

# Model-Driven Architecture

## Using OMG and Other Industry Standards for Scalable Software Development

David S. Frankel

Chief Consulting Architect, Advanced Technologies  
IONA Technologies, Inc.

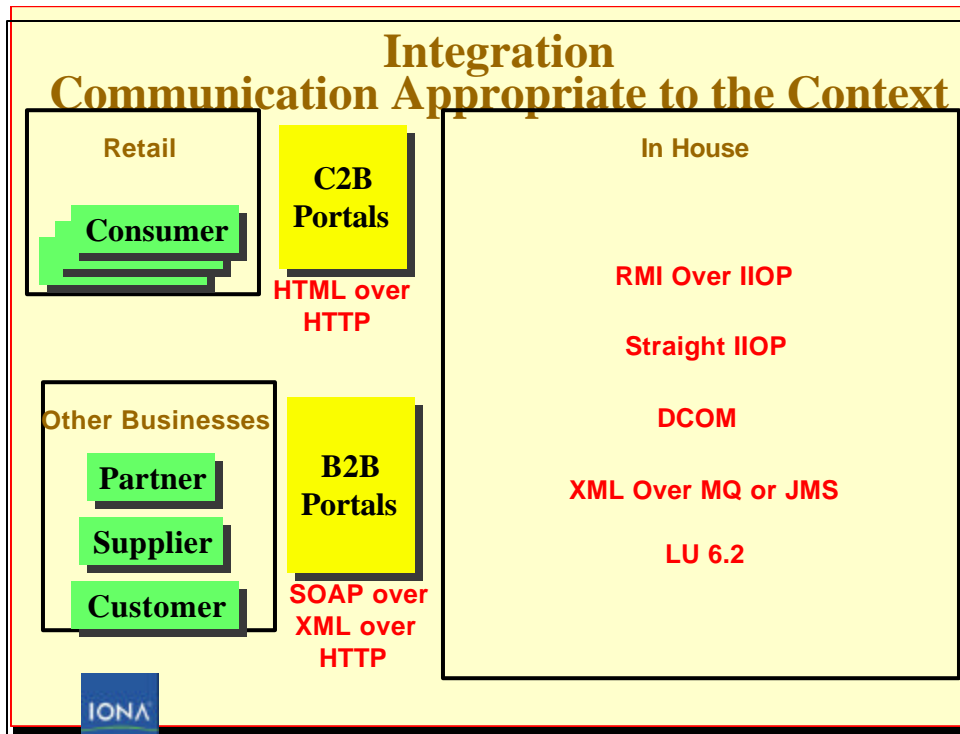
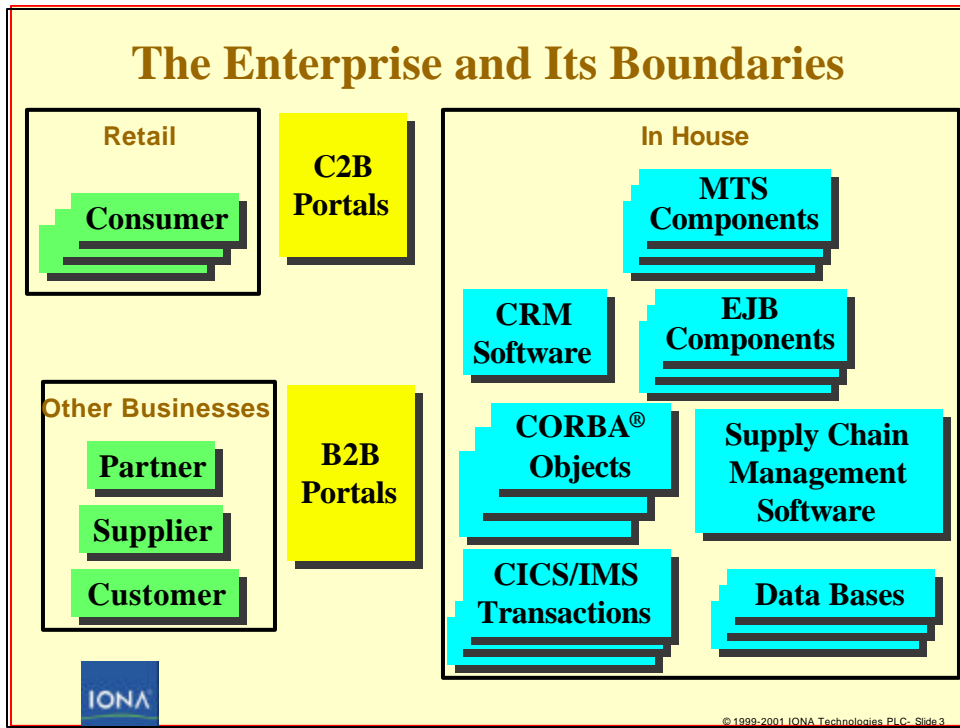
OMG Architecture Board Member  
[david.frankel@iona.com](mailto:david.frankel@iona.com)



