一个拥有500多会员的国际标准化组织，著名的CORBA标准即出自该组织。
- 公共仓库元模型（Common Warehouse Metamodel）的主要目的是在异构环境下，帮助不同的数据仓库工具、平台和元数据知识库进行元数据交换。

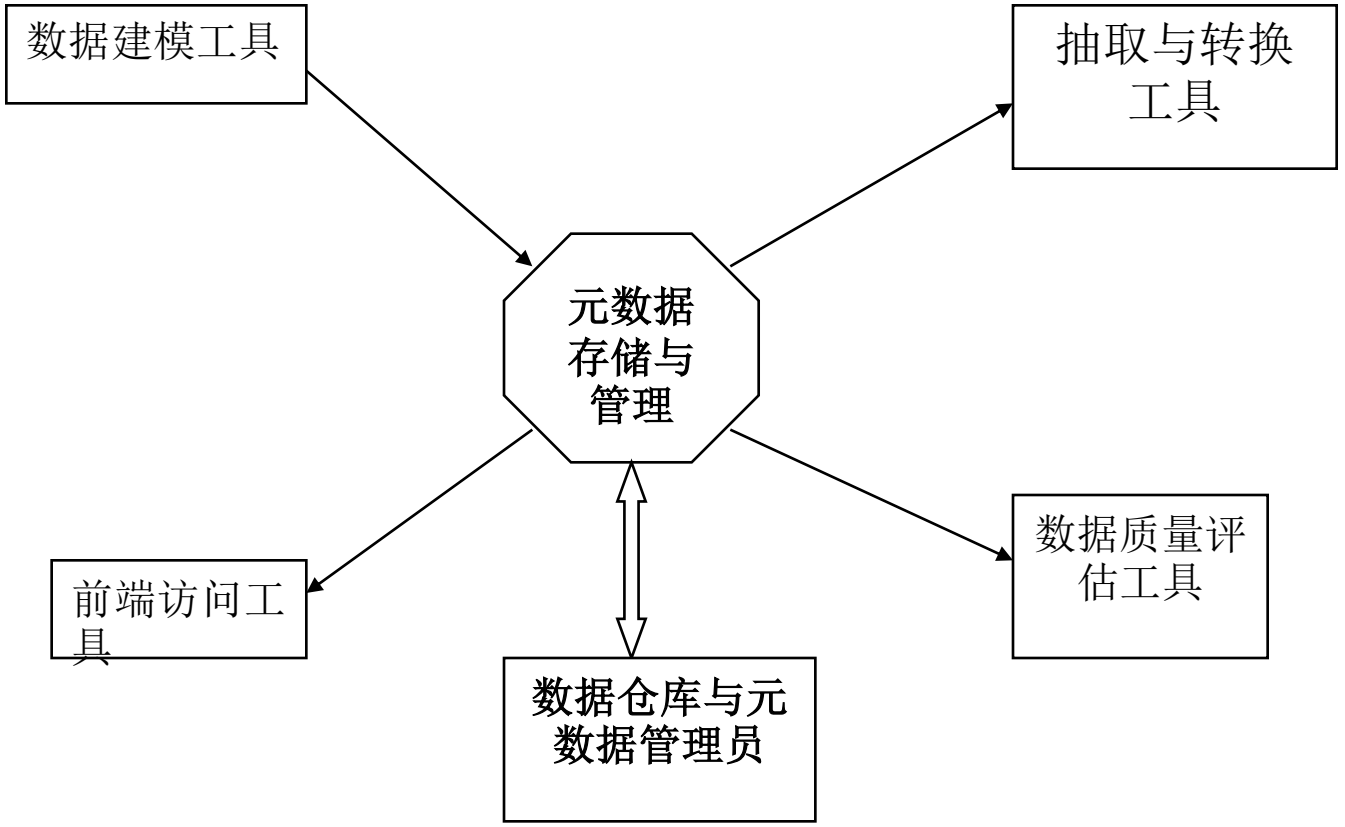
CWM标准的意义

- 在形成标准以前，要进行集成的情况如下图所示：



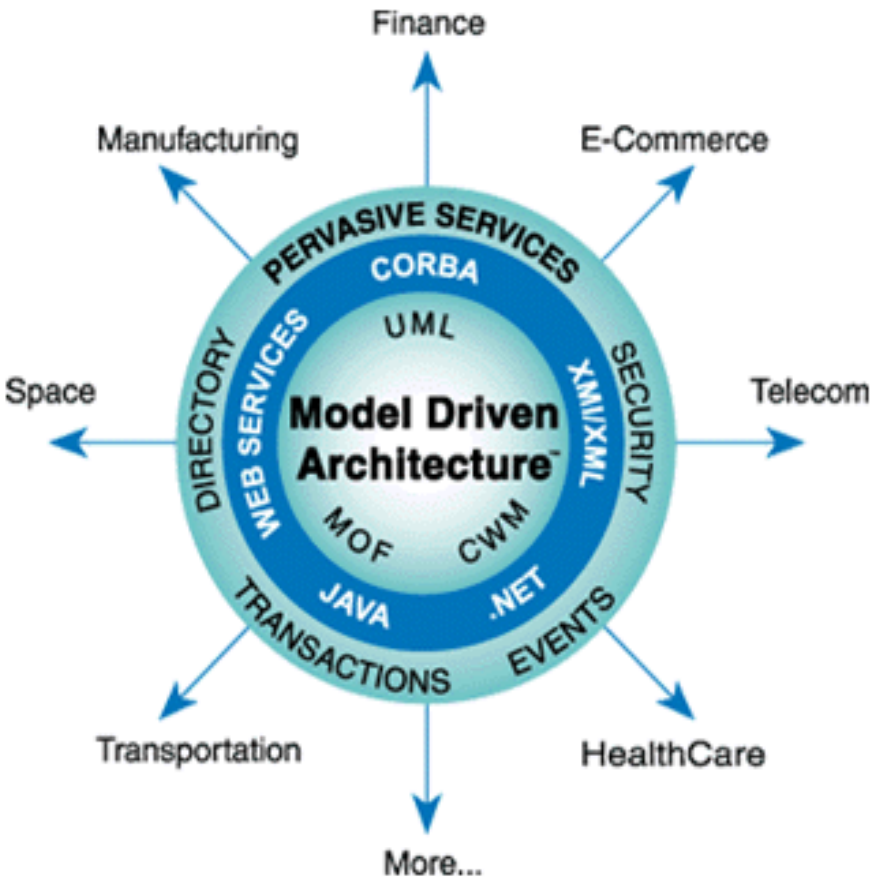
CWM标准的意义

- 在形成标准以后的情况如下图所示：



CWM的发展状况

- 成为OMG提出的基于模型驱动的体系结构（MDA）的核心之一（其它是MOF和UML）



CWM标准概述

- CWM标准是基于以下工业标准制定的：
  - UML：它对CWM模型进行建模。
  - MOF（元对象设施）：它是OMG元模型和元数据的存储标准，提供在异构环境下对元数据知识库的访问接口。
  - XMI（XML元数据交换）：它可以使元数据以XML文件流的方式进行交换。
  - CORBA IDL(CORBA 接口定义语言)

CWM元数据模型

Meta-level	MOF terms	Examples
M3	meta-metamodel	The "MOF Model"
M2	metamodel, meta-metadata	UML Metamodel, CWM Metamodel
M1	model, metadata	UML models, CWM metadata
M0	object, data	Modeled systems, Warehouse data

CWM的发展状况

- 绝大多数数据仓库和元数据管理工具已经支持CWM，或已经宣布在下一版本的产品中支持CWM。
- 已经被JAVA标准化组织着手扩展到J2EE体系结构当中，形成JMI（JAVA Metadata Interchange）规范、用于OLAP分析的JOLAP规范和用于数据挖掘的JDMAPI规范。

CWM的合作伙伴

- IBM
- Unisys
- NCR
- Hyperion
- Oracle
- UBS
- Genesis
- Dimension EDI

CWM的支持者

- Deere
- SUN
- HP
- Data Access
- Inline
- Aonix
- Hitachi
- SAS
- Meta Integration
- Adaptive

ETL产品

产品名称	是否支持CWM	元数据互换其他形式
IBM DB2 Warehouse Manager	支持Metadata Interchange Specification (MDIS).	通过API输入/输出
Oracle Warehouse Builder	是	
Sagent	不能确定	通过API输入/输出
Informatica PowerCenter	是	
Cognos Decision Stream	即将支持	通过API输入/输出
TeraData ETL组件	是	

OLAP产品

产品名称	是否支持CWM	元数据互换其他形式
Essbase/DB2 OLAP Server	支持Metadata Interchange Specification (MDIS).	通过API输入/输出
Cognos	即将支持	通过API输入/输出
Oracle 9i OLAP	是	

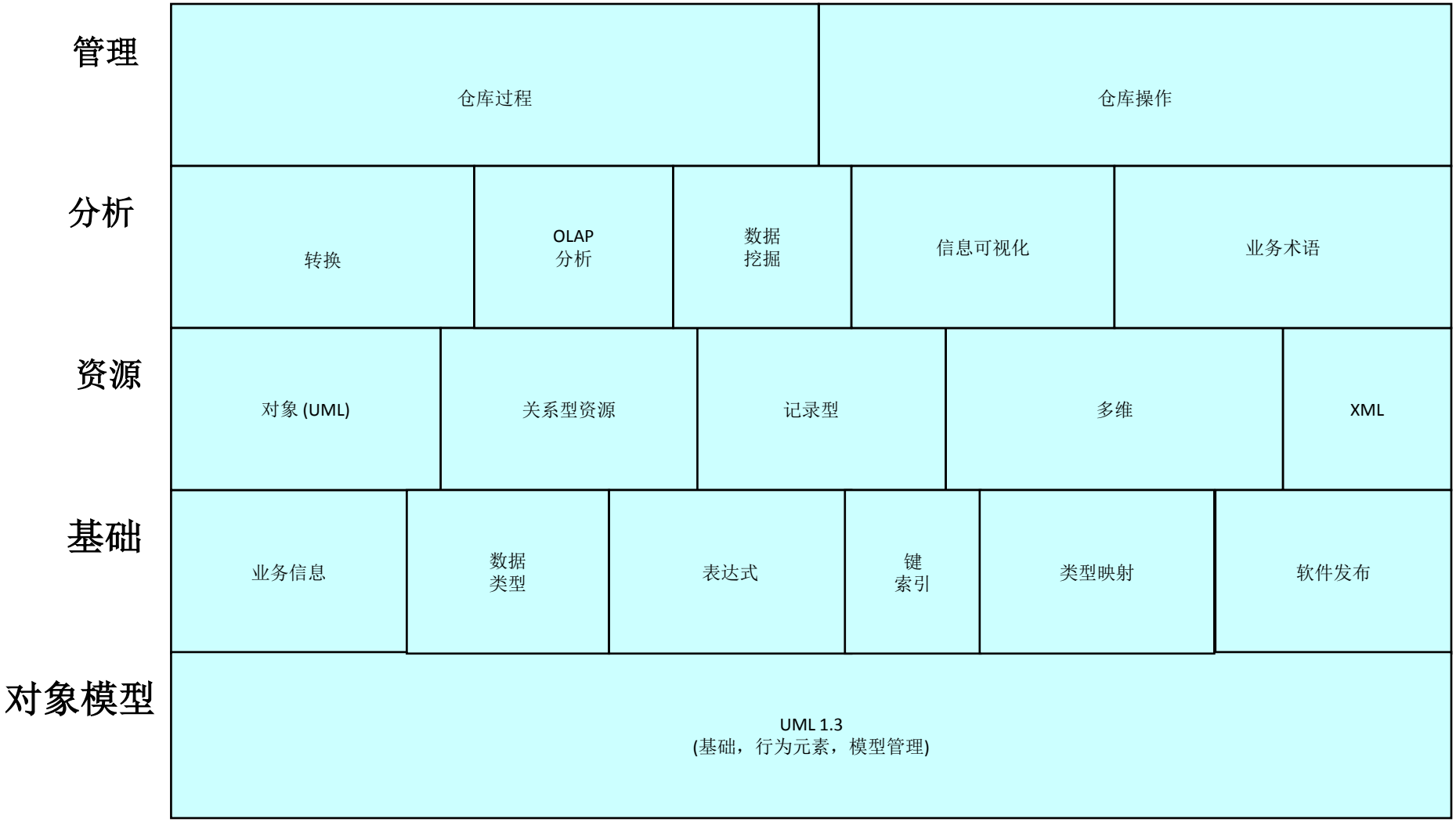
数据仓库元数据管理产品

产品名称	是否支持CWM	元数据互换其他形式
IBM DB2 Information Catalog	支持Metadata Interchange Specification (MDIS).	通过API输入/输出
Warehouse Control Center	是	
CA PLATINUM Repository	是	通过API输入/输出
TeraData Meta Data Services	是	通过API输入/输出
Oracle Warehouse Builder Repository	是	

