Using UML to Construct a Model Driven Solution for Unified Access to Disparate Data

Randall M. Hauch

VP Development, Chief Architect – Metadata Management

metamatrix®

OMG's Second Workshop on
UML™ for Enterprise Applications:
Model Driven Solutions for the Enterprise
December 3-6, 2001

OBJECT MANAGEMENT GROUP

Overview

- Enterprise Information Metadata
  - The information integration problem
  - *Information metadata* and how its used for information access
- Two ways MetaMatrix uses UML
  - Presentation and diagramming
  - Foundation metamodel
- Lessons Learned

Integrating Enterprise Information

- The need:
  - Faster Time-To-Market
- The benefits:
  - Increase competitive position
  - Greater efficiency and reduced costs
- The approach:
  - Use existing information sources more effectively
  - Eliminate redundant information
  - Leverage new information sources in applications
  - Decouple applications from information sources
  - *Don’t copy or move information!*

Solution is to Use Information Metadata

- Understand information sources
  - Where is information? Which platform?
  - What form is it in?
- Understand information consumers
  - How is information used?
  - In what form is it expected?
- Integrate information
  - Is similar information related?
  - Is information a combination or transformation of other information?

the ***key*** to managing data
is managing ***metadata***

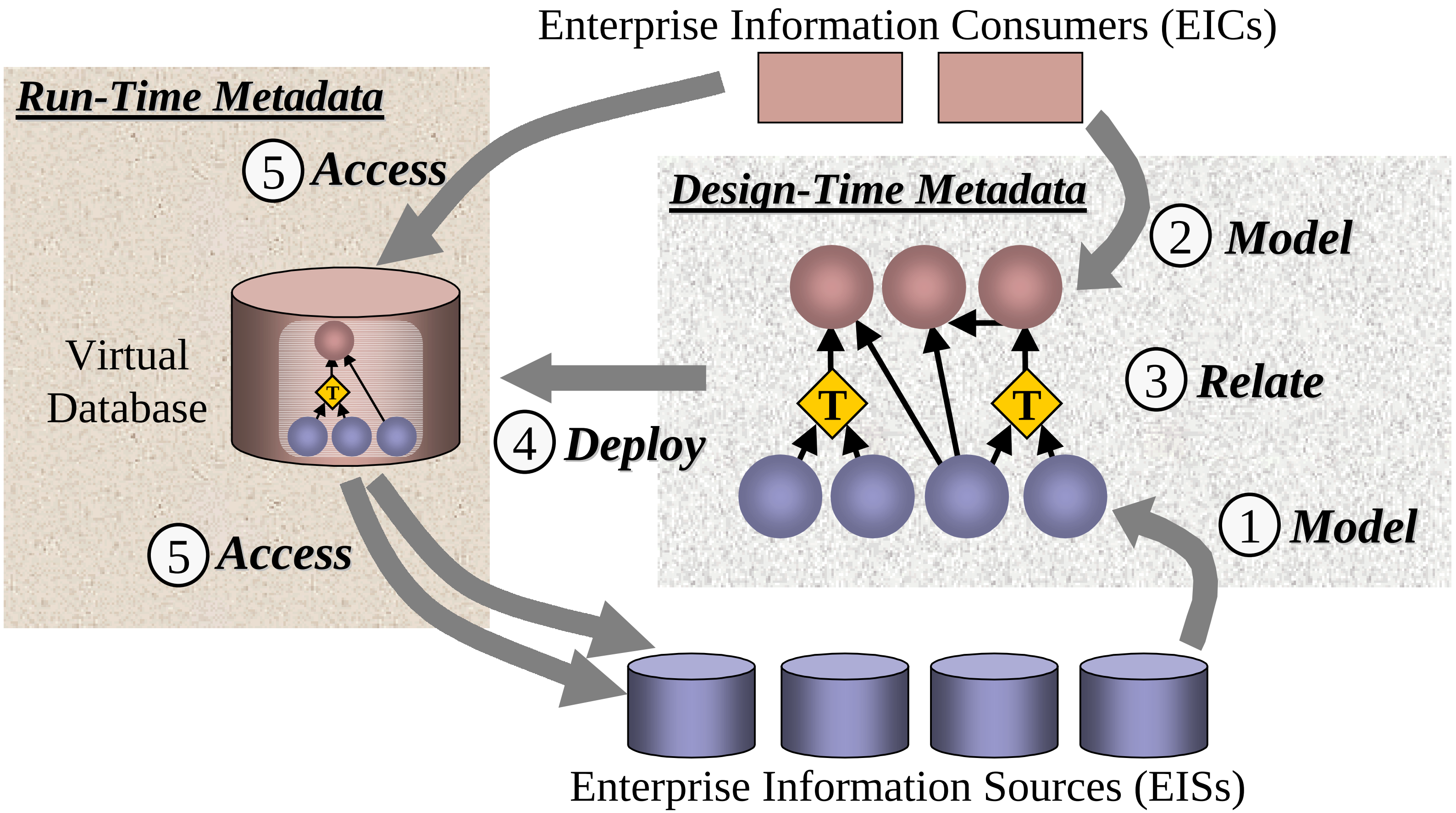
Challenges of Information Metadata

- Want platform-specific metadata
  - Terminology & construction rules