## Topics

- Covered
  - *Separating Business Logic from Platform Concerns*
  - Formal modeling
  - Relevant Industry Standards Activities
  - Meta Object Facility (MOF)





## Technology Selection for New Components to be Developed

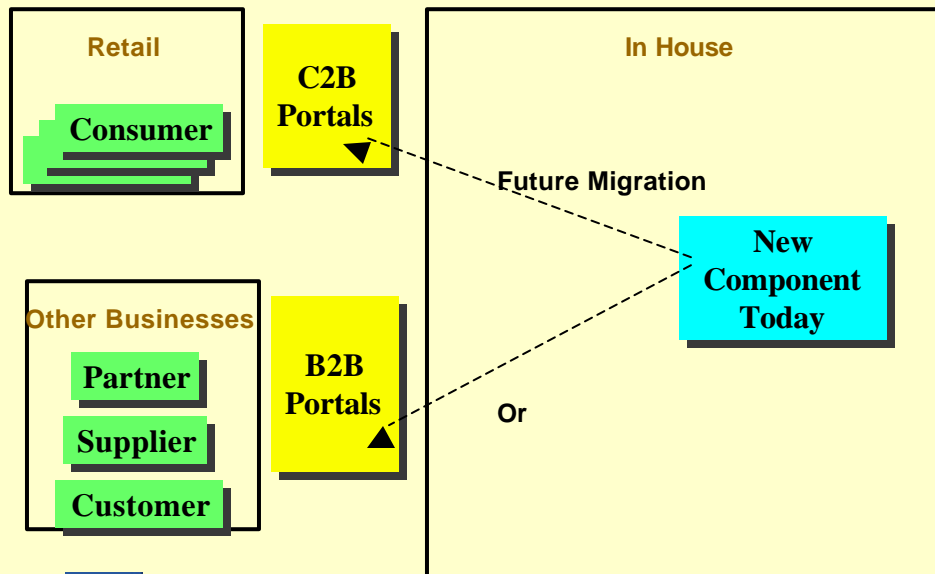
- For Components That Will Run In-House
  - EJB
  - CORBA™
  - MTS/COM+
  - MQ Series
  - MQ Series / XML
  - JMS / XML
  - CICS / IMS
- B2B and C2B Components
  - HTML / ASP / JSP
  - XML
  - SOAP / XML