提纲

- 元数据管理基本概念
  - 元数据定义
  - 元数据管理
- CWM元数据标准
  - CWM标准概述
  - 对象模型层
  - 基础层
  - 资源层
  - 分析层
  - 管理层

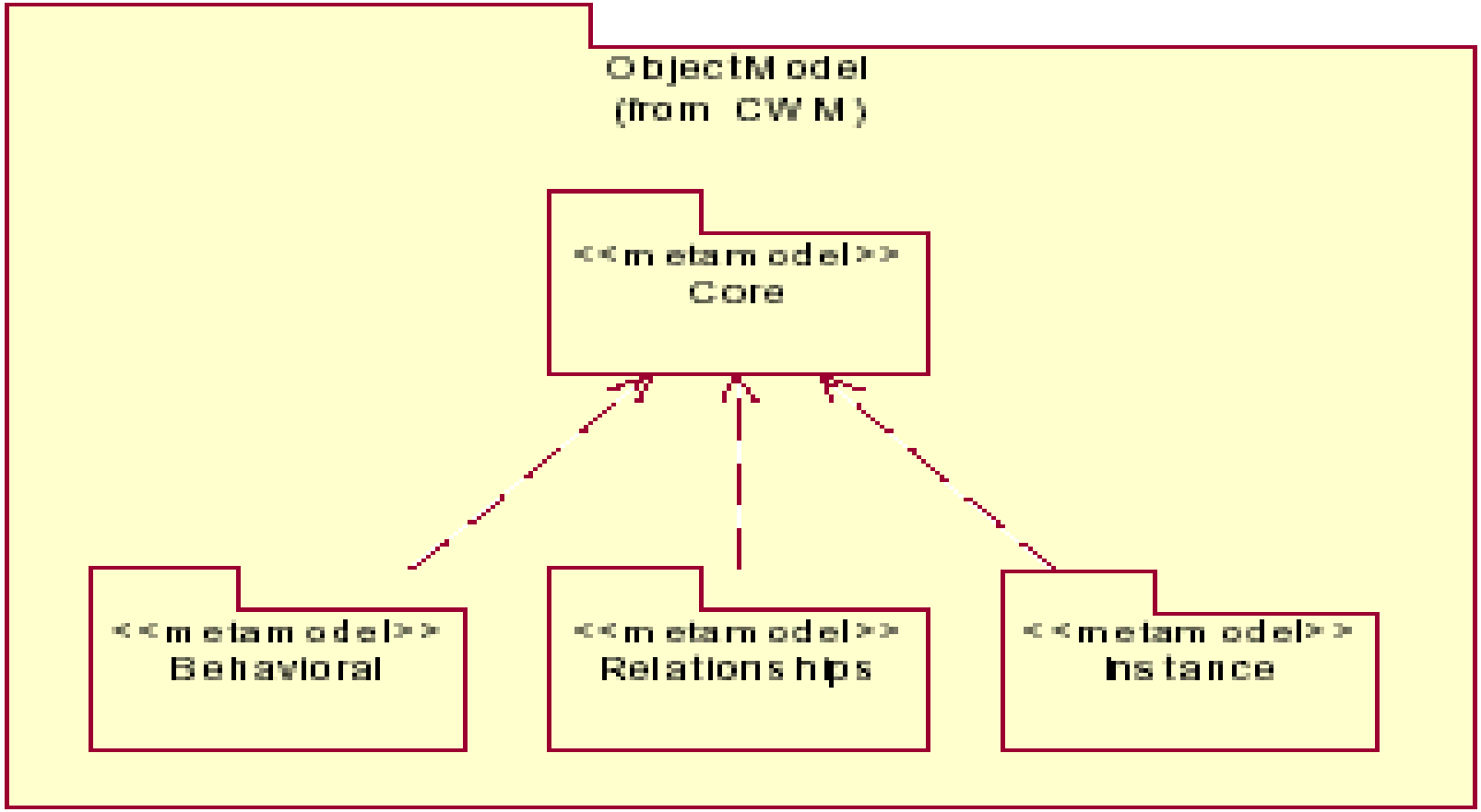
CWM标准包及其分层


管理
仓库过程
仓库操作
分析
转换
OLAP
分析
数据
挖掘
信息可视化
业务术语
资源
对象 (UML)
关系型资源
记录型
多维
XML
基础
业务信息
数据
类型
表达式
键
索引
类型映射
软件发布
对象模型
UML 1.3
(基础，行为元素，模型管理)

对象模型层（Object Model）

- CWM对象模型提供了描述其他所有包中元数据模型的类的基本结构和相应的类型属性
- 定义基本元模型的概念，关系和约束
- 包括4个基本包：
  - 核心包(Core)
  - 行为包(Behavioral)
  - 关系包(Relationship)
  - 实例包(Instance)

对象模型层


ObjectModel
(from CWM)
«metamodel»
Core
«metamodel»
Behavioral
«metamodel»
Relationships
«metamodel»
Instance

核心包(Core)

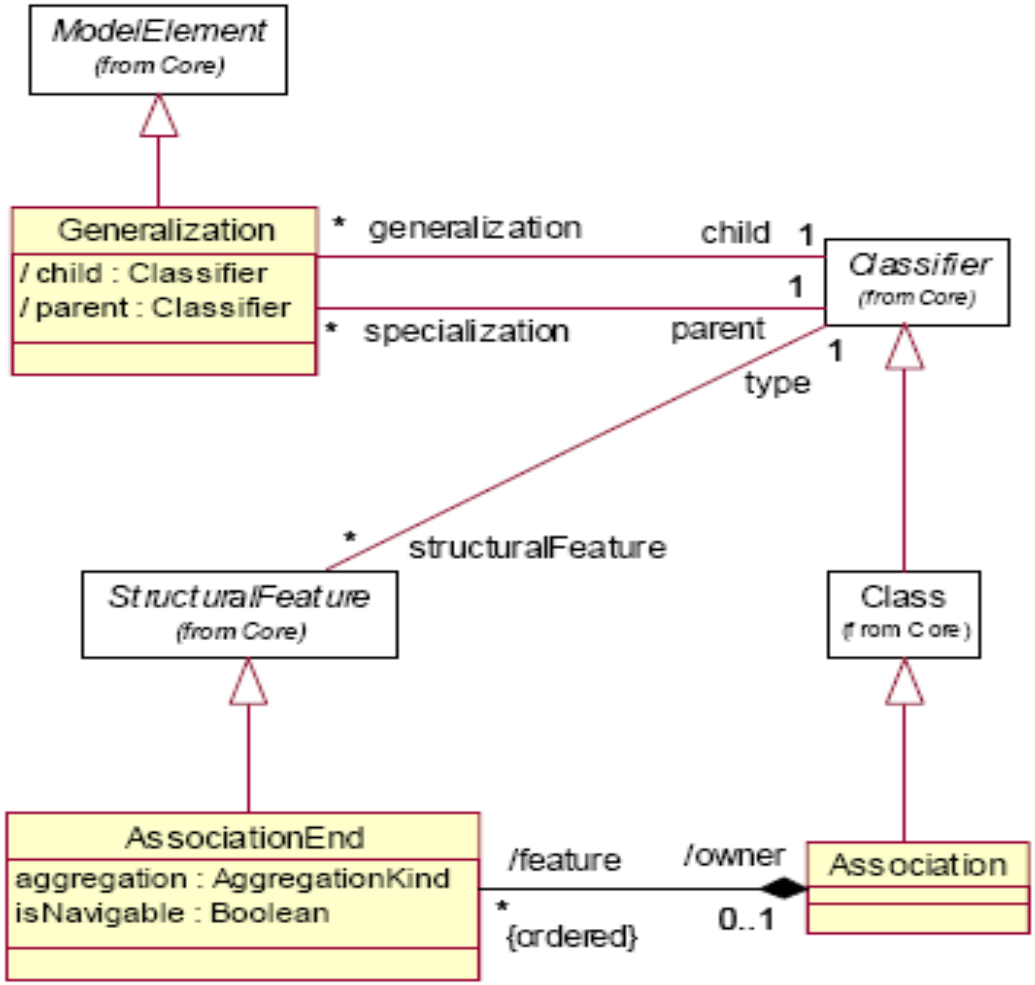
- 包含所有的其他CWM包使用的基本类和关联
- 不依赖于其他任何包

行为包(Behavioral)

- 描述其他CWM包中类的行为特征，提供一个记录特定行为请求的基础
- 包括操作，方法，接口，事件等

关系包（Relationship）

- 描述CWM对象之间如何互相联系
- 定义了两种类型的关系
  - 泛化(Generalization)
  - 关联(Association)
- 泛化是具有普遍性的对象和特定对象的关联，层次化的结构
- 关联定义两个或多个类元之间的特定关系


ModelElement
(from Core)
Generalization
/ child : Classifier
/ parent : Classifier
* generalization
child 1
1
* specialization
parent
1
type
Classifier
(from Core)
*
structuralFeature
StructuralFeature
(from Core)
Class
(from Core)
AssociationEnd
aggregation : AggregationKind
isNavigable : Boolean
/feature
*
{ordered}
/owner
0..1
Association

