# XML Metadata Interchange (XMI)

## Revised Proposal to OA&DTF RFP - 3
## Stream based Model Interchange Format (SMIF)

## Co-Submitters

Unisys, IBM, DSTC, Oracle, Platinum,
Fujitsu, Softeam, Reccerca, Daimler-Benz

## Supporters

Cayenne, Genesis , Inline, Rational, Select, Sprint, Sybase, Xerox, MCI Systemhouse,
Boeing, Ardent, Aviatis, ICONIX, Integrated Systems, Verilog, Telefonica I+D,
Universitat Politecnica de Catalunya, NCR, Nihon Unisys, NTT

**OMG TC Meeting, Burlingame : November 11, 1998**

**Sridhar Iyengar : Sridhar.Iyengar@mv.unisys.com**

**Steve Brodsky : sbrodsky@us.ibm.com**

# Topics Covered

- Introduction & History
- Overview of Proposal
- Requirements and proof of concept
- An overview of XML (or why XML?)
- Proposal Details
- Summary

# Introduction

- In November 1997, the MOF and UML were adopted as OMG standards.
- The specifications included metamodel and set of CORBA interfaces for manipulating MOF based meta objects and UML based models
- However a file/stream based interchange format was not specified (time constraints…)
  - In December 1997, the SMIF RFP was issued
- The three initial submissions XMI, CDIF and UOL have now been integrated into one - XMI

# XMI Submission History/factoids

- <2/98 : Unisys and IBM independently evaluate XML and other interchange options
- 2/98 : Unisys and IBM begin collaboration
- 4/98 : DSTC, Oracle and Platinum join
- 5/98 : Select Software joins
  Early discussions with other submitters
- 6/98 : XMI submission preview (Orlando)
  Rational Software, Sybase, Inline, Genesis, Cayenne, Sprint join
- 7/98 : XMI 1.0 submission and presentation, CDIF and UOL submitters begin working with XMI submission
- 9/98 : XMI revised submission status, industry support increases
- 11/98 : Combined XMI submission : 25+ submitters and supporters

# Overview of proposal

- Use W3C Extensible Markup Language (XML) for the transfer syntax and interchange format
  - Specify XML Document Type Definitions (DTD) to enable transfer and verification of
    - UML based models (using UML DTD)
    - MOF based metamodels (using MOF DTD and production rules for mapping MOF metamodels to DTDs)
- Specify a precise MOF to XML mapping
  - Use of OCL to specify stream production rules
  - Allows interchange of any MOF based metamodel
  - Enables automatic generation of DTDs
- Use MOF as metameta model

# OMG Repository and Modeling Architecture

Tools, Applications, Repositories

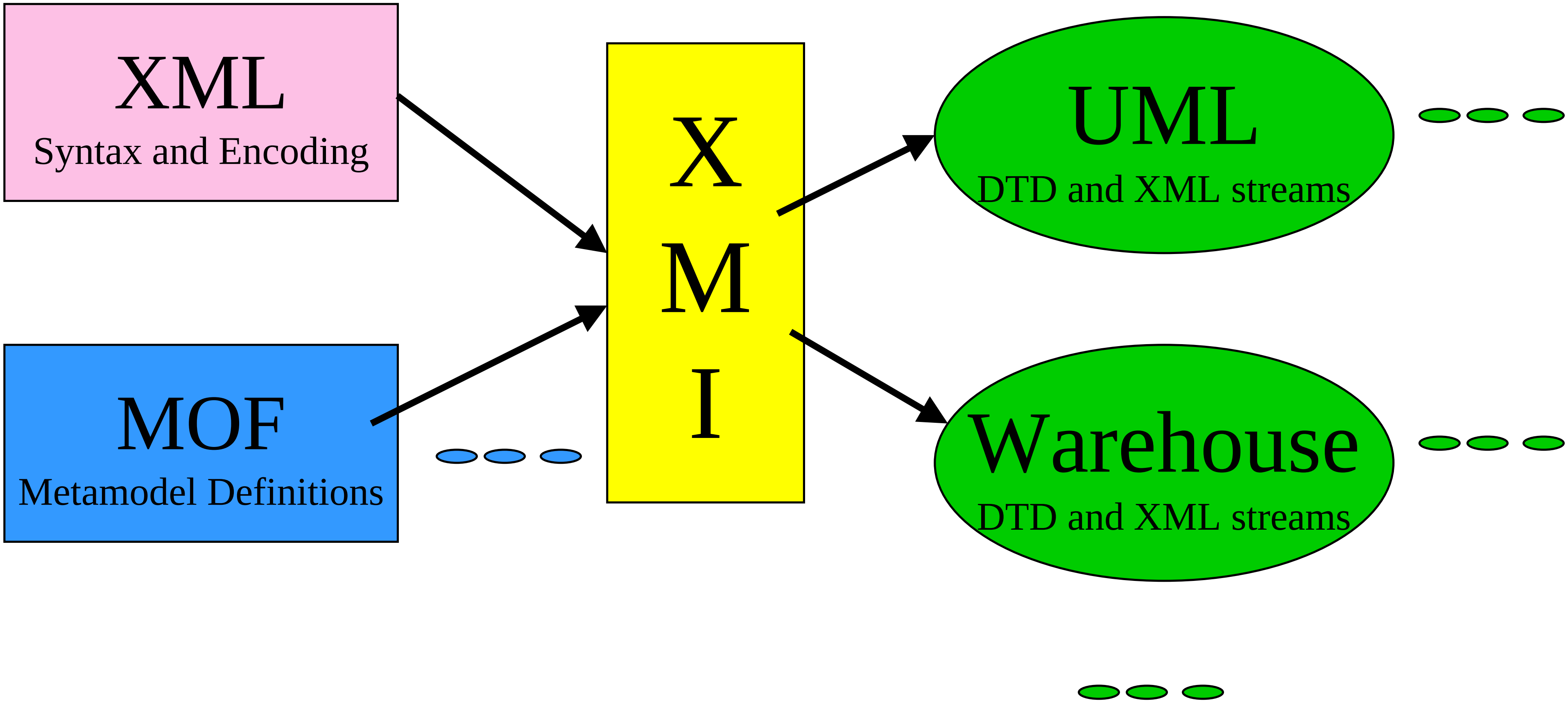
MetaModels (UML, CWM…)