  - Model the sources as naturally as possible
- Want platform-independent metadata
  - Explore and relate metadata from different platforms
  - Relate metadata through transformations and mappings
  - Create “virtual” sources that encapsulate multiple sources

These seemingly contradictory needs can be satisfied by applying MDA, MOF, and UML

MetaMatrix MDA

Model Driven Architecture:
Formal models define access functionality



metamatrix®

MetaMatrix MetaBase™

Design-Time Metadata Management

- MetaData Modeler™
  - Visual tool to collect, model, and manage metadata for enterprise information sources
  - Stand-alone or used with MetaData Server ™
- MetaData Server™
  - Repository for metadata
  - Manage and version models
  - Facilitate enterprise management and sharing

MetaMatrix Information Integration Server

Runtime Metadata for Information Access

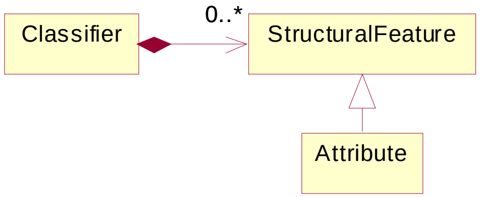
- Includes MetaBase™
- Information Integration™ Server
  - Scalable, fault-tolerant distributed server
  - Pluggable connectors for various information source platforms
  - Access disparate sources as if single source
- Connector Development Kit™ (CDK)
  - Bench-test environment for custom connectors
- Console
  - Tool for remote administration
  - Monitor, manage, and configure distributed servers

MetaMatrix MetaData Modeler™

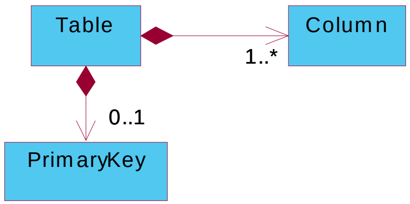
Platform-Specific Modeling

- Metamodels used to define platform metadata
  - Terminology & construction rules
  - Model the sources as naturally as possible
  - Based upon CWM metamodels *(see Lessons)*
- Samples:

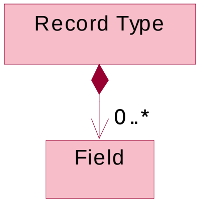
Object-Oriented



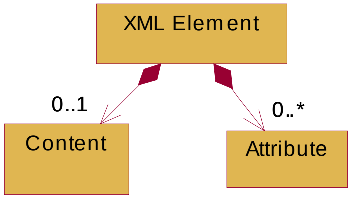
Relational



File

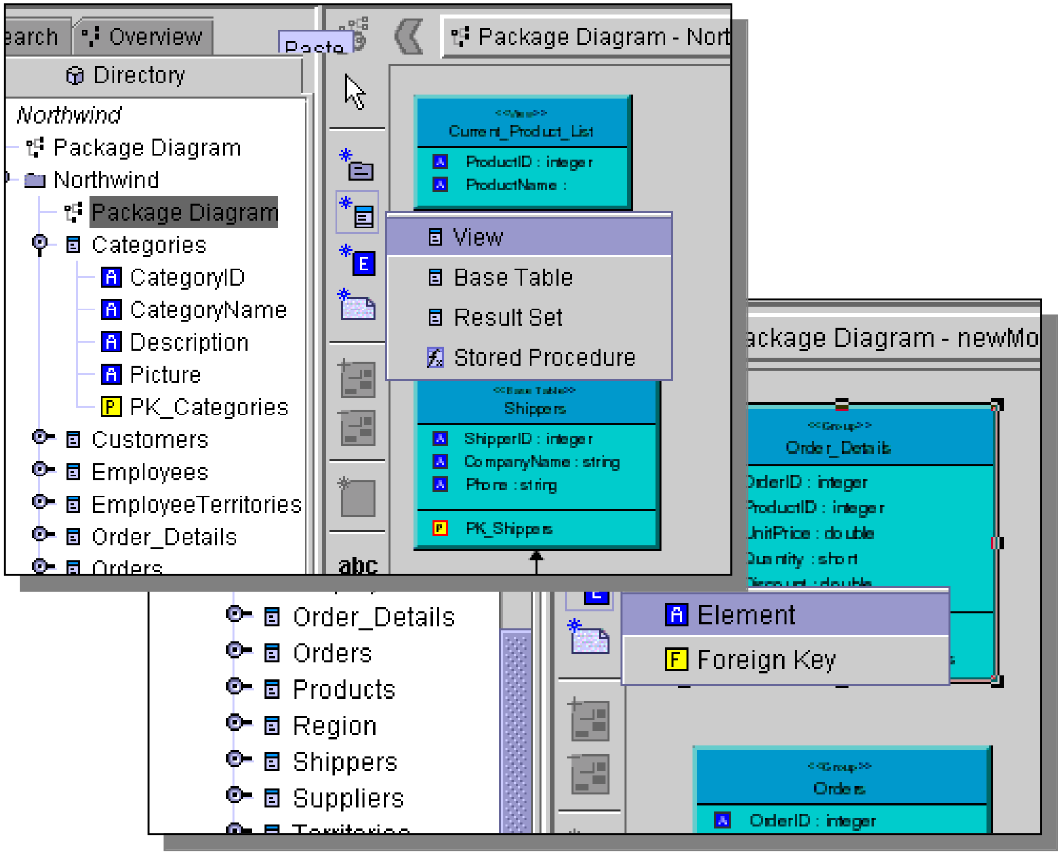


XML



MetaMatrix MetaData Modeler™

Platform-Specific Modeling

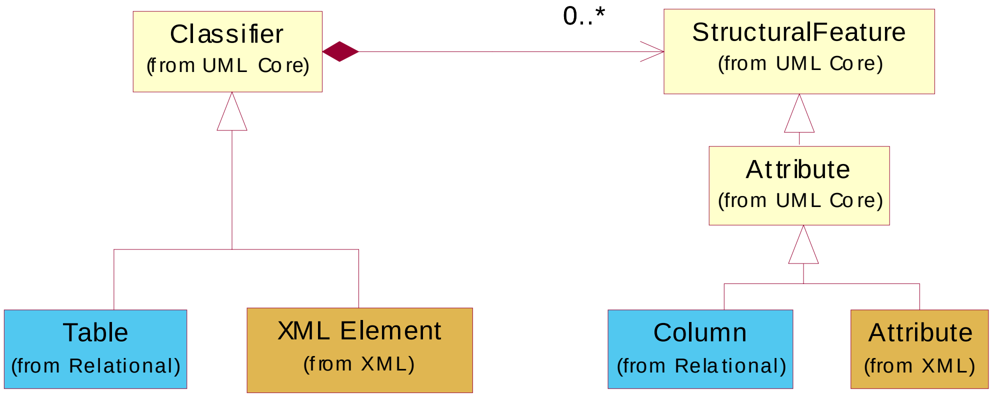


- Metamodels dictate the constructs and instantiation rules
- Platform-specific terminology maintained and presented to user