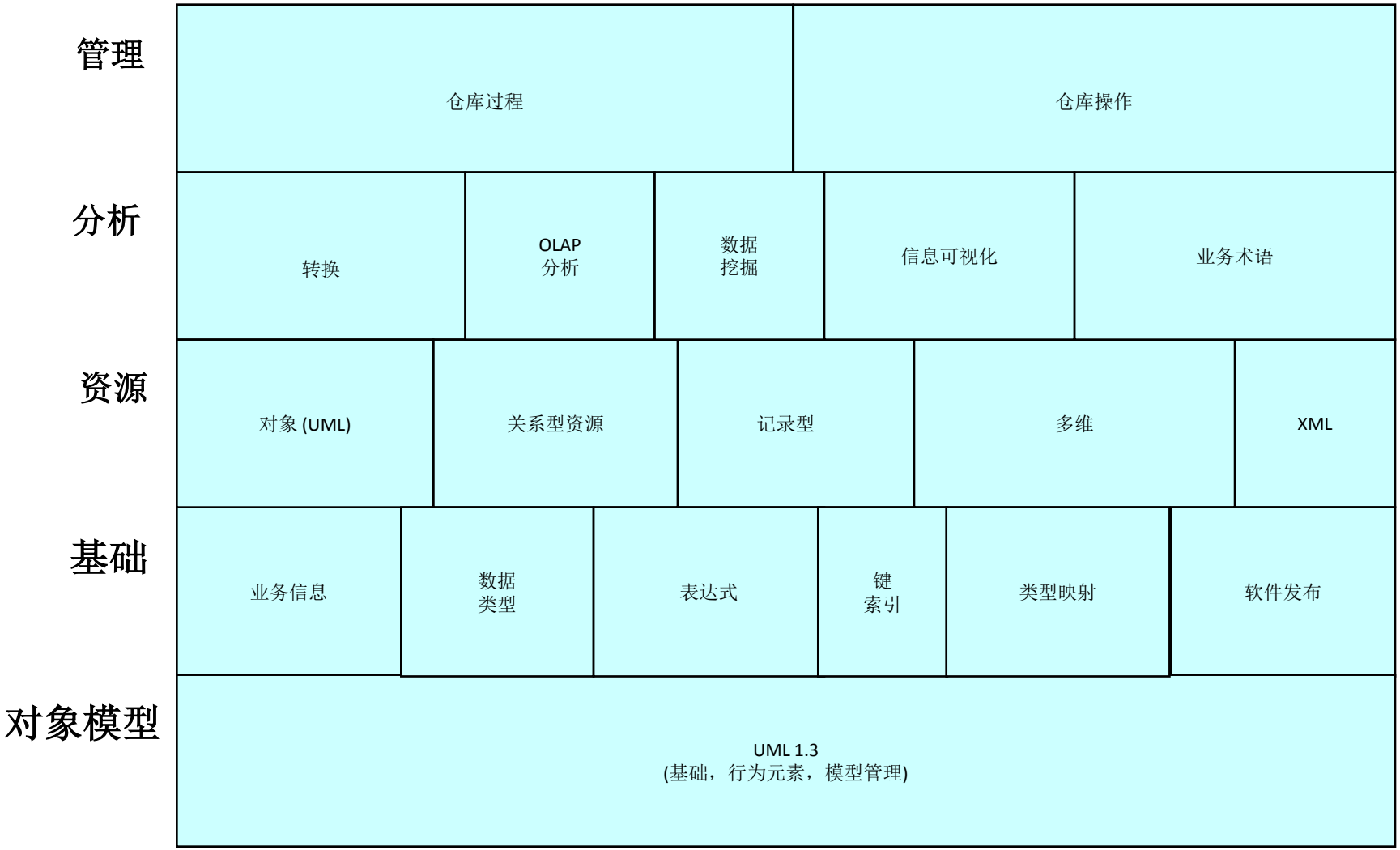
实例包（Instance）

- 提供了在CWM交换中包含带值元数据的基础结构

提纲

- 元数据管理基本概念
  - 元数据定义
  - 元数据管理
- CWM元数据标准
  - CWM标准概述
  - 对象模型层
  - 基础层
  - 资源层
  - 分析层
  - 管理层

基础层（Foundation）

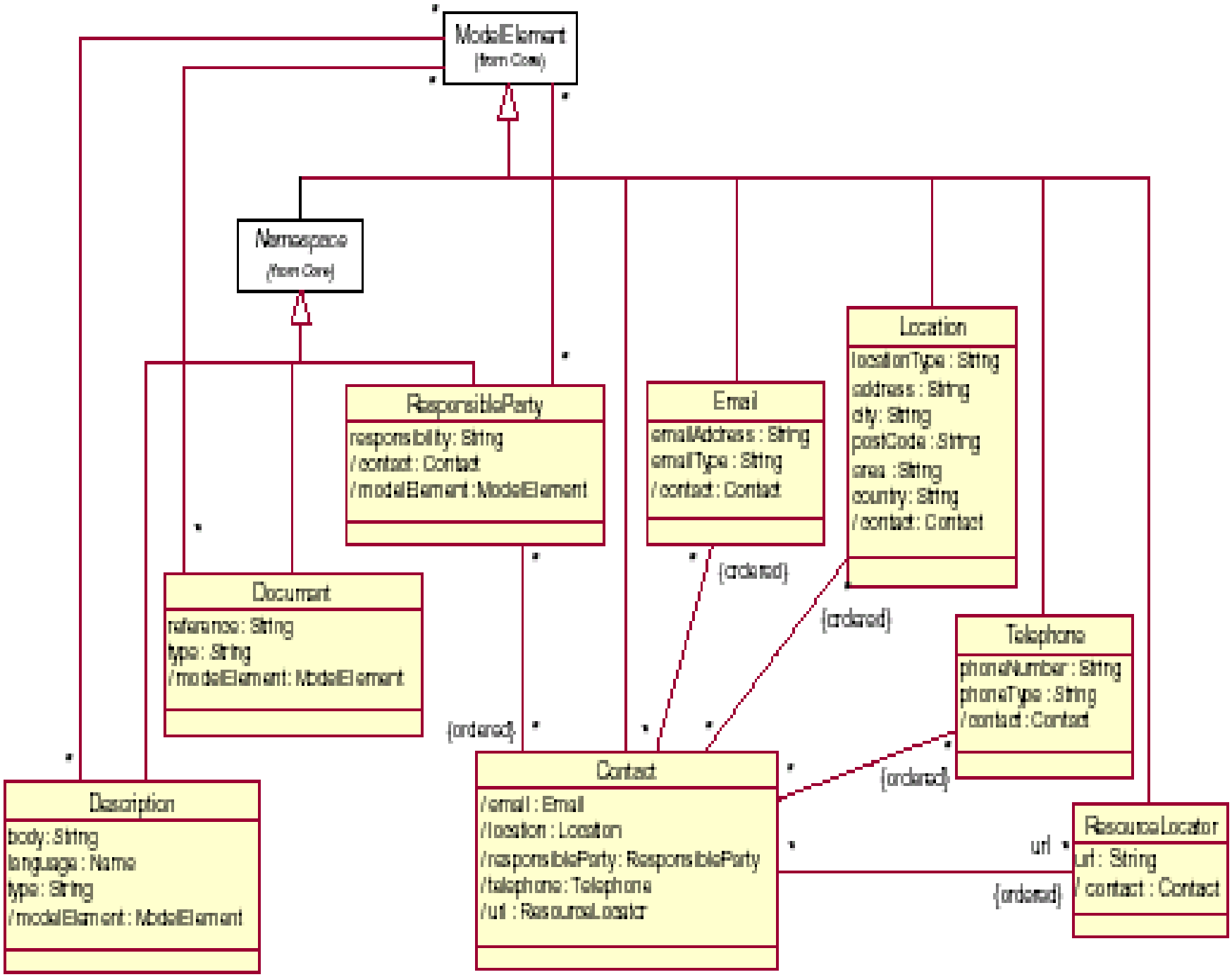


基础层（Foundation）

- 提供为驻留在更高层次的其他包提供CWM特定的服务的包
- 包括6个包
  - 业务信息包（Business Information）
  - 数据类型包（Data Types）
  - 表达式包（Expression）
  - 键和索引包（keys and indexes)
  - 软件部署包（Software Deployment）
  - 类型映射包（Type Mapping）

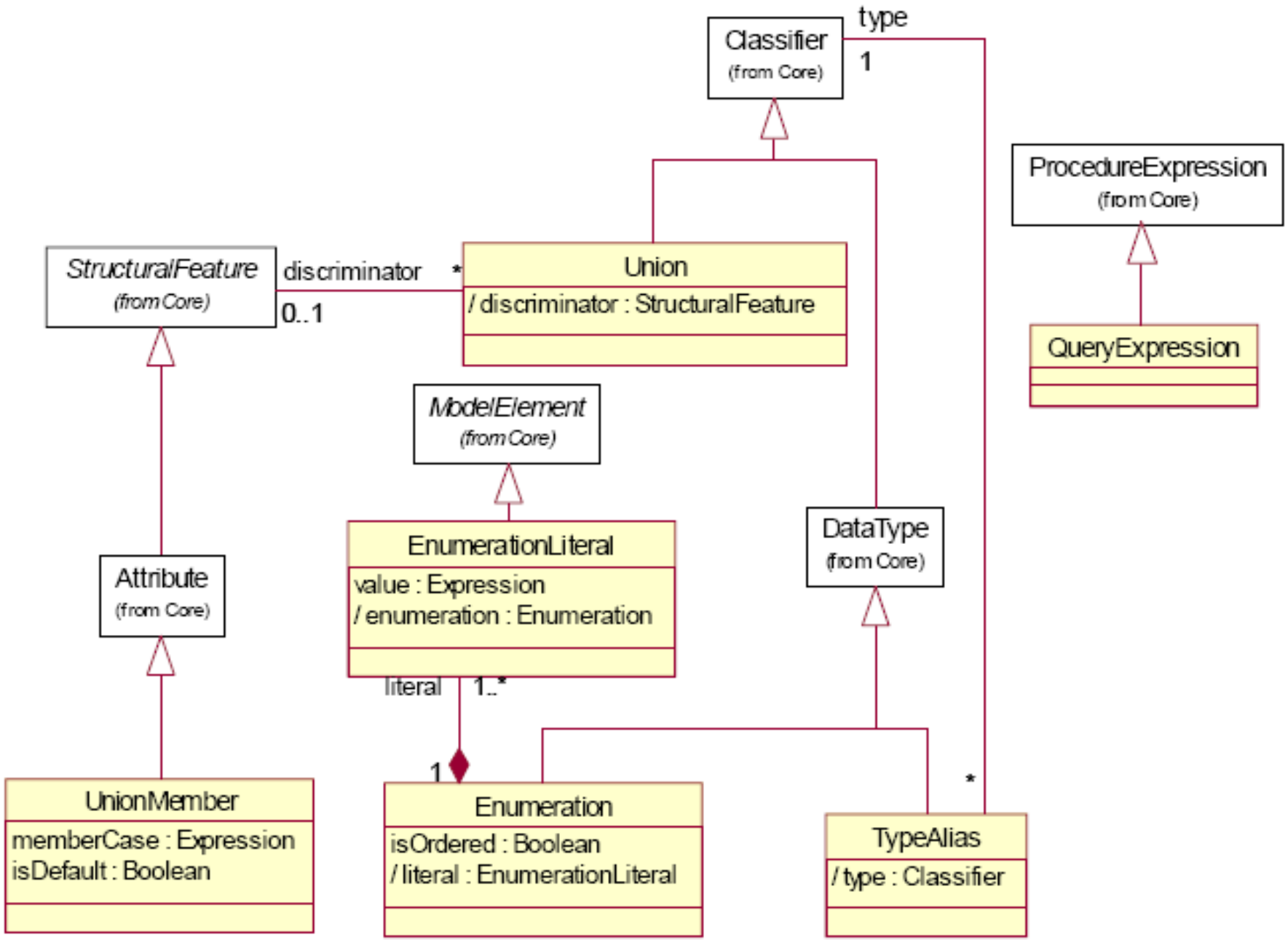
业务信息包（Business Information）

- 业务信息元模型给所有CWM包提供了面向业务的信息
- 这里面向业务指的是支持数据仓库和商业智能
  - 负责单位
  - 如何联络
  - 离线文档
  - ……