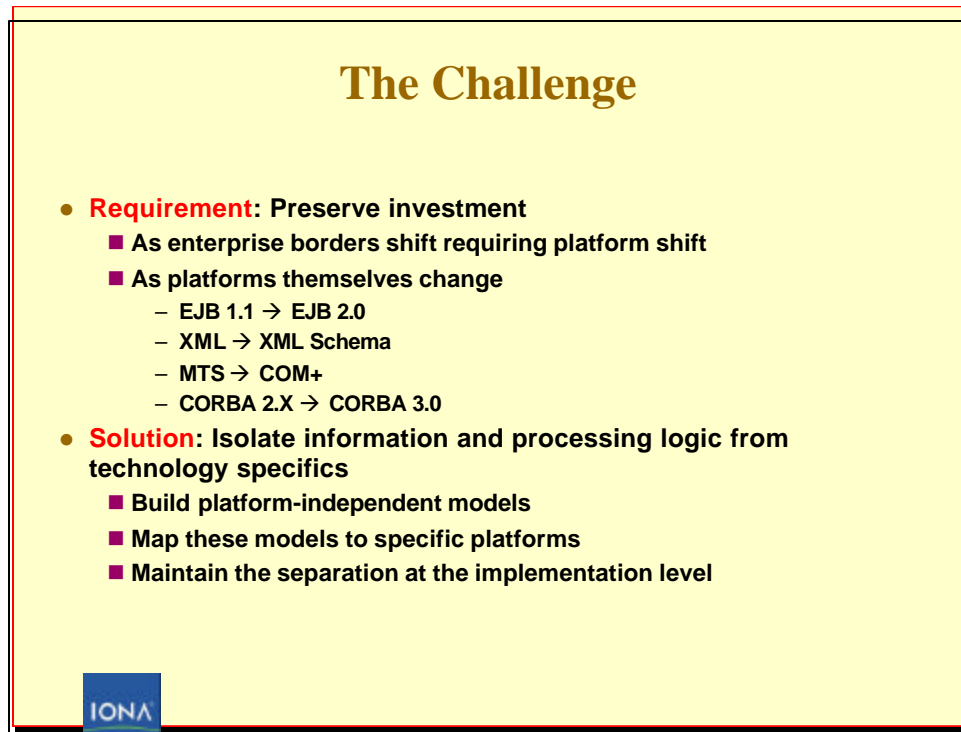
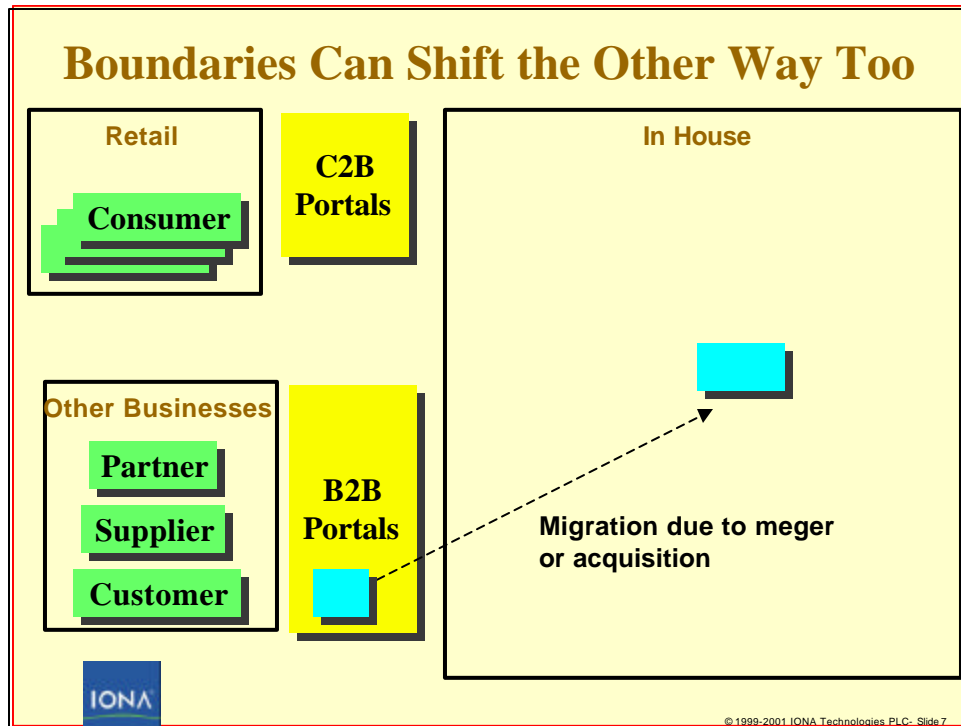
Seems fine, but there's a problem...