Meta Object Facility (MOF),
Stream based Model Interchange Format (XMI)

CORBA  Object Services

CORBA

# XMI Simplified



XML

Syntax and Encoding

MOF

Metamodel Definitions

XMI

UML

DTD and XML streams

Warehouse

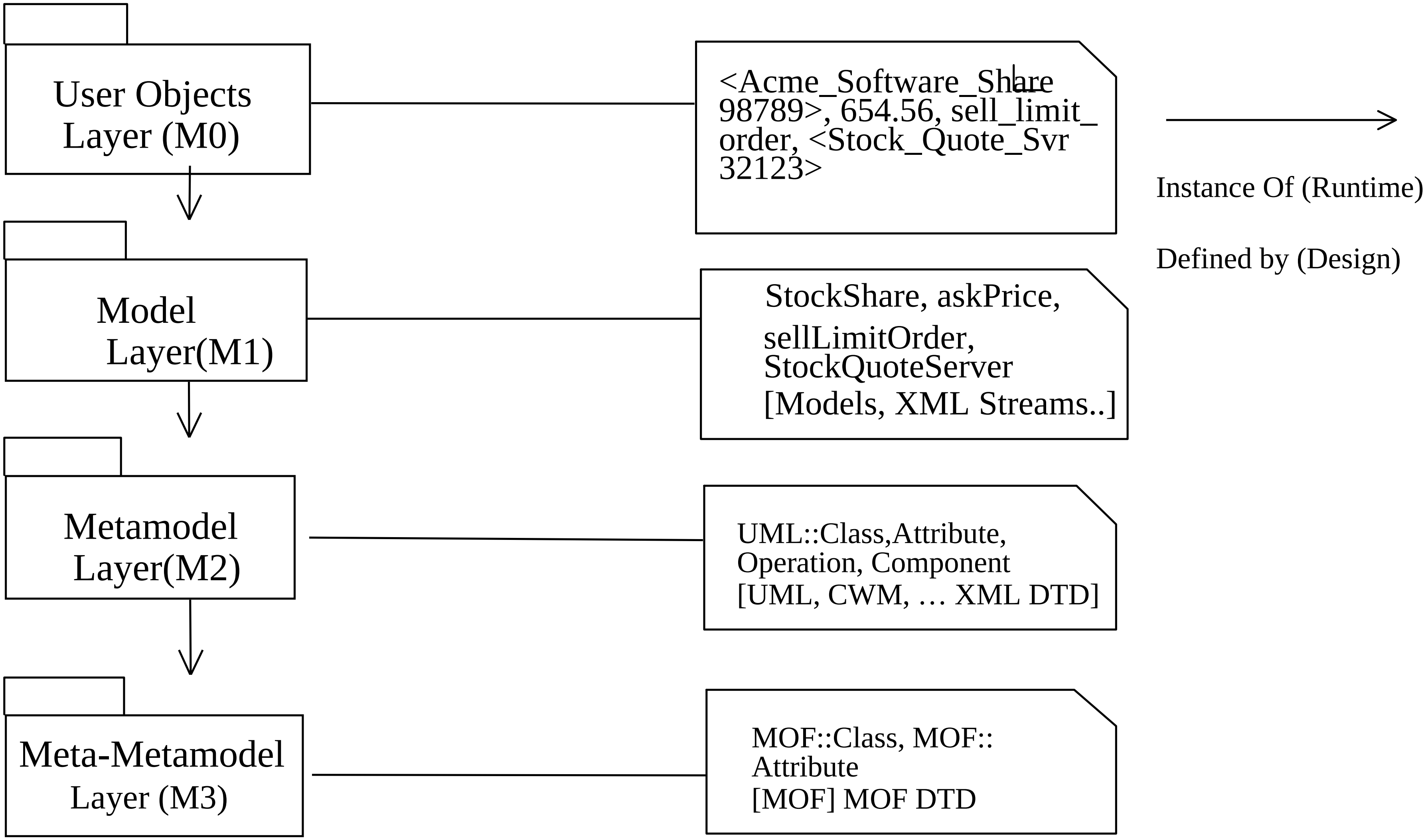
DTD and XML streams

# OMG Metamodeling Architecture and XMI

| | | | |
|---|---|---|---|
| **M3** | **MOF MetaMetaModel** | **MOF as XML DTD** | |
| **M2** | **UML & other MetaModels** | **UML & others as XML DTDs** | **MOF MetaModels as XML Documents** |
| **M1** | **Model** | | **UML & other Models as XML Documents** |
| **M0** | **Instances** | | |

- DTDs defined for MOF, UML
- MOF metamodel DTD generation
- Models are XML documents with a DTD
- Document and DTD interchange

# Metamodeling Architecture and XMI



User Objects Layer (M0)

<Acme_Software_Share 98789>, 654.56, sell_limit_order, <Stock_Quote_Svr 32123>

Model Layer(M1)

StockShare, askPrice,
sellLimitOrder,
StockQuoteServer
[Models, XML Streams..]

Metamodel Layer(M2)

UML::Class,Attribute,
Operation, Component
[UML, CWM, … XML DTD]

Meta-Metamodel Layer (M3)

MOF::Class, MOF::Attribute
[MOF] MOF DTD

Instance Of (Runtime)

Defined by (Design)

# Potential XMI Usage Scenarios

- Interchange of UML and other MOF compliant models between
  - Modeling and design tools, generators..
  - Between tools and repositories
  - Between repositories
- Publish design meta data on the web
  - Leverage XML/HTML infrastructure that already exists
- Use in addition to/instead of MOF based CORBA interfaces to meta objects
  - XMI allows disconnected/occasionally connected users

# Summary of mandatory RFP Requirements

| **Requirement** | **Requirement addressed** |
|---|---|
| Use MOF as metameta model | MOF is the meta-metamodel (M3) |