ModelElement
{from Core}
Namespace
{from Core}
ResponsibleParty
responsibility: String
/contact: Contact
/modelElement:ModelElement
Email
emailAddress : String
emailType : String
/contact: Contact
Location
locationType : String
address : String
city: String
postCode : String
area :String
country: String
/contact: Contact
Document
reference: String
type: String
/modelElement: ModelElement
{ordered}
{ordered}
Telephone
phoneNumber : String
phoneType : String
/contact:Contact
{ordered}
{ordered}
Description
body: String
language : Name
type: String
/modelElement: ModelElement
Contact
/email : Email
/location : Location
/responsibleParty: ResponsibleParty
/telephone: Telephone
/url : ResourceLocator
url
{ordered}
ResourceLocator
url : String
/ contact : Contact

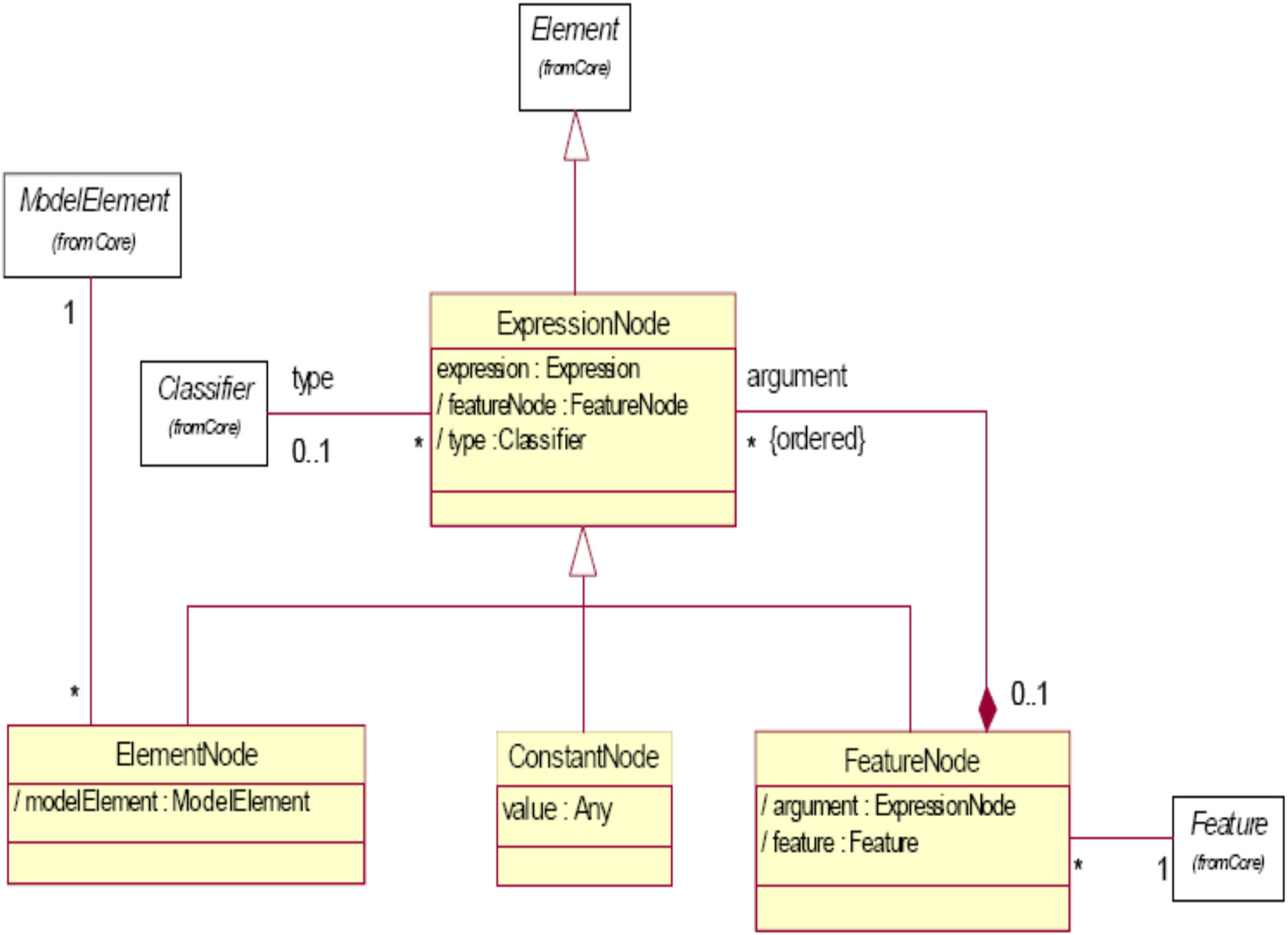
数据类型包（Data Types）

- 提供了支持定义基本数据类型和构造数据类型所需的基本结构
- CWM模型本身没有定义很多的现有类型，但是通过数据类型包可以建立目前现有的大多数系统的数据类型


Classifier
(from Core)
type
1
ProcedureExpression
(from Core)
QueryExpression
StructuralFeature
(from Core)
discriminator
*
0..1
Union
/ discriminator : StructuralFeature
ModelElement
(from Core)
Attribute
(from Core)
EnumerationLiteral
value : Expression
/ enumeration : Enumeration
DataType
(from Core)
literal
1..*
1
*
UnionMember
memberCase : Expression
isDefault : Boolean
Enumeration
isOrdered : Boolean
/ literal : EnumerationLiteral
TypeAlias
/ type : Classifier

表达式包（Expression）

- 表达式包提供了统一的表达式树的格式，可以为转换包等进行服务。
- 统一的表达式格式给ETL流程分析或其他的元数据分析提供了基础


Element
(fromCore)
ModelElement
(from Core)
1
Classifier
(fromCore)
type
0..1
ExpressionNode
expression : Expression
/ featureNode : FeatureNode
/ type : Classifier
*
argument
* {ordered}
*
0..1
ElementNode
/ modelElement : ModelElement
ConstantNode
value : Any
FeatureNode
/ argument : ExpressionNode
/ feature : Feature
*
1
Feature
(fromCore)

键和索引包 （keys and indexes)

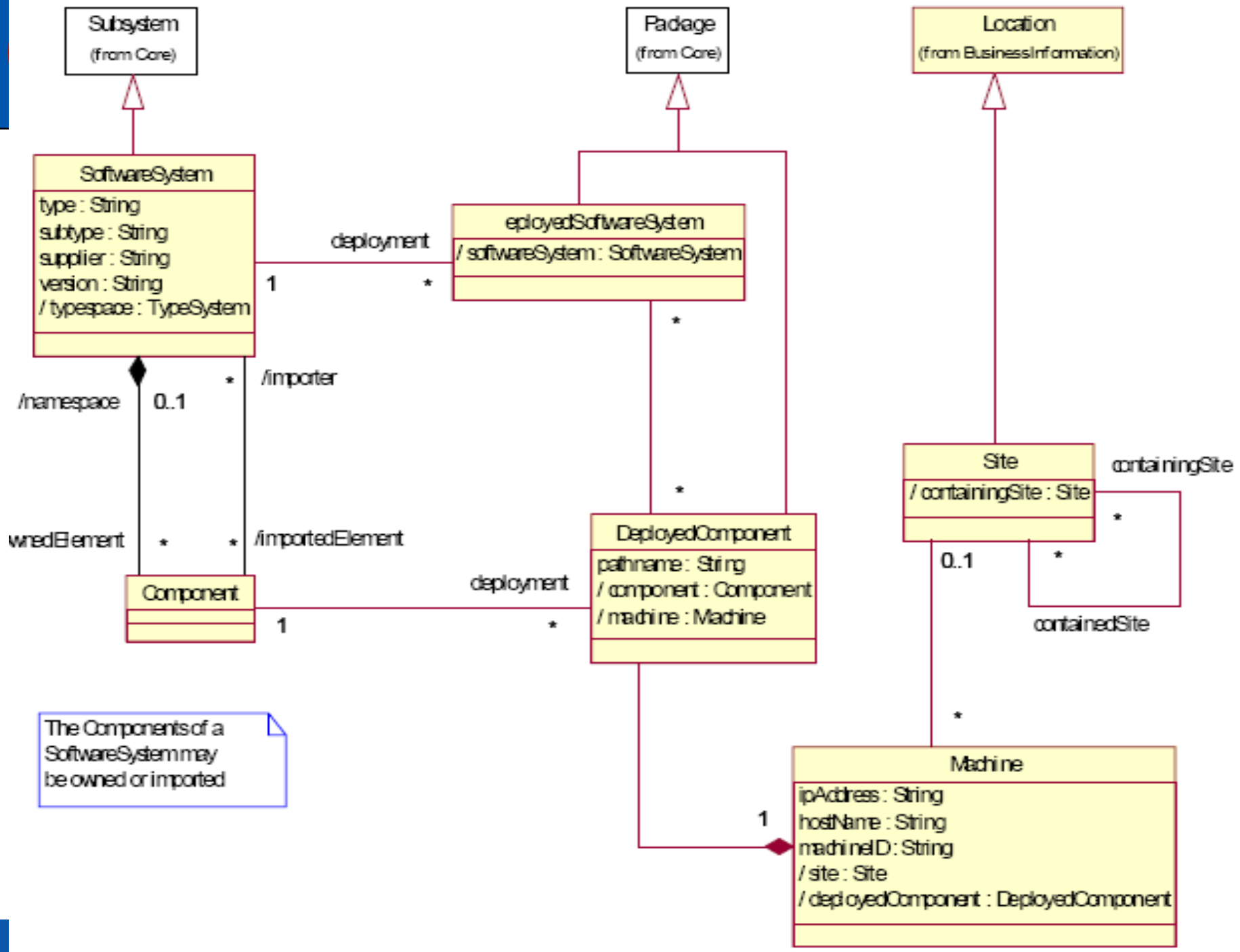
- 键和索引包提供了统一的对元素进行标识、排序和检索所需的方法，可以为其它包所共享
- 索引是按顺序安排的元素列表
- 键是一个或多个值的集合，用来确定数据库中的某项记录