© 1999-2001 IONA Technologies PLC. Slide 5

## Shifting Boundaries!!!





## Topics

- **Covered**
  - **Separating Business Logic from Platform Concerns**
  - ***Formal modeling***
  - **Relevant Industry Standards Activities**
  - **Meta Object Facility (MOF)**



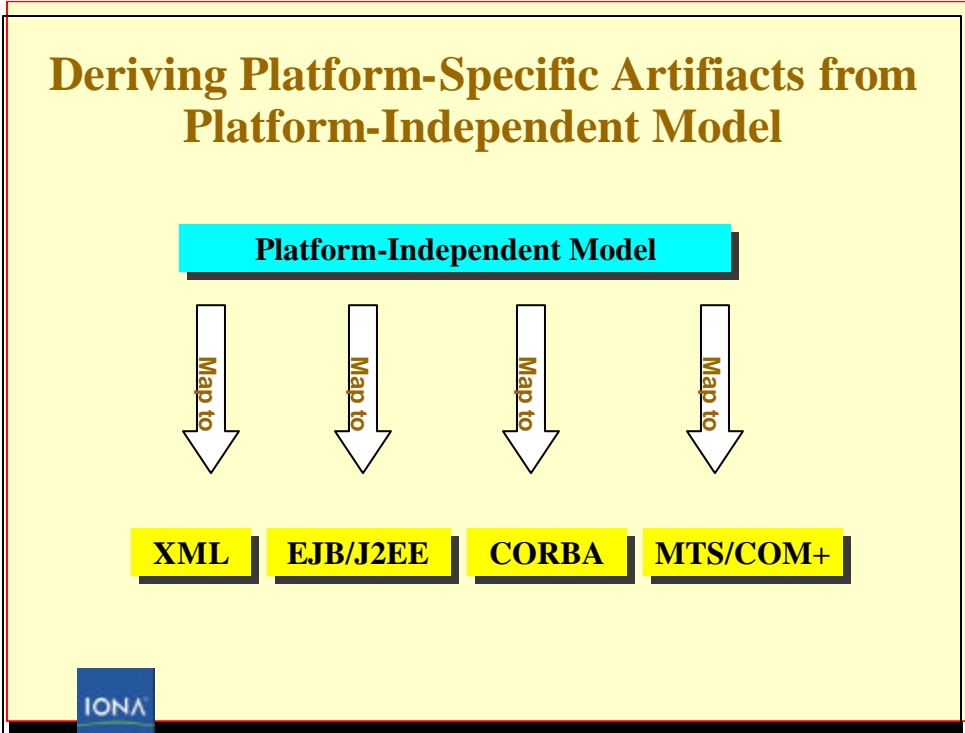
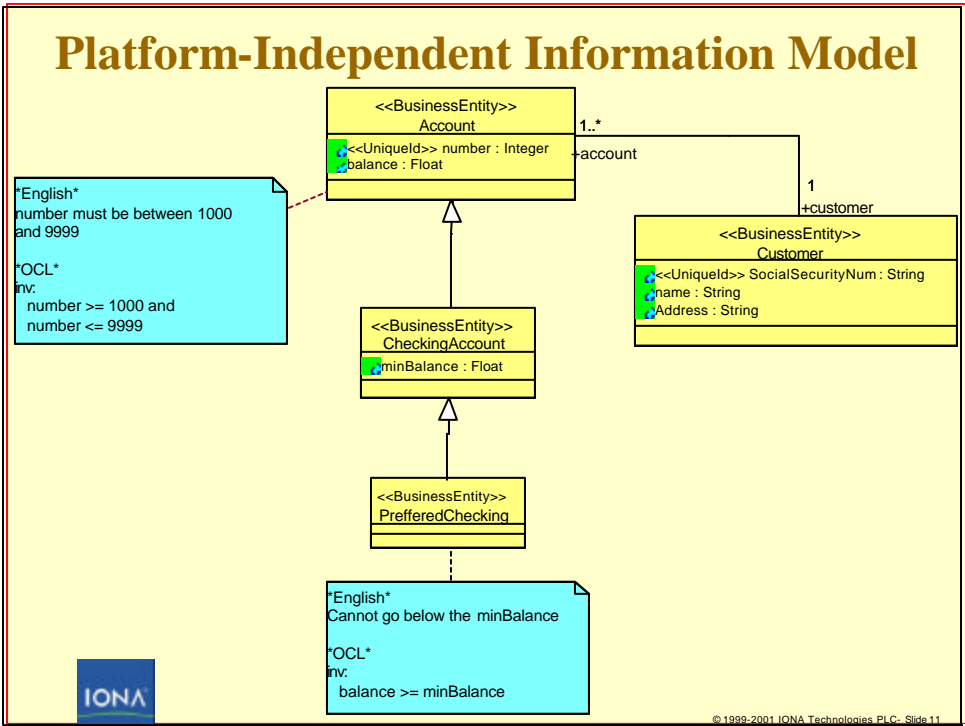
© 1999-2001 IONA Technologies PLC. Slide 9