MetaMatrix MetaData Modeler™

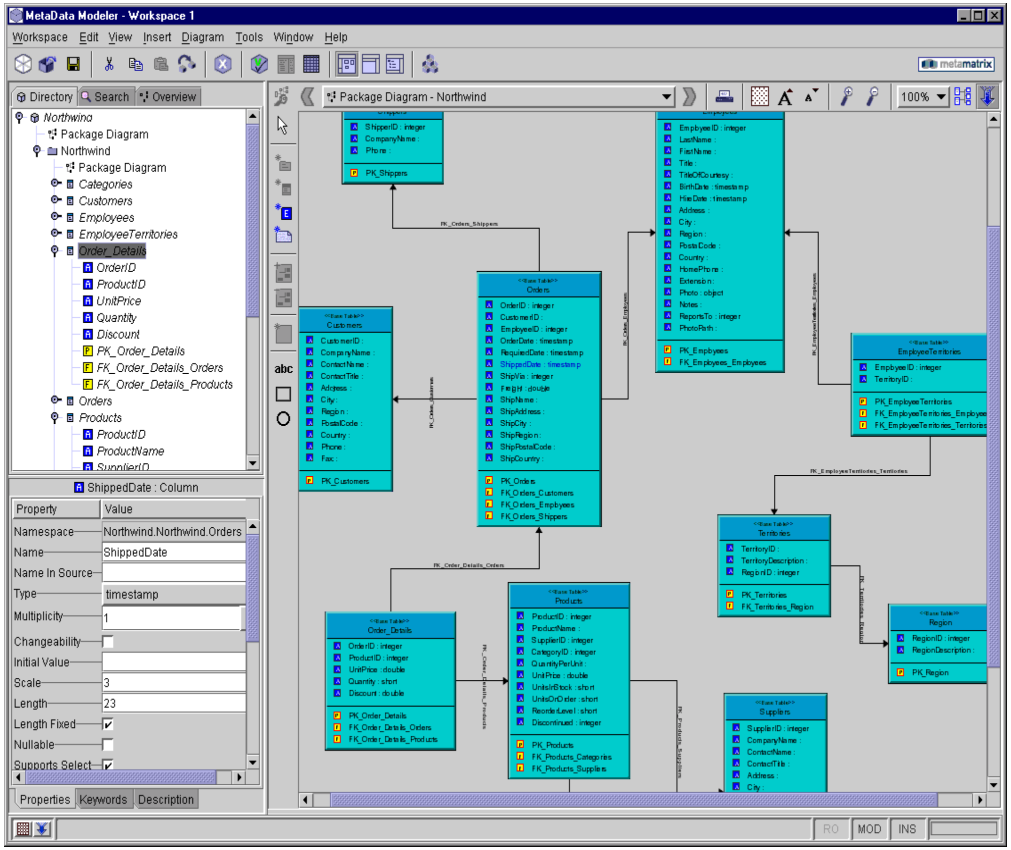
Platform-Independent Modeling

- Supports multiple metamodels
  - – Each defined with MOF
  - – Each extend UML "plus" *(see Lessons)*
- Enables abstraction
  - – UML used for presentation
  - – Transformations defined independently of platform
- Samples:



MetaMatrix MetaData Modeler™

Platform-Independent Modeling



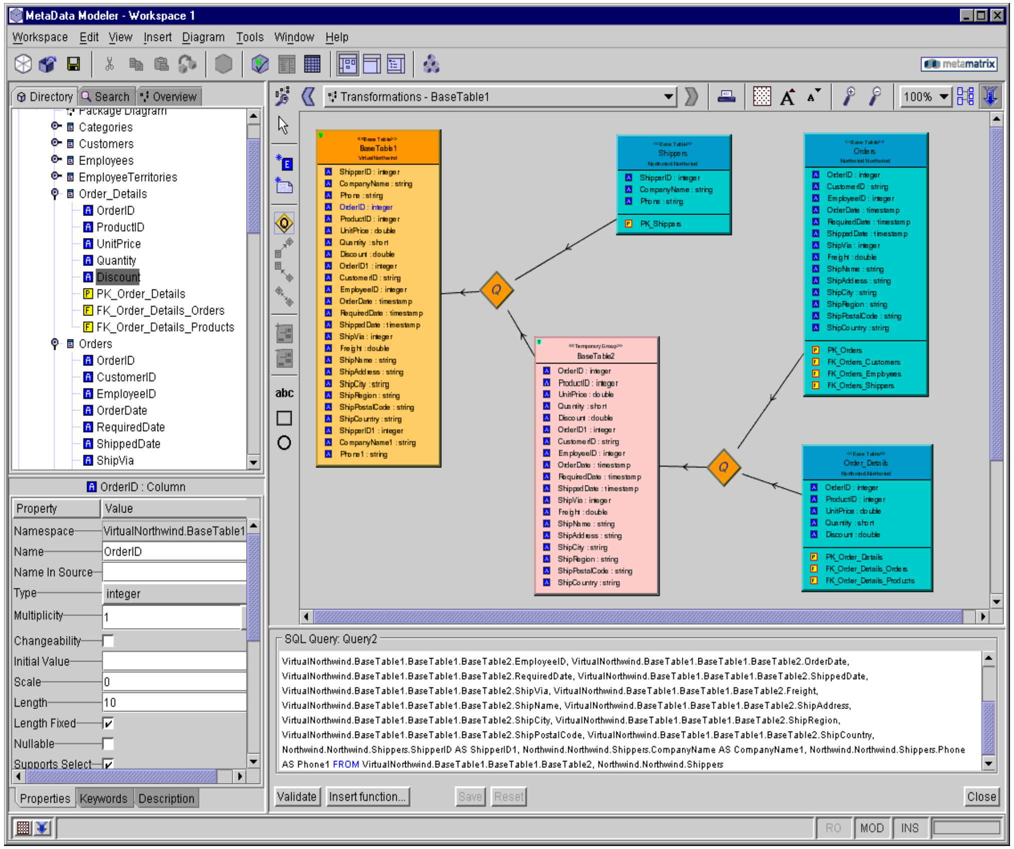
ShippedDate : Column

Property	Value
Namespace	Northwind.Northwind.Orders
Name	ShippedDate
Name In Source	
Type	timestamp
Multiplicity	1
Changeability	□
Initial Value	
Scale	3
Length	23
Length Fixed	☑
Nullable	□
Supports Select	☑