类型映射包 （Type Mapping）

- 定义了作为数据类型集合的类型系统的概念，还支持类型系统间数据类型的转换
- 主要是为满足不同系统之间数据类型差异进行的映射
- 可以进行多对多映射

软件部署包（Software Deployment）

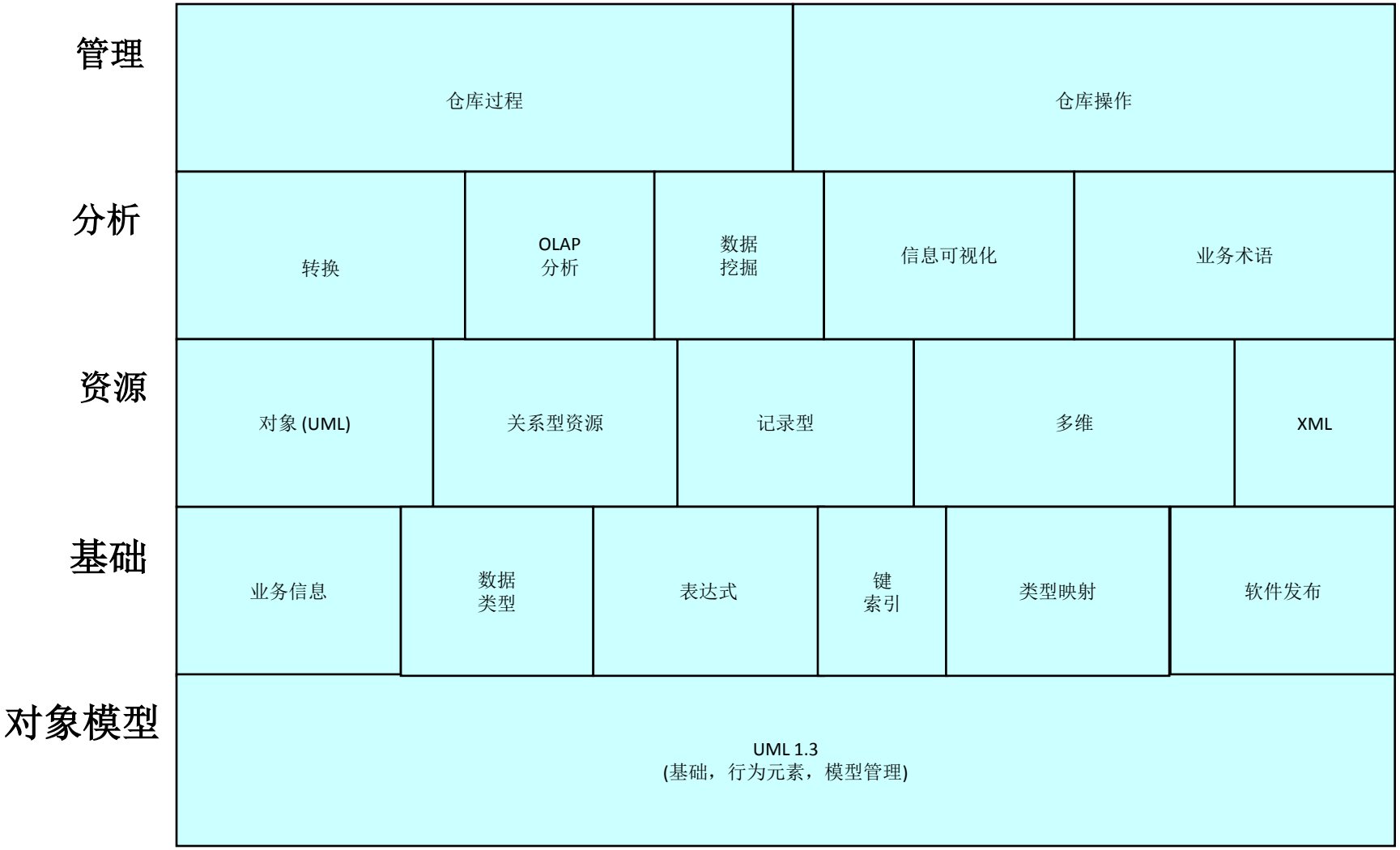
- 为了管理和记录各个软件系统的分布和连接情况，用于记录如何使用数据仓库中的软件和硬件
- 捕捉尽可能多的，其他CWM包需要的可操作的配置信息，而不是完全的通用模型
- 具体内容包括：
  - 系统软件
  - 子系统类型
  - 部署的组件和离散的组件
  - 独立计算机
  - 站点（地点）
  - 数据管理者
  - 数据提供者等


Subsystem
(from Core)
Package
(from Core)
Location
(from BusinessInformation)
SoftwareSystem
type : String
subtype : String
supplier : String
version : String
/ typespace : TypeSystem
eployedSoftwareSystem
/ softwareSystem : SoftwareSystem
deployment
1
*
*
/importer
/namespace
0..1
wnedElement
/importedElement
Component
DeployedComponent
pathname : String
/ component : Component
/ machine : Machine
deployment
1
*
Site
/ containingSite : Site
containingSite
0..1
containedSite
The Components of a SoftwareSystem may be owned or imported
1
Machine
ipAddress : String
hostName : String
machineID : String
/ site : Site
/ deployedComponent : DeployedComponent

提纲

- 元数据管理基本概念
  - 元数据定义
  - 元数据管理
- CWM元数据标准
  - CWM标准概述
  - 对象模型层
  - 基础层
  - 资源层
  - 分析层
  - 管理层

资源层


管理
仓库过程
仓库操作
分析
转换
OLAP
分析
数据
挖掘
信息可视化
业务术语
资源
对象 (UML)
关系型资源
记录型
多维
XML
基础
业务信息
数据
类型
表达式
键
索引
类型映射
软件发布
对象模型
UML 1.3
(基础，行为元素，模型管理)

资源层(Resource)

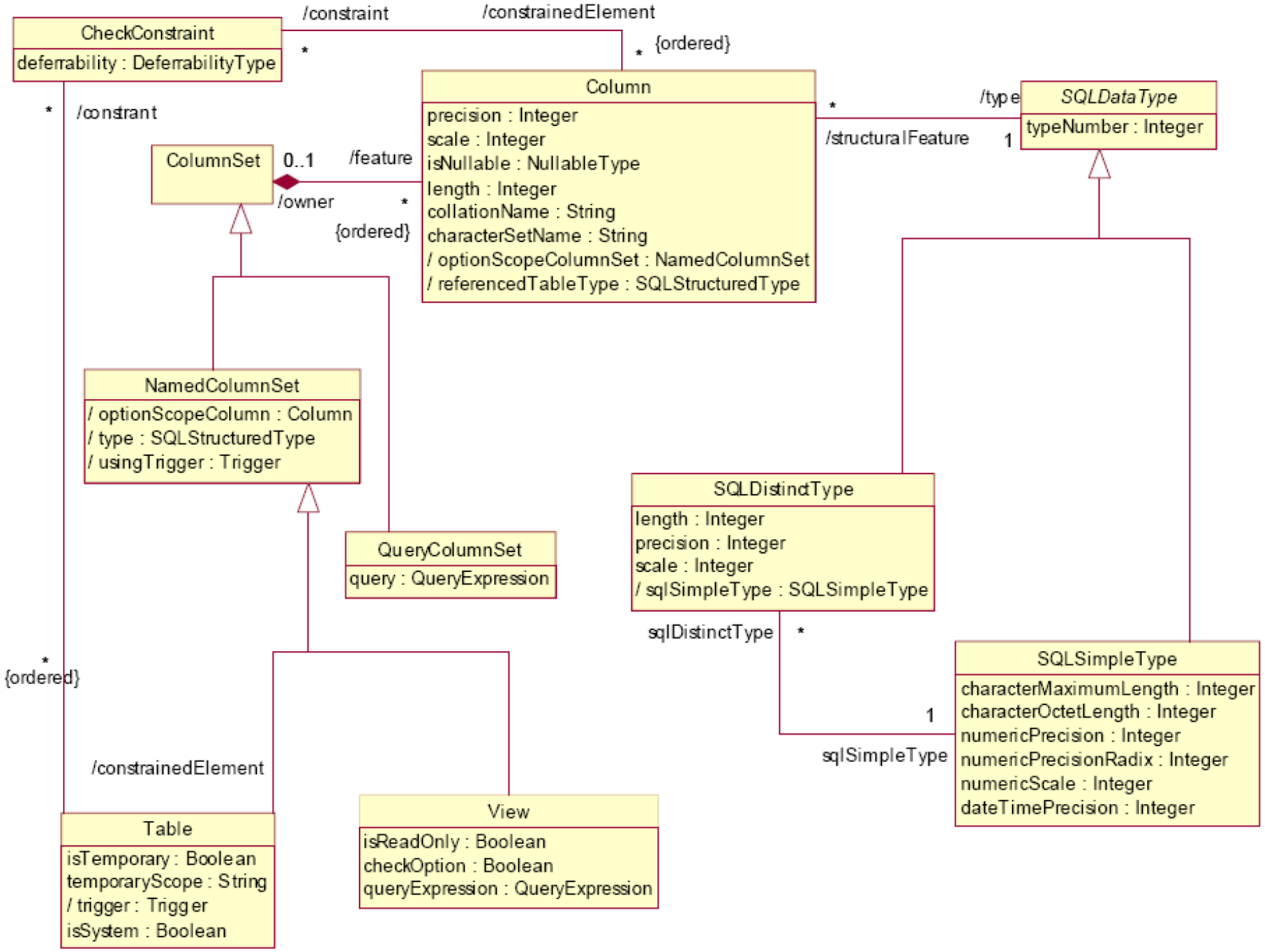
- 描述以CWM为中介的交换中既可作为源又可作为目标的数据资源的结构
- 包括5个包：
  - 对象包（Object Model）
  - 关系型包（Relational）
  - 记录包（Record）
  - 多维包（Multidimensional）
  - XML包（XML）

对象包（Object Model）

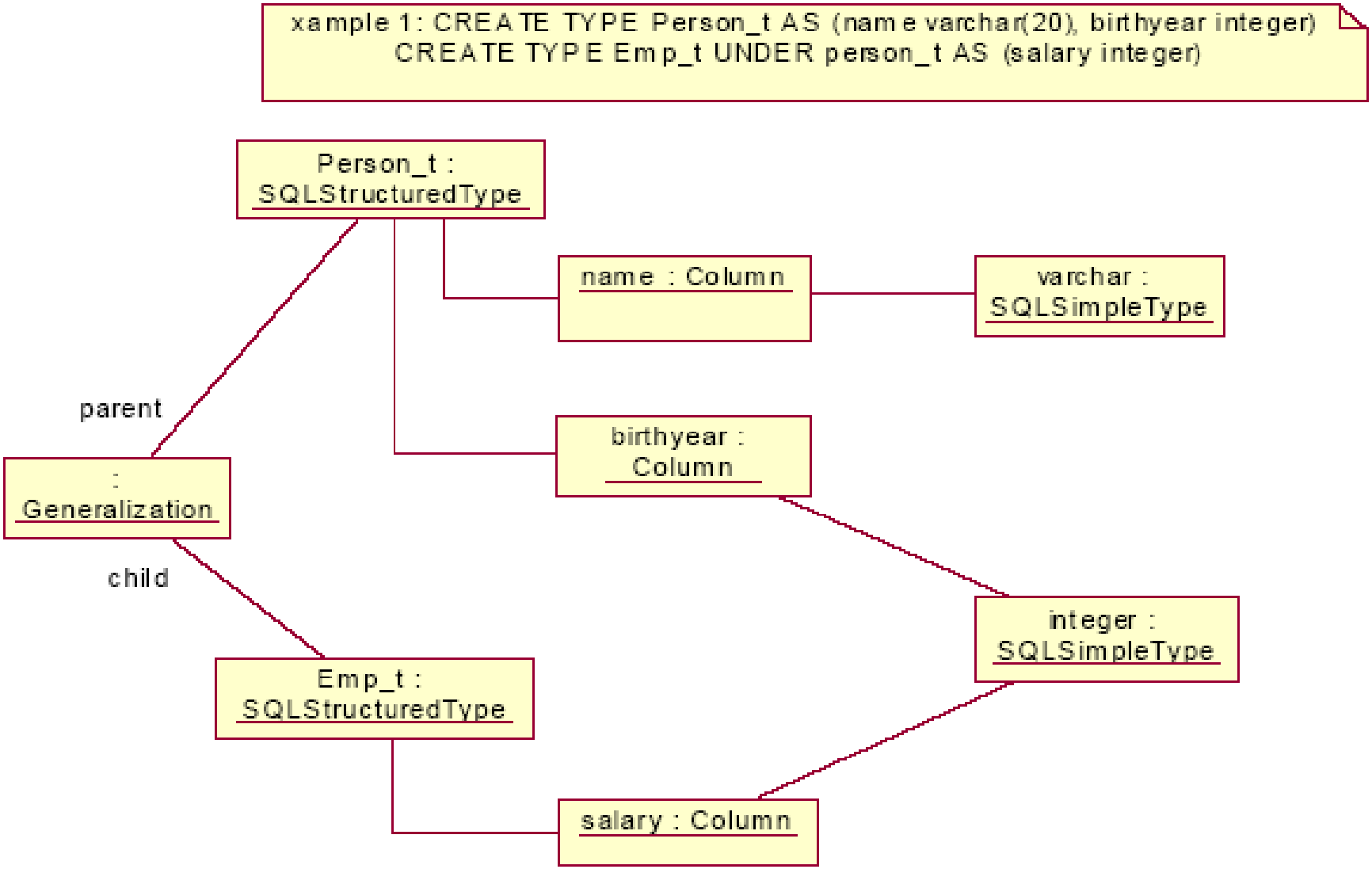
- CWM已经在对象模型层包含了一个非常好的对象模型
- 对象模型层的核心包，行为包，关系包和实例包都可以直接建立面向对象的数据资源描述
- 也用于描述面向对象数据库的结构和面向对象应用组件的结构
- 如果遇到不能处理的特征和功能时，可以定义扩展包来增加处理能力