## Unified Modeling Language™

- **UML™ is independent of**
  - **CORBA**
  - **COM**
  - **EJB**
  - **XML**
  - **Etc.**

**Unified Modeling Language and UML are registered trademarks of the Object Management Group (OMG)**





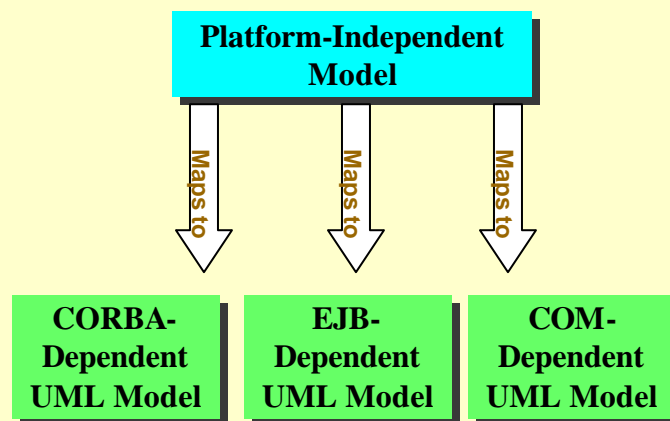
## A Problem

- CORBA, EJB, and COM declarative languages (IDL, Java Interface, MIDL) are very thin semantically
  - Can't make distinctions between
    - 0..1 and 1
    - 0..\* and 1..\*
  - Invariants, pre and post conditions can't be expressed