- Shows structure with class and package diagrams
- Displays in a consistent way metadata from different platforms
- Uses and retains platform-specific terminology

MetaMatrix MetaData Modeler™

Platform-Independent Modeling



OrderID : Column

Property	Value
Namespace	VirtualNorthwind.BaseTable1
Name	OrderID
Name In Source	
Type	integer
Multiplicity	1
Changeability	□
Initial Value	
Scale	0
Length	10
Length Fixed	☑
Nullable	□
Supports Select	☑

SQL Query: Query2

```
VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.EmployeeID, VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.OrderDate,
VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.RequiredDate, VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.ShippedDate,
VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.ShipVia, VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.Freight,
VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.ShipName, VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.ShipAddress,
VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.ShipCity, VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.ShipRegion,
VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.ShipPostalCode, VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2.ShipCountry,
Northwind.Northwind.Shippers.ShipperID AS ShipperID1, Northwind.Northwind.Shippers.CompanyName AS CompanyName1, Northwind.Northwind.Shippers.Phone
AS Phone1 FROM VirtualNorthwind.BaseTable1.BaseTable1.BaseTable2, Northwind.Northwind.Shippers
```

- Shows structural transformations from one or more other classifiers
- Defines transformations with
  - Selects
  - Joins
  - Criteria
  - Functions
  - Unions
  - Etc.

Summary

- Enterprise Information Metadata
  - Differs from application metadata
  - Used to model, integrate, and access information sources
  - Needs both platform-independent and platform-specific metadata

Formal modeling of metadata enables information access and integration

metamatrix®

Summary (cont'd)

- Metamodels
  - Defined with MOF and UML
  - Dictate constructs and rules for platform-specific metadata
  - Can be treated in a platform-independent manner if extend UML
  - Are data, enabling adaptive modeling or "metamodel-driven" modeling

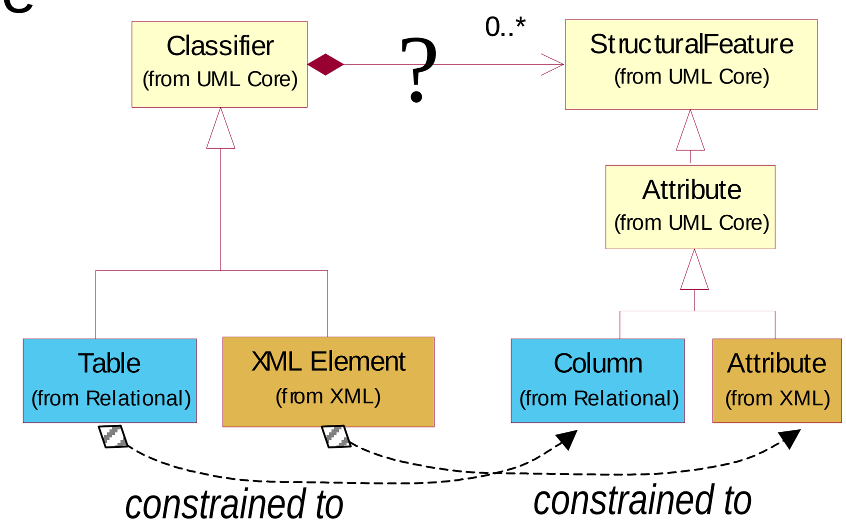
Formal use of metamodels enables modeling of enterprise information metadata

Lessons

- UML and Information Metadata
  - Good for displaying structural features of data source metadata
    - Relationship between MOF and UML critical
  - Missing important constructs found in information metadata
    - Primary Keys
    - Indexes
- XMI
  - Powerful and flexible
  - Overuse of XMI extensions will reduce effectiveness of exchanging XMI for interoperability

Lessons (cont'd)

- CWM metamodels are good for interchange
- CWM metamodels limited for use in metamodel-driven solutions
  - Contain inconsistencies
    - Similar constructs don't extend the same UML construct
  - DTD form is not conducive
    - XMI would be better
  - Missing information
    - Well-formedness rules and constraints in the CWM metamodel specifications but not metamodel files



metamatrix®

Randall M. Hauch
VP, Chief Architect – Metadata Management
rhauch@metamatrix.com

metamatrix®

the ***key*** to managing data
is managing ***metadata***