| Syntax and Encoding<br>Unambiguous identification | Mapping from MOF to XML specified using precise mapping rules of metamodel to DTDs.<br>Use of XML and XMI headers (including metamodel reference), XMI.extensions |
| Referenced Concepts (unambiguous id) | Specified using XLinks and XPoiners |
| UML & MOF Support | UML and MOF DTDs, Proof of concept with multiple modeling tools |
| International codesets | XMI supports Unicode via the optional encoding declaration of XML |

# Revised Submission - What’s New

- Generated DTDs for MOF and UML
- UUIDs for unique identification of metadata
- Stream Differencing
- Transfer of imcomplete models
- More complete support for datatypes
- Improved support for linking for enhanced integration with the Internet

*Some of the enhancements (eg: UUIDs and Stream differencing) were influenced by the CDIF and UOL submissions respectively.*

# Proof of Concept (prototypes)

- Unisys
  - UML and MOF DTDs for exchange of models to and from Select Enterprise and Unisys UREP UML 1.1 and MOF 1.1 repositories
  - Auto generation of DTDs and XML streams from MOF repository
- IBM
  - UML DTD for exchange of models to and from Rational Rose
  - Auto generation of DTDs and XML streams
  - XMLstream differencing
  - Export and Import of UML from TeamConnection Repository
- Oracle
  - Export and Import of UML models from Oracle Repository

# Proof of Concept (prototypes - 2)

- DSTC
  - Auto generation and loading of XML streams from/into MOF based repository
  - Test with multiple metamodels
- Select Software
  - Export and Import of UML models from Select Enterprise
- And more from other vendors coming up soon…
- Some of these will be demonstrated at Burlingame
  - Oracle sponsored XMI reception on Wednesday evening.
- The XMI submission Appendices - the generated DTDs!
  - From IBM and Unisys