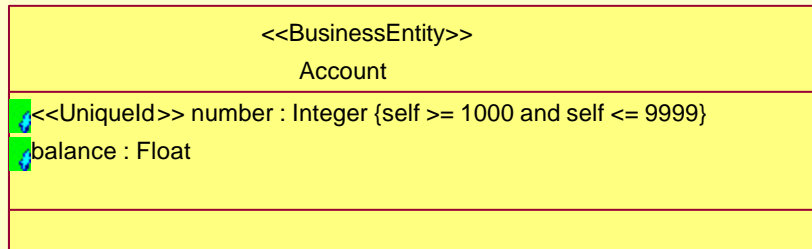


© 1999-2001 IONA Technologies PLC. Slide 13

## Solution: Use UML for Platform-Specific Specifications Too



## Example Fragment of a Platform-Independent Model



© 1999-2001 IONA Technologies PLC. Slide 15

## CORBA Example Not Using CORBA-Specific Design Model

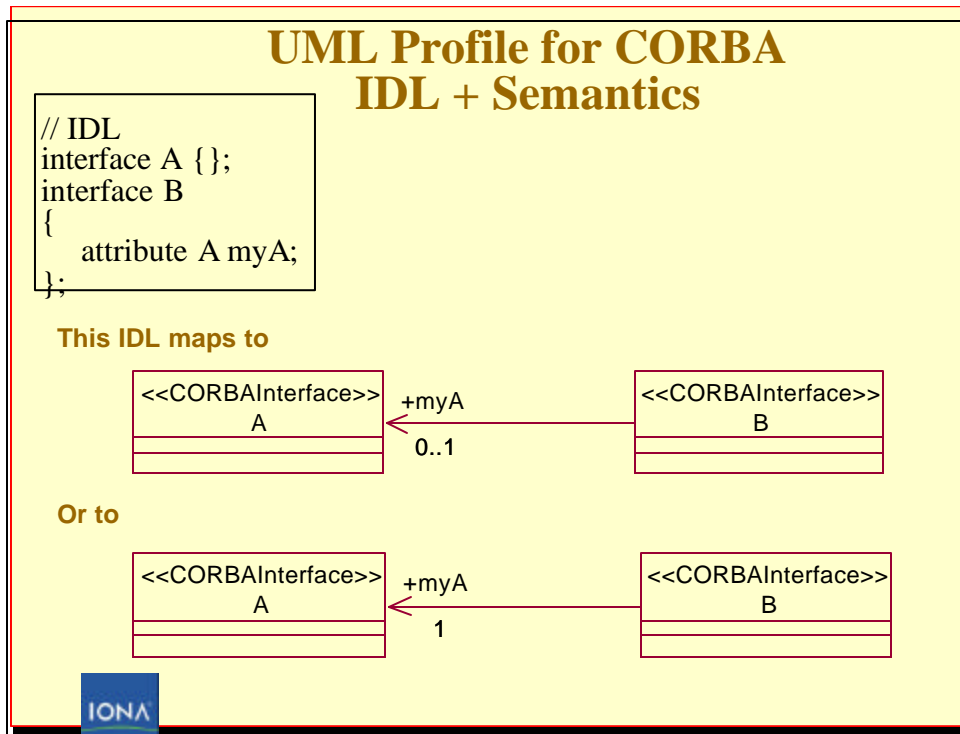
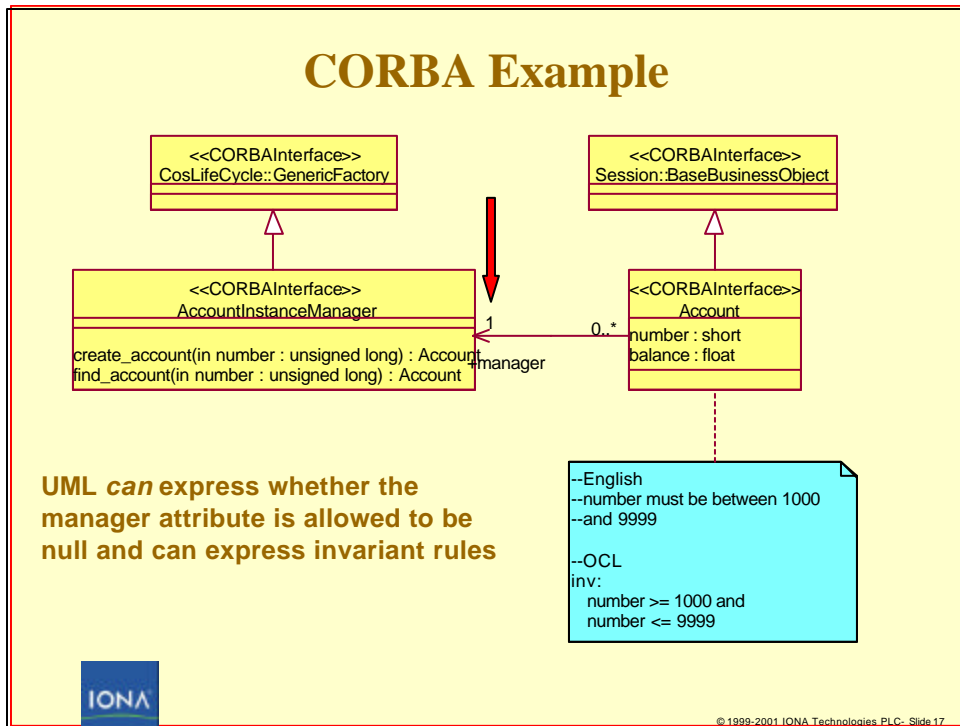
```

// Assuming that our bank model was contained within a UML Package entitled "Bank"
module Bank
{
  interface AccountInstanceManager : CosLifecycle::GenericFactory
  {
    Account create_account(in unsigned short number);
    Account find_account(in unsigned short number);
  };

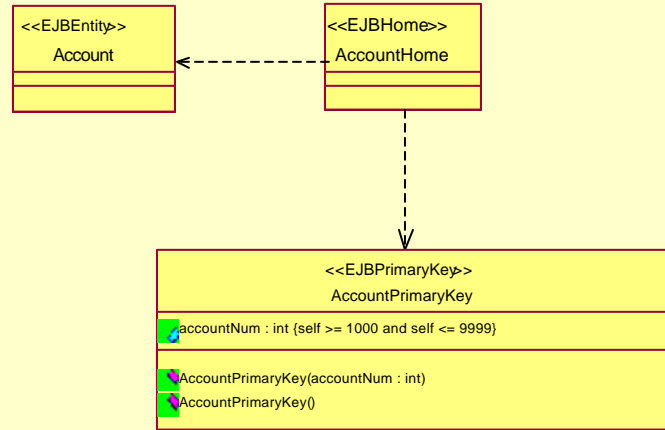
  interface Account : Session::BaseBusinessObject
  {
    attribute AccountInstanceManager manager; ←
    attribute short number;
    attribute float balance;
  };
};
  