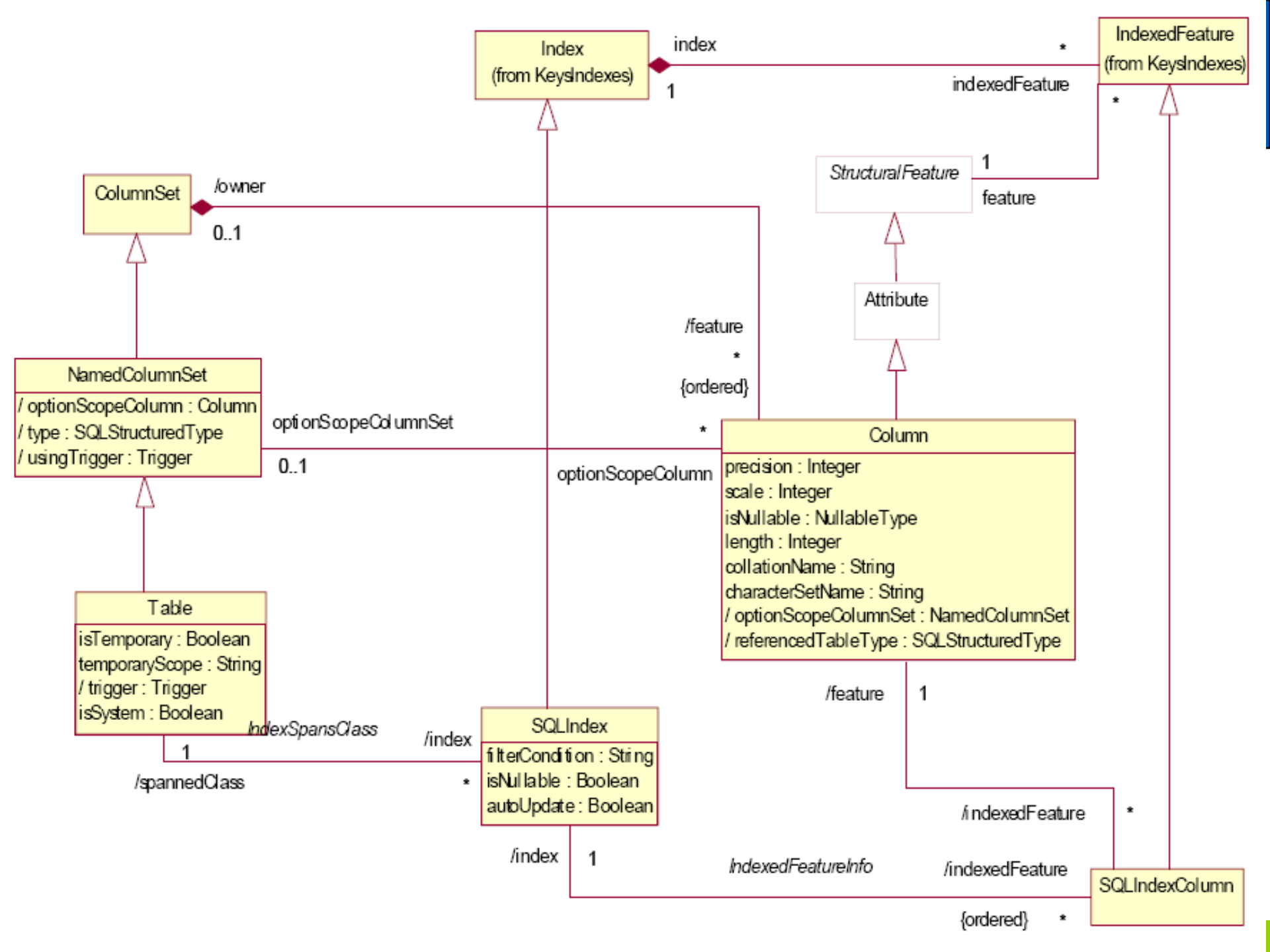
关系型包（Relational）

- 描述关系数据库的模式
- 支持遵从SQL99标准及其面向对象扩展的关系数据库描述
- CWM中最大的包，涉及的类总共68个
  - *Containers*
  - *Tables, Columns, and Data Types*
  - *Structured Types and Object Extensions*
  - *Keys*
  - *Index*
  - *Triggers*
  - *Procedures*
  - *Instances*
  - *…*

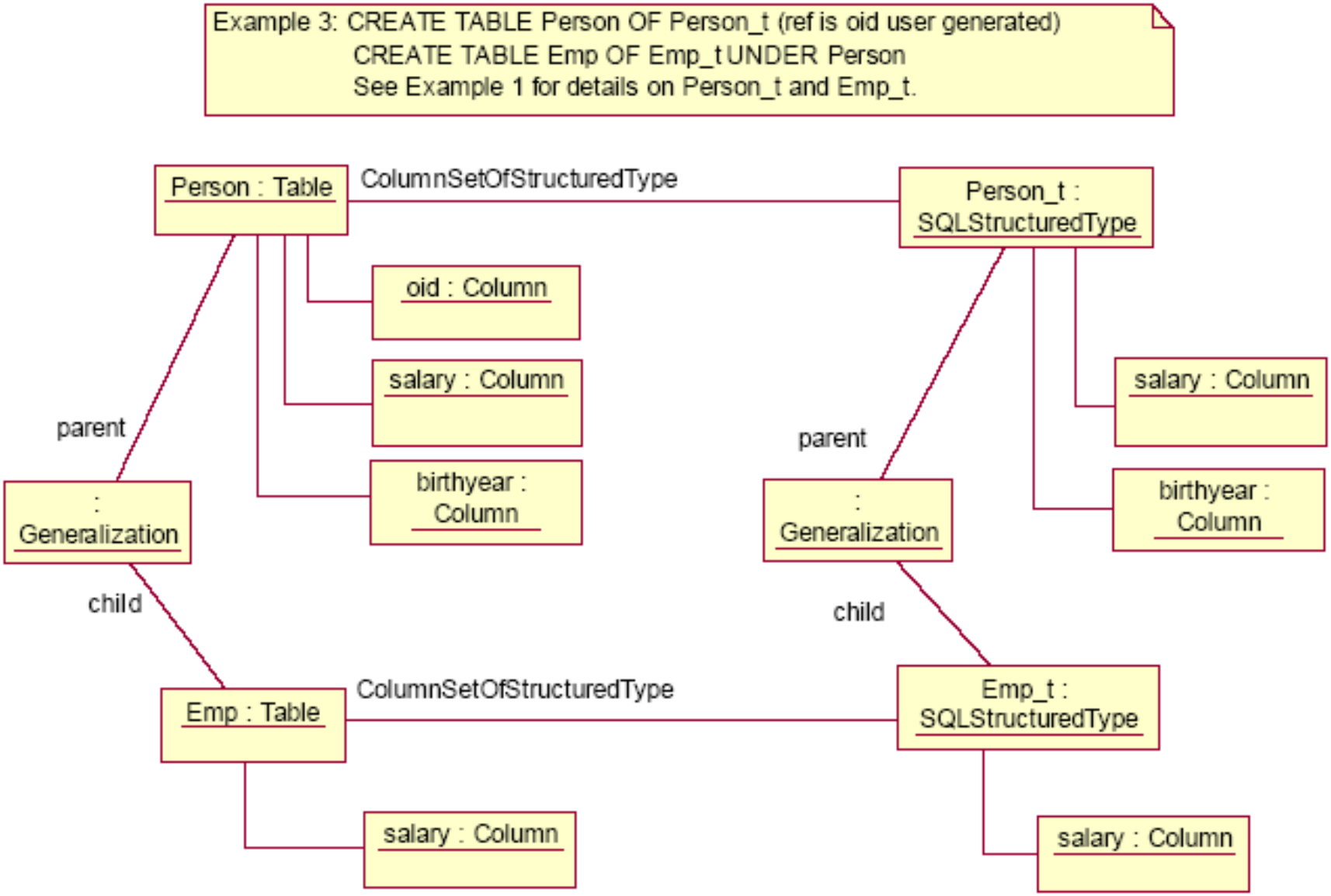

CheckConstraint
deferrability : DeferrabilityType
/constraint
/constrainedElement
*
*
{ordered}
*
/constrant
Column
precision : Integer
scale : Integer
isNullable : NullableType
length : Integer
collationName : String
characterSetName : String
/ optionScopeColumnSet : NamedColumnSet
/ referencedTableType : SQLStructuredType
*
/type
SQLDataType
typeNumber : Integer
/structuralFeature
1
ColumnSet
0..1
/feature
/owner
*
{ordered}
NamedColumnSet
/ optionScopeColumn : Column
/ type : SQLStructuredType
/ usingTrigger : Trigger
QueryColumnSet
query : QueryExpression
SQLDistinctType
length : Integer
precision : Integer
scale : Integer
/ sqlSimpleType : SQLSimpleType
sqlDistinctType
*
1
sqlSimpleType
SQLSimpleType
characterMaximumLength : Integer
characterOctetLength : Integer
numericPrecision : Integer
numericPrecisionRadix : Integer
numericScale : Integer
dateTimePrecision : Integer
*
{ordered}
/constrainedElement
Table
isTemporary : Boolean
temporaryScope : String
/ trigger : Trigger
isSystem : Boolean
View
isReadOnly : Boolean
checkOption : Boolean
queryExpression : QueryExpression

列及数据类型应用举例


xample 1: CREATE TYPE Person_t AS (name varchar(20), birthyear integer)
CREATE TYPE Emp_t UNDER person_t AS (salary integer)
Person_t : SQLStructuredType
name : Column
varchar : SQLSimpleType
parent
birthyear : Column
: Generalization
child
integer : SQLSimpleType
Emp_t : SQLStructuredType
salary : Column