# Technology Alternatives Considered

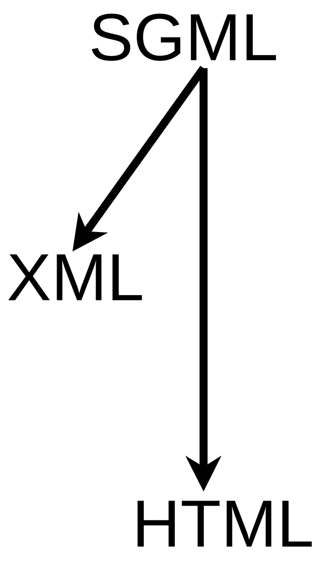
- Transfer Format
  - CDIF
  - ***XML***
  - UOL
  - Other
- Potential for integrating ideas concepts from other submissions over next few months
- Next we will cover ‘’Why XML?”

# Overview of XML and importance to OMG

# Extensible Markup Language

- XML technology
- XML example
- XML and the industry
- XML benefits
- XML and the OMG

# XML technology



- Open standard by the W3C.
- Markup language based on SGML.
- Combines data & metadata for information interchange.
- Simple, flexible, eXtensible.
- Tags form a tree information structure.
- DTD provides the tag rules.

# XML example

Document

```
<Auto>
        <Make> Ford </Make>
        <Model> Mustang
     </Model>
        <Year>  98 </Year>
        <Color> blue </Color>
        <Price> 25000 </Price>
</Auto>
```

DTD

```
<!Element Auto (Make, Model, Year, Color, Price)>
```

# XML and the Industry

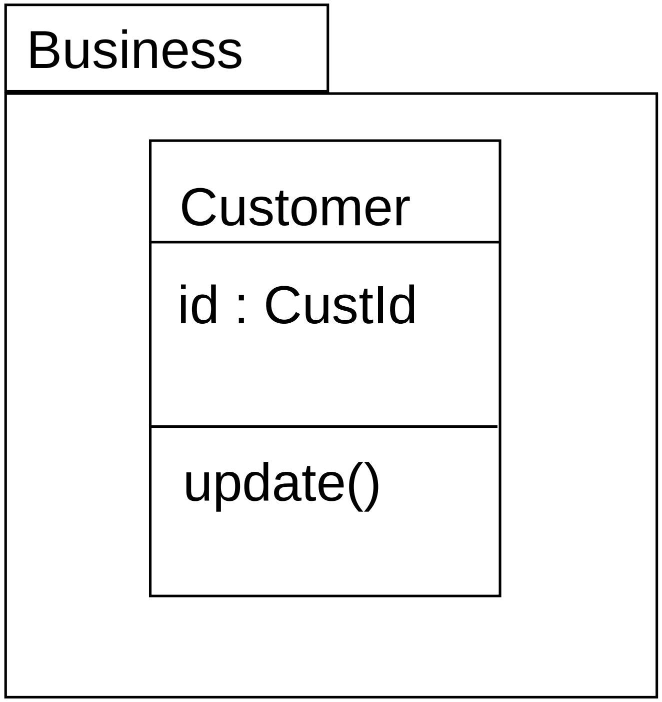
- Standards
  - W3C open standard on Feb 10, 1998.
  - International ISO character sets
  - Additional standards in progress:
    - XLink/XPointer, Namespaces, XSL, RDF, DOM, SAX, Web-DAV
- Support is exploding
  - 27 books on Amazon.com in < 1 year
  - XML supported by Adobe, ArborText, DSTC, HP, IBM, Microsoft, Netscape, Oracle, Platinum, Unisys, Select, Sun, Xerox
  - Web, publishing, repositories, modeling, databases/warehouses, services, financial, health care, semiconductors, ...

# XML benefits

- XML is system-independent, vendor independent, proven with HTML on the web.
- Metadata delivery via the web
- Validation, tool support, low cost of entry
- Advanced linking
- Stylesheets for views, transforms