```

**IDL cannot express whether the manager  
attribute is allowed to be null**



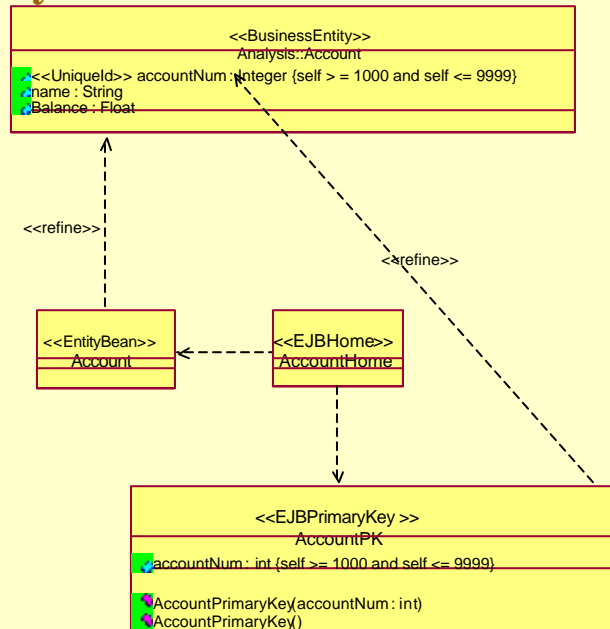


## Example Fragment of the EJB-Dependent Model

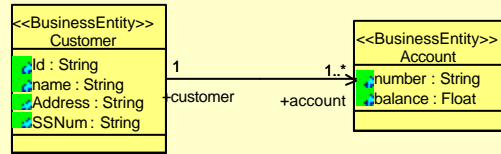


© 1999-2001 IONA Technologies PLC. Slide 19

## Tracability Between Elements of the Models



## Mapping of Attributes and Associations

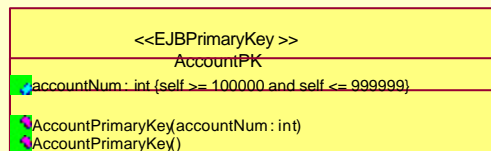


- An association expressed in a platform-independent model implies a set of CRUD (create, read, update, delete) operations on Customer
  - Signatures
  - Pre & post conditions
  - Mappings provided in works of Haim Kilov (and MOF)