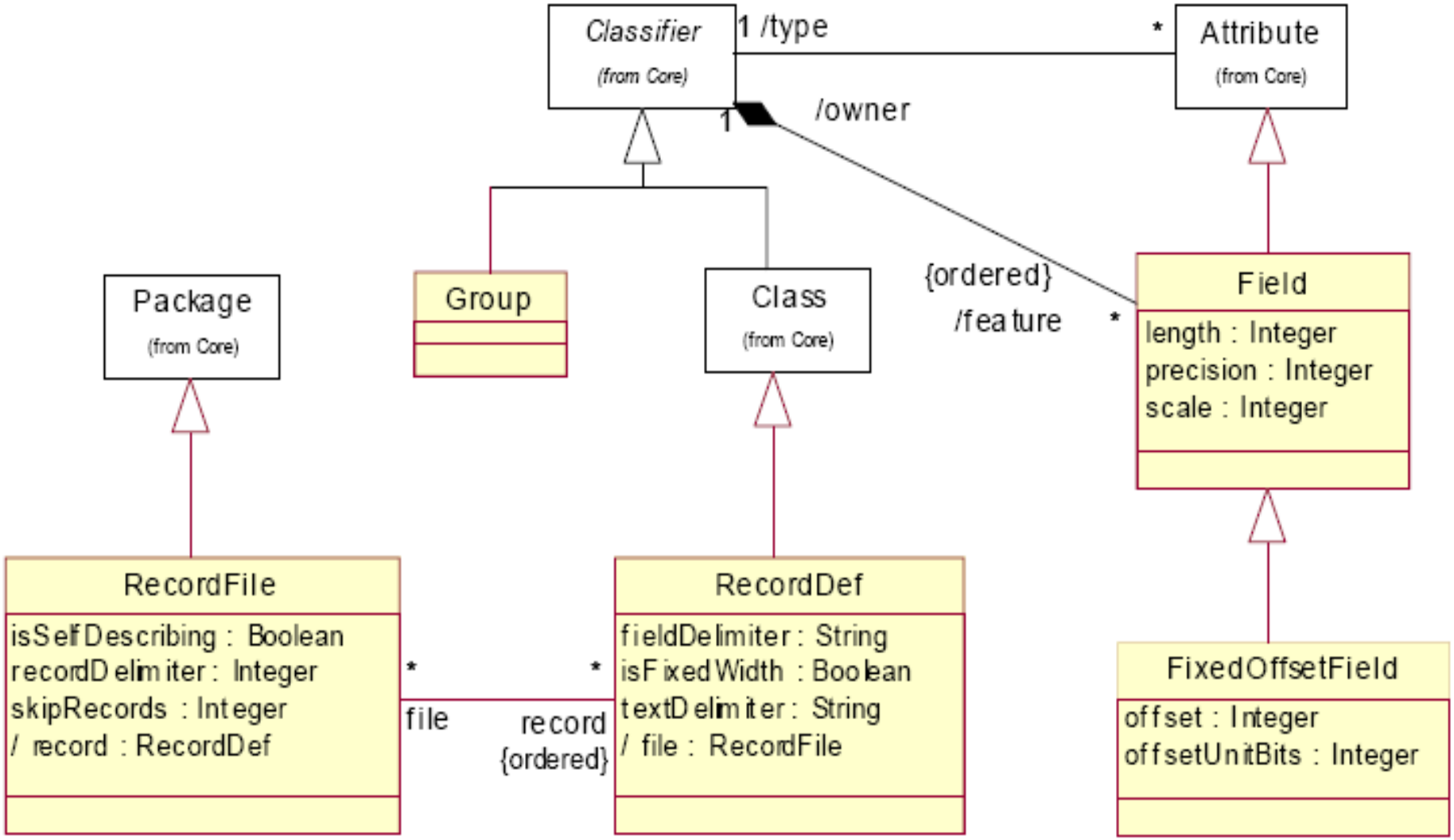

Index
(from KeysIndexes)
index
1
*
IndexedFeature
(from KeysIndexes)
indexedFeature
*
StructuralFeature
1
feature
ColumnSet
/owner
0..1
Attribute
/feature
*
{ordered}
NamedColumnSet
/ optionScopeColumn : Column
/ type : SQLStructuredType
/ usingTrigger : Trigger
optionScopeColumnSet
0..1
*
optionScopeColumn
Column
precision : Integer
scale : Integer
isNullable : NullableType
length : Integer
collationName : String
characterSetName : String
/ optionScopeColumnSet : NamedColumnSet
/ referencedTableType : SQLStructuredType
Table
isTemporary : Boolean
temporaryScope : String
/ trigger : Trigger
isSystem : Boolean
IndexSpansClass
1
/spannedClass
/index
*
SQLIndex
filterCondition : String
isNullable : Boolean
autoUpdate : Boolean
/feature
1
/indexedFeature
*
/index
1
IndexedFeatureInfo
/indexedFeature
SQLIndexColumn
{ordered}
*

表应用举例


Example 3: CREATE TABLE Person OF Person_t (ref is oid user generated)
CREATE TABLE Emp OF Emp_t UNDER Person
See Example 1 for details on Person_t and Emp_t.
Person : Table
ColumnSetOfStructuredType
Person_t : SQLStructuredType
oid : Column
salary : Column
birthyear : Column
parent
: Generalization
child
Emp : Table
ColumnSetOfStructuredType
Emp_t : SQLStructuredType
salary : Column

记录包（Record）

- 提供了用于描述各种面向记录的数据结构的基础结构
- 包括记录的结构、记录的实例、记录文件等


Classifier
(from Core)
1 /type
*
Attribute
(from Core)
/owner
1
Package
(from Core)
Group
Class
(from Core)
{ordered}
/feature
*
Field
length : Integer
precision : Integer
scale : Integer
RecordFile
isSelfDescribing : Boolean
recordDelimiter : Integer
skipRecords : Integer
/ record : RecordDef
*
*
file
record
{ordered}
RecordDef
fieldDelimiter : String
isFixedWidth : Boolean
textDelimiter : String
/ file : RecordFile
FixedOffsetField
offset : Integer
offsetUnitBits : Integer

多维包（Multidimensional）

- 提供关于多维数据库的通用描述
- 包括多维模型中的维、维的层次，维属性、维成员和维度量等数据结构，以及钻取等操作。

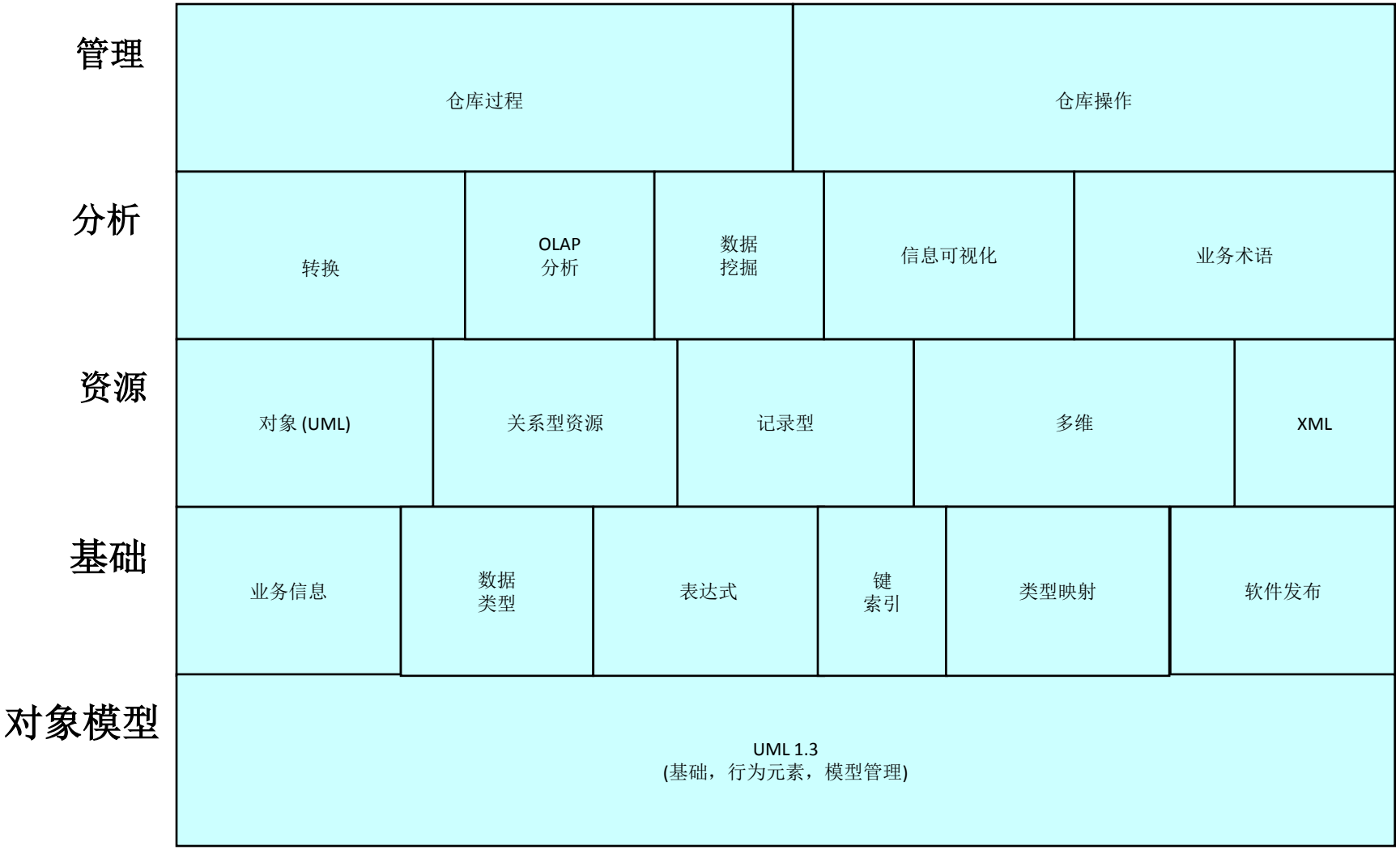
XML包（XML）

- 定义了如何在CWM中使用XML文档描述数据仓库中的数据源
- XML包包含用于描述XML数据源的通用类和关联
- 基于XML 1.0

提纲

- 元数据管理基本概念
  - 元数据定义
  - 元数据管理
- CWM元数据标准
  - CWM标准概述
  - 对象模型层
  - 基础层
  - 资源层
  - 分析层
  - 管理层

分析层


管理
仓库过程
仓库操作
分析
转换
OLAP
分析
数据
挖掘
信息可视化
业务术语
资源
对象 (UML)
关系型资源
记录型
多维
XML
基础
业务信息
数据
类型
表达式
键
索引
类型映射
软件发布
对象模型
UML 1.3
(基础，行为元素，模型管理)

转换包

- 转换包包括了所有BI中涉及到ETL过程的元模型
  - object-oriented
  - Relational
  - record
  - multidimensional
  - XML
  - OLAP
  - data mining.