# XMI drill down with examples
# MOF-XML mapping rule sample

# Abbreviated Example (UML)



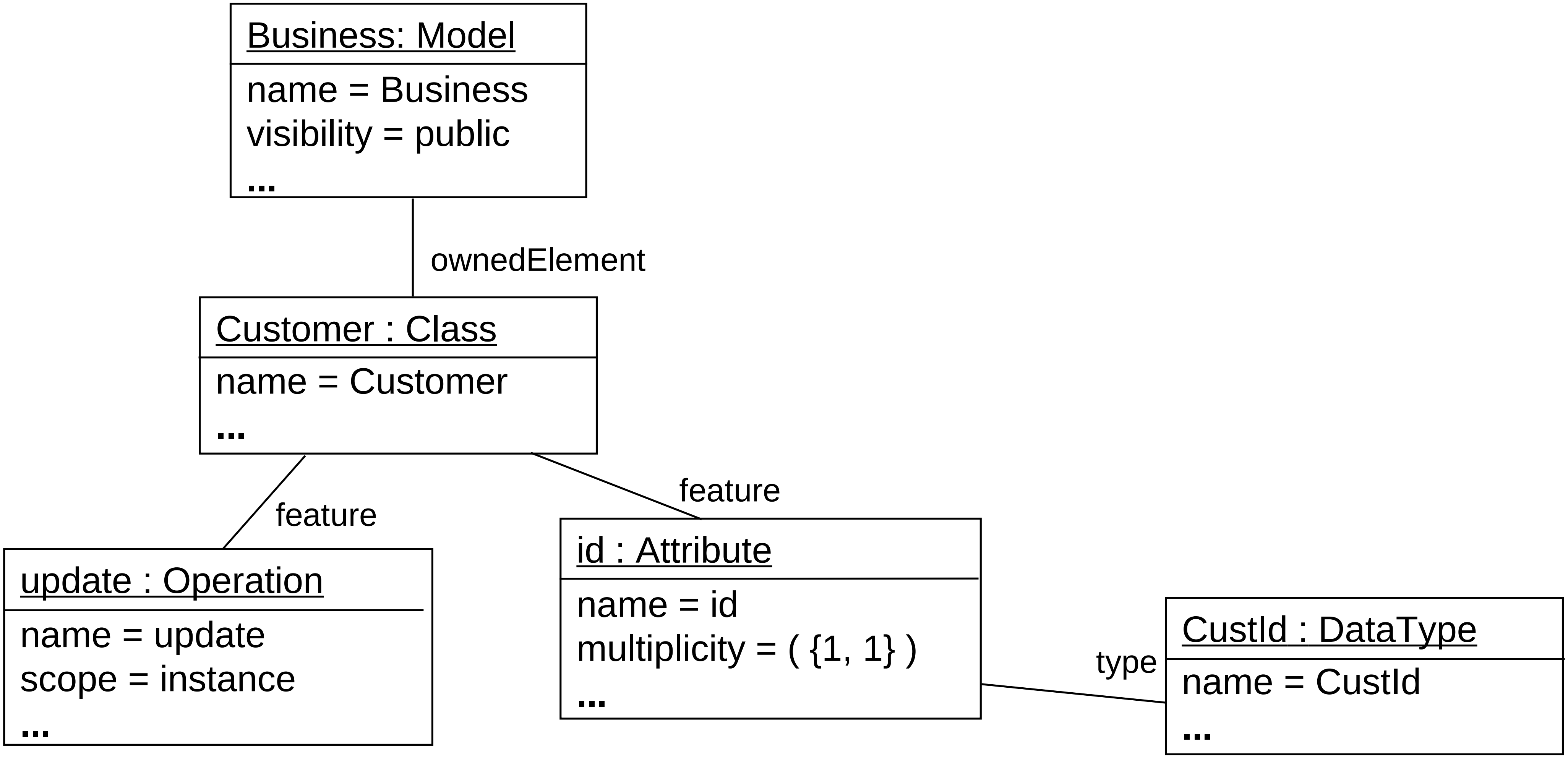
Business

Customer

id : CustId

update()

# As a metamodel instance



Business: Model

name = Business
visibility = public
...

ownedElement

Customer : Class

name = Customer
...

feature

update : Operation

name = update
scope = instance
...

feature

id : Attribute

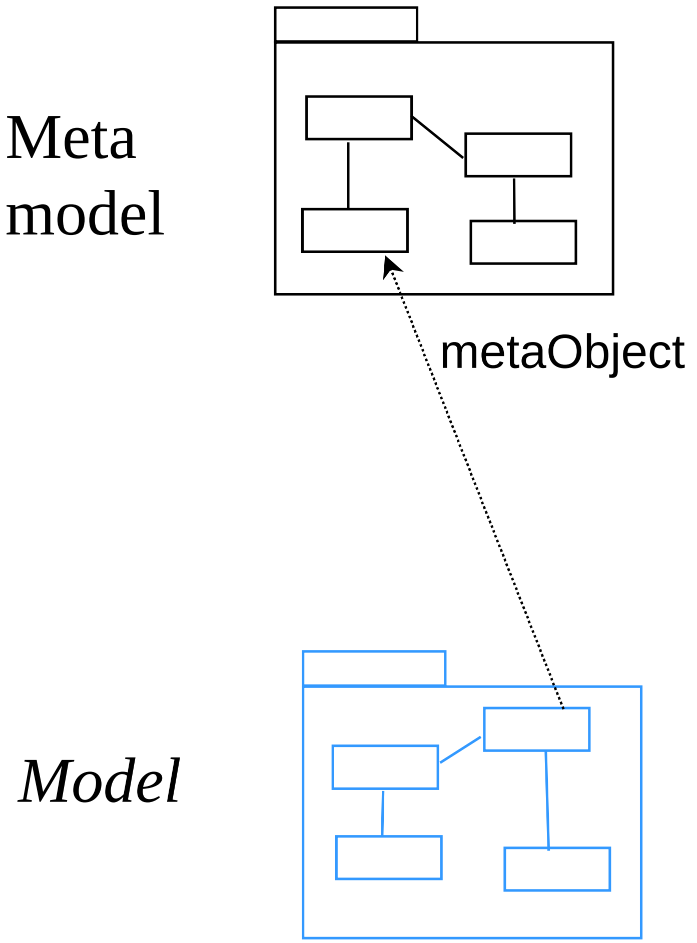
name = id
multiplicity = ( {1, 1} )
...

type

CustId : DataType

name = CustId
...

# Tags from metamodel, content from model



```xml
<Model>
    <name>Business</name>
    <visibility xmi.value="public"/>
    <Class>
        <name>Customer</name>
        <feature>
            <Attribute>
                <name>id</name>
                <multiplicity>
                    <XMI.field>1</ XMI. field>
                    < XMI. field>1</ XMI. field>
                </multiplicity>
            </Attribute>
        </feature>
    </Class>
</Model>
```

# As an XML Document Fragment (Simplified)

```
<!-- Document Prologue, etc.   -->
<Model xmi.id="a1"> <name>Business</name><visibility xmi.value="public"/>
   <ownedElement>
      <Class xmi. id="a7"><name>Customer</name>
         <feature>
            <Attribute><name>id</name>
               <multiplicity><XMI.field>1</ XMI.field>
                                  < XMI.field>1</ XMI.field></multiplicity>
               <type>< DataType href=”|a247"/></type>
            </Attribute>
            <Operation><name>update</name>
               <scope xmi.value="instance"/>
            </Operation>
         </feature>
      </Class>
   </ownedElement>
</Model>
```

# As an XML Document Fragment

(Prefixes have been removed for readability)

```
<!-- Document Prologue, etc.   -->
<Model XMI.id="a1"> <name>Business</name><visibility XMI. value="public"/>
    <isRoot XMI.value = "false"/><isLeaf XMI.value="false"/><isAbstract XMI.value="false"/>
    <ownedElement>
        <Class XMI. id="a7"><name>Customer</name><visibility XMI. value="public"/>
          <isRoot XMI.value = "false"/><isLeaf XMI.value="false"/><isAbstract XMI.value="false"/>
          <isActive XMI.value="false"/>
            <feature>
                <Attribute><name>id</name><visibility XMI.value="public"/>
                     <ownerScope XMI.value="instance">
                         <multiplicity><field>1</field><field>1</field></multiplicity>
                      <changeable XMI.value="true"/><targetScope XMI.value="instance"/>
                    <type>< XMI.reference href=”a247"/></type>
                </Attribute>
                <Operation><name>update</name><visibility XMI.value="public"/>
                    <ownerScope XMI.value="instance"/><isQuery XMI.value="false"/>
                 <specification/><isPolymorphic XMI.value="false"/><concurrency XMI.value="guarded"/>
                </Operation>
            </feature>
        </Class>
    </ownedElement>
</Model>
```

# XML Production Rules

- EBNF Grammar
- Production Defined as OCL Query
- Query input:
  - a RefObject (an object whose metaclass is defined via MOF elements)
- Returns XML Document