转换包

- 转换包提供了描述ETL工具和ETL行为的通用的元数据，尤其是：
  - 将ETL过程与数据源和数据目标进行关联。数据源和数据目标可以是任何类型（基于关系或面向对象），任何粒度（类、属性、表、列），并且可以是永久的或易失的。
  - 允许将ETL过程进行分组，并行执行以提高执行效率。包括ETL过程的加载情况，行为和步骤等等。

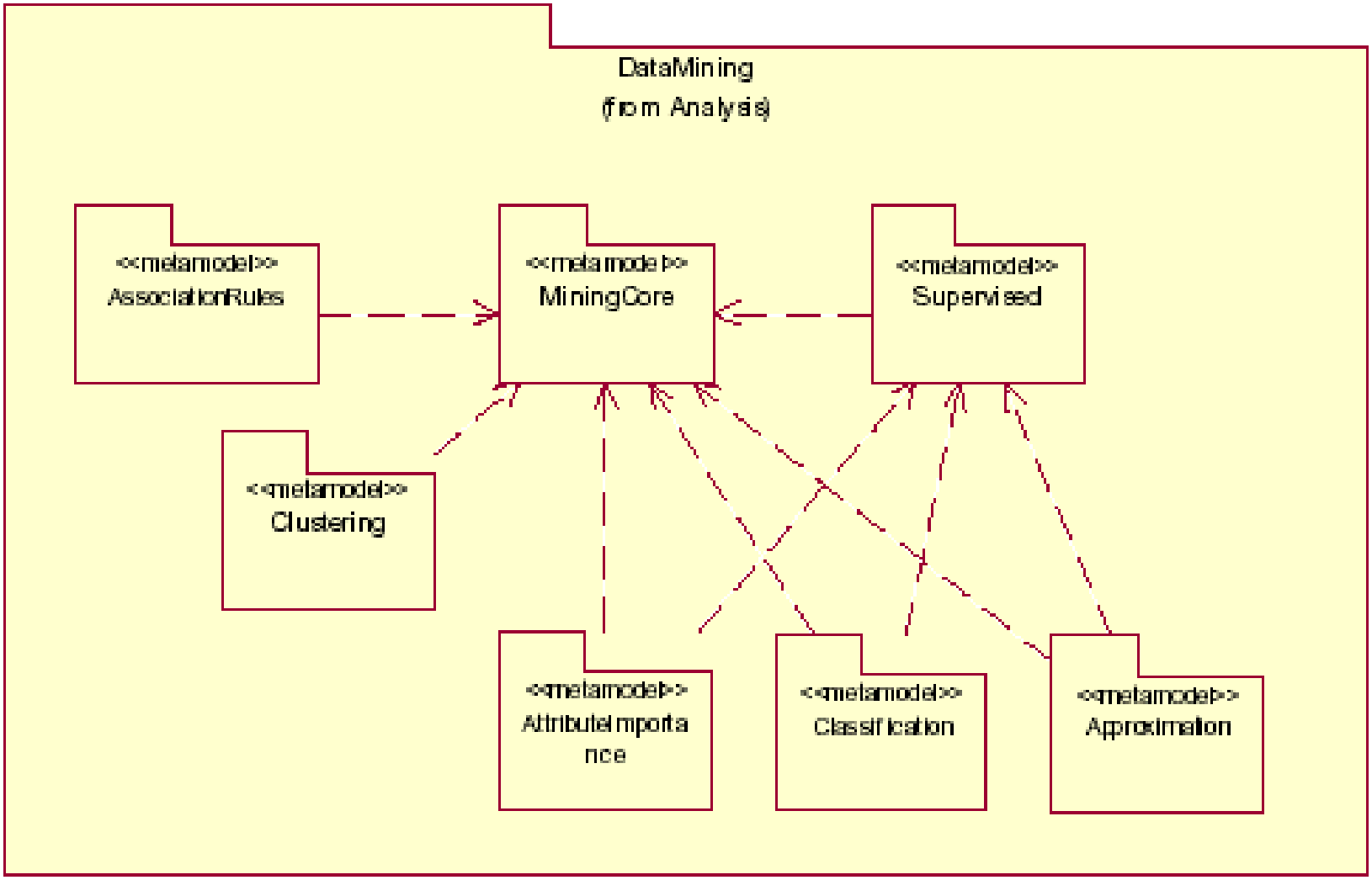
OLAP包

- 定义了描述OLAP系统通用概念的元数据，提供了将OLAP中的元数据内容映射到具体的物理数据源中的方法
- 将OLAP模型映射到CWM数据源的包中，如 CWM 关系型包（ROLAP）或 多维包（MOLAP）.

数据挖掘包

- 为数据挖掘模型构建较为通用的表示方法
- 除数据挖掘计划和模型之外其他一些跨挖掘模型或者挖掘工具的实体（例如分类矩阵）以及它们之间的关系和对技术元数据的映射都包括在数据挖掘元数据的范围之内
- 分析系统中有关数据挖掘的元数据分为七个领域：核心挖掘元数据、和聚类相关的元数据，关联规则元数据，和监督相关元数据，和分类相关元数据，和近似估计相关的元数据与属性重要性的元数据

数据挖掘包


DataMining
(from Analysis)
«metamodel»
AssociationRules
«metamodel»
MiningCore
«metamodel»
Supervised
«metamodel»
Clustering
«metamodel»
AttributeImportance
«metamodel»
Classification
«metamodel»
Approximation

信息可视化包

- 信息可视化元模型定义了支持信息发布和信息可视化的元数据
- CWM信息可视化元模型定义了通用的容器，为实现更加复杂的可视化机制提供支持

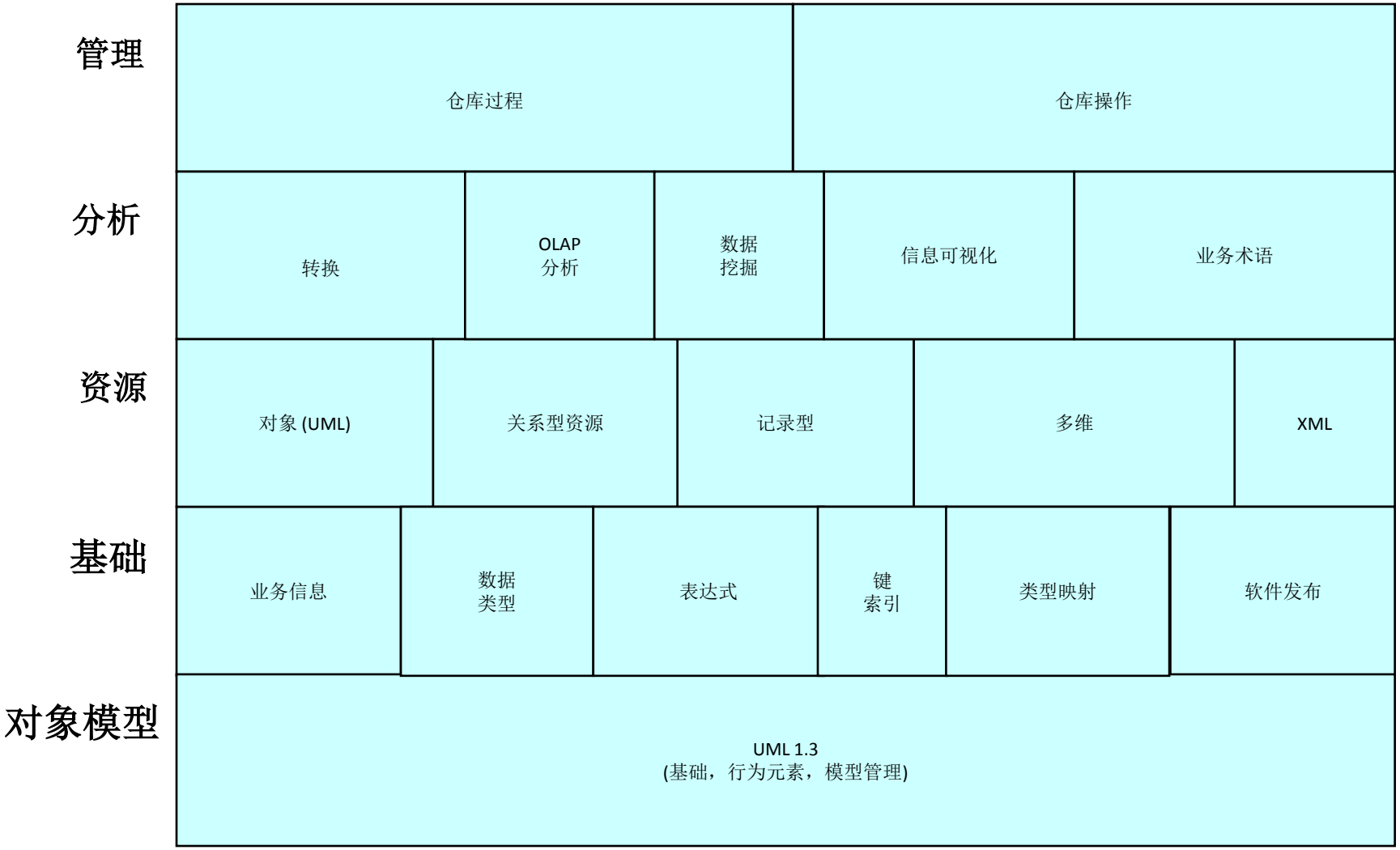
业务术语包

- 数据仓库的用户需要很好的理解仓库中包含的信息，以及仓库提供的工具。比如信息的意义，信息来自于哪个数据源，有哪些工具可以管理及展示这些信息。
- 业务术语包提供了能表达业务元数据的实体和关系。

提纲

- 元数据管理基本概念
  - 元数据定义
  - 元数据管理
- CWM元数据标准
  - CWM标准概述
  - 对象模型层
  - 基础层
  - 资源层
  - 分析层
  - 管理层

管理层


管理
仓库过程
仓库操作
分析
转换
OLAP
分析
数据
挖掘
信息可视化
业务术语
资源
对象 (UML)
关系型资源
记录型
多维
XML
基础
业务信息
数据
类型
表达式
键
索引
类型映射
软件发布
对象模型
UML 1.3
(基础，行为元素，模型管理)

仓库过程

- 数据仓库过程主题描述了数据仓库的信息流。信息流被用来表示转换包中描述的ETL过程。
- 一个仓库过程对象将一个ETL转换过程与一个事件集联系起来，事件集被用来触发转换的执行。

仓库操作（1）

- 仓库操作主题包含了描述数据仓库处理中的日常操作的实体和关系，记录了数据仓库处理中的三类重要事件
  - 转换执行事件
  - 度量事件
  - 请求更改事件

仓库操作（2）

- *转换执行（Transformation Executions）*
  - 记录了最近的ETL过程执行的细节信息，标识了ETL过程开始和结束的时间等。
  - 这些信息可以用来确定数据仓库中一些与过程执行状态相关的特定信息。

仓库操作（3）

- *度量（Measurements）*
  - 度量事件能够为模型元素维护一些度量的标准。
  - 比如它们可以用于保存一个表的真实大小、估计大小和计划的大小。可以协助预测系统的规模并作出决策。

仓库操作（4）

- *请求更改（Change Requests）*
  - 请求更改事件使得影响模型元素的改变能够被详细记录，也可以被用于维护更改的历史记录。
  - 一般会记录哪些请求被执行或拒绝

小结

- 元数据，分析系统关心的元数据
- 元数据管理的意义
- CWM标准
- 对象模型层，基础层，资源层，分析层，管理层

Department of Computer Science, Xiamen University, May 9, 2012