# Example EBNF Production Rule

**ObjectAsElement ::=**

< ***class-name*** **xmi.id="** **IdOfObject** **">**

**ObjectContents**

**</** ***class-name*** **>**

# Example OCL Production Rule

**ObjectAsElement(obj : RefObject) : Sequence(string)**

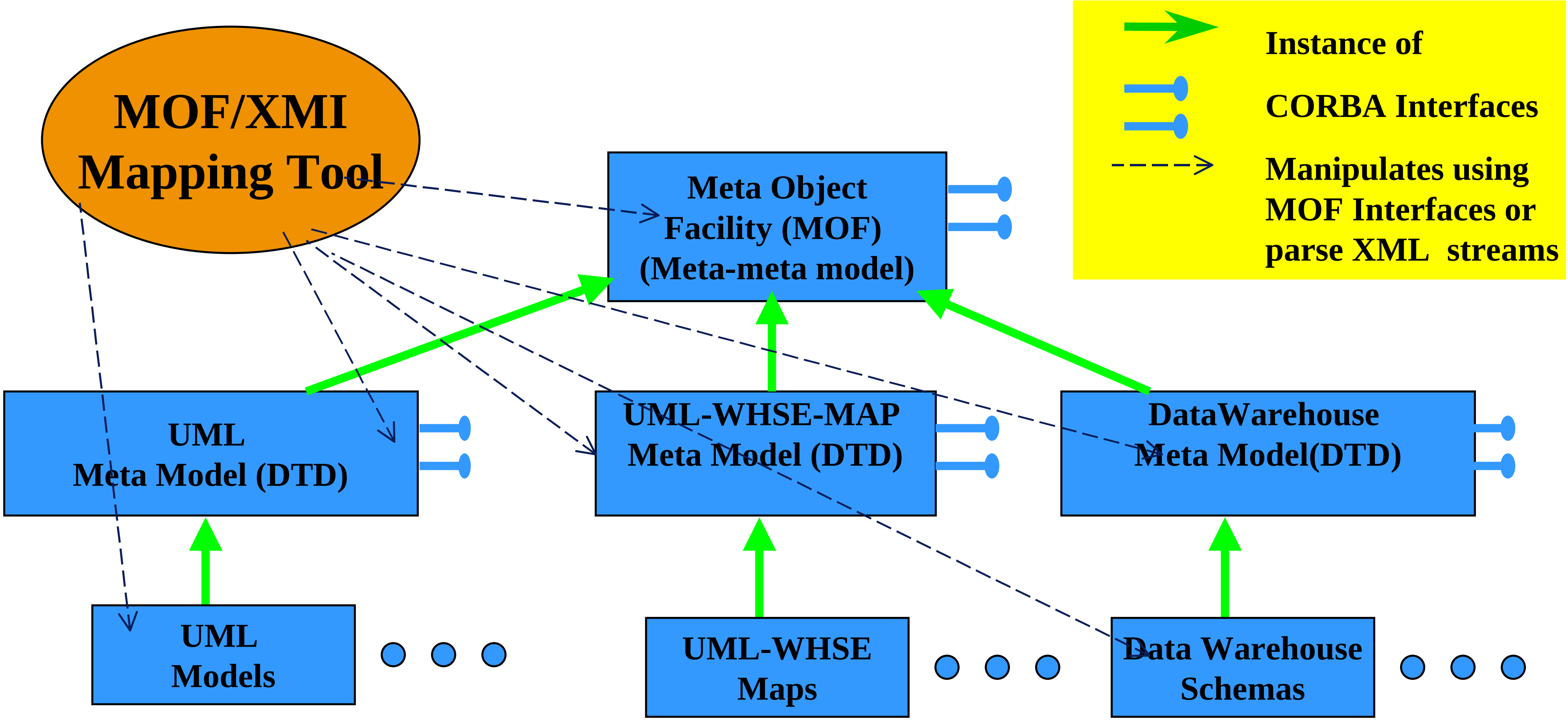
```
ObjectAsElement(obj) =
Sequence{ ’<’,
  DotNotation(obj.metaObject().oclAsType(Class).qualifiedName),
  ’ xmi.id="’,
  IdOfObject(obj),
  ’">’,
  ObjectContents(obj.metaObject().oclAsType(Class), obj),
  ’</’,
  DotNotation(obj.metaObject().oclAsType(Class).qualifiedName),
  ’>’
  }
```

# UML DTD Fragment

```
<!ELEMENT Class              (name, visibility, isRoot, isLeaf, isAbstract, isActive,
                             XMI.extension*,
                             constraint*, requirement*, provision*, stereotype*,
                             elementReference*, collaboration*, partition?,
                             template?, view*, presentation*,
                             namespace?, behavior*, binding?
                             implementation*, generalization*, specialization*,
                             parameter*, structuralFeature*, specification*,
                             associationEnd*, participant*, createAction*,
                             instance*, classifierRole*, realization*,
                             classifierInState*, taggedValue*, ownedElement*,
                             feature*)?>
<!ATTLIST Class XMI.element.att; XMI.link.att;>
<!ELEMENT name (#PCDATA | XMI.reference)*>
<!ELEMENT feature (Feature| StructuralFeature| Attribute| BehavioralFeature| Operation| Method
    Reception)*>
...
```

# XMI Usage Scenario

## (or why do we go thru all these hoops!)



- ➡ **Instance of**
- ⊸ **CORBA Interfaces**
- ⇢ **Manipulates using MOF Interfaces or parse XML streams**

**MOF/XMI Mapping Tool**

**Meta Object Facility (MOF) (Meta-meta model)**

**UML Meta Model (DTD)**

**UML-WHSE-MAP Meta Model (DTD)**

**DataWarehouse Meta Model(DTD)**

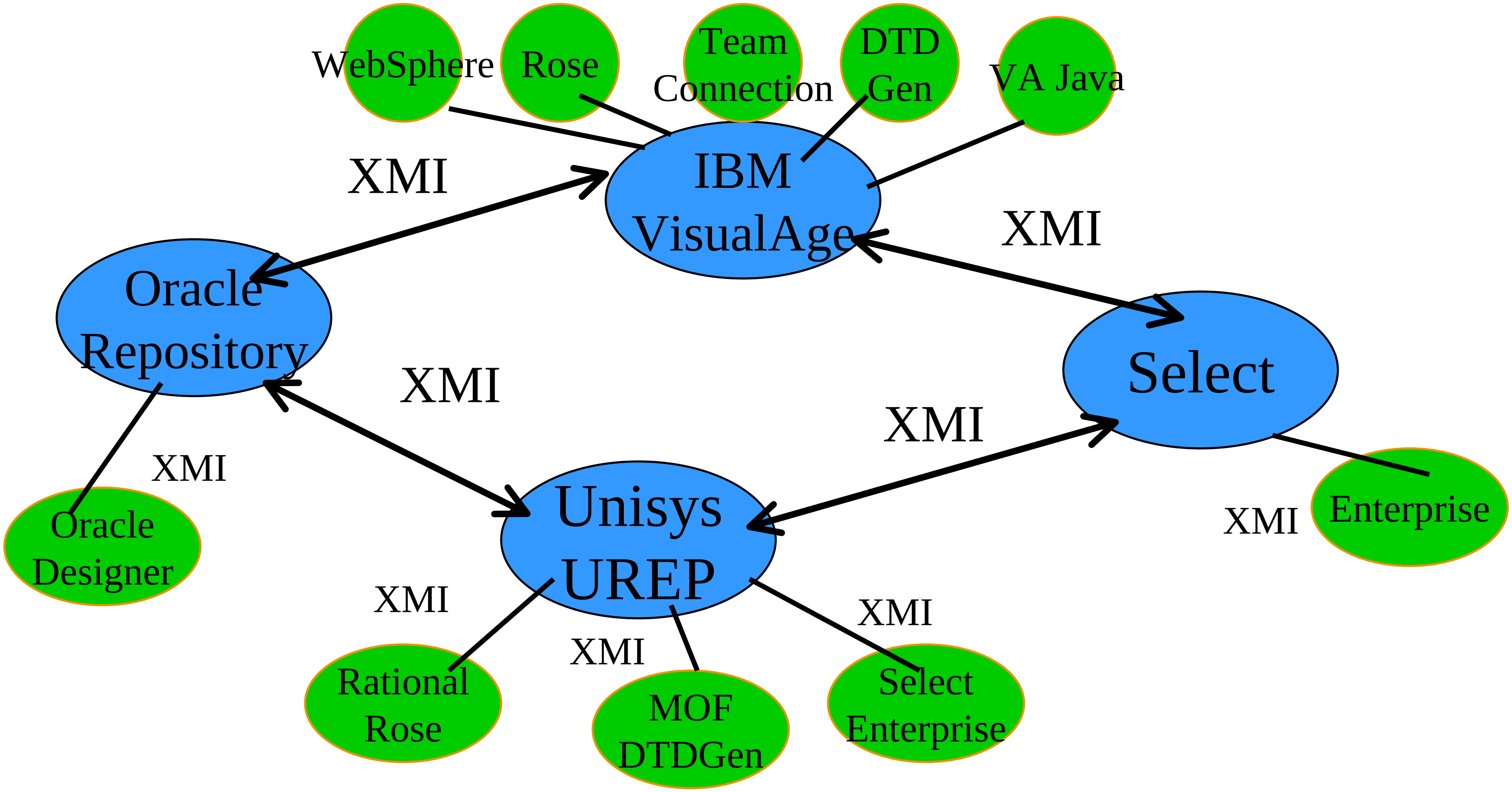
**UML Models** ● ● ●

**UML-WHSE Maps** ● ● ●

**Data Warehouse Schemas** ● ● ●

# XMI Proof of Concept

## Visit Oracle Reception at 6 PM



# Summary: XMI proposal and the OMG

- Synergy with OMG metamodeling architecture
  - DTD <=> Metamodel
  - Document <=> Model
- OMG leverage of XML:
  - Standardize key metamodels and convenient transfer format for UML, MOF and more
  - Create industry standard metamodels/DTDs
  - Leadership for standardizing object-oriented information interchange of the future
- OMG CORBA interchange - XML interchange for metadata

# For more information

- Stay tuned to future meetings
  - Comments to : xmi-feedback@omg.org
  - TC and AB vote in January, BOD vote in March
- E-mail
  - sridhar.iyengar2@unisys.com
  - sbrodsky@us.ibm.com
- XML Information
  - www.w3c.org
- UML and MOF Information
  - www.omg.org

# Questions & Comments