© 1999-2001 IONA Technologies PLC. Slide 21

## Round Trip Engineering



```

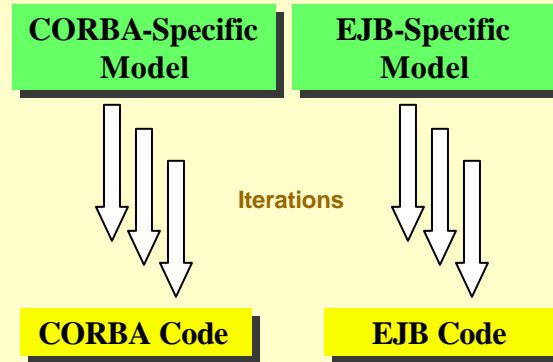
public class AccountPK implements java.io.Serializable {
    public int accountNum;
    public AccountPK(int accountNum) {

        //+Programmers add validation code here

        //+End programmer-inserted validation code
        this.accountNum = accountNum;
    }
    public AccountPK() { }
}
  
```

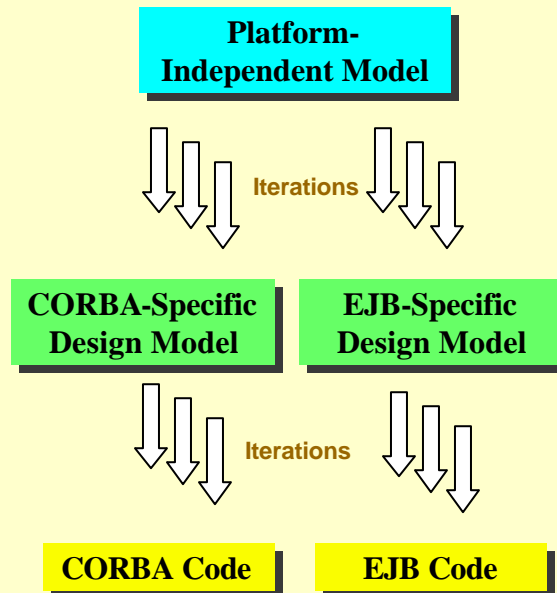


## Bi-Level RoundTrip Engineering



© 1999-2001 IONA Technologies PLC. Slide 23

## Tri-Level Round Trip Engineering



## XML Mapping Can be Automated OMG Standard Mapping: XMI™

**Platform-Independent Model**

XMI's UML-XML  
Mapping Rules  
Produce

**XML DTD**

```

classDiagram
    class BusinessEntity {
        <<BusinessEntity>>
    }
    class Customer {
        <<UniqueId>> SocialSecurityNum : String
        name : String
        Address : String
    }
    BusinessEntity <|-- Customer
    
```

```

...
<!ELEMENT Bank.Customer.SocialSecurityNum (#PCDATA |
XML.reference)*>
<!ELEMENT Bank.Customer.name (#PCDATA | XML.reference)* >
<!ELEMENT Bank.Customer.Address (#PCDATA | XML.reference)* >

<!ELEMENT Bank.Customer.account (Bank.Account |
    Bank.SavingsAccount |
    Bank.CheckingAccount |
    Bank.PrefferedChecking |
    Bank.RegularChecking)*
>
<!ELEMENT Bank.Customer (Bank.Customer.SocialSecurityNum?,
    Bank.Customer.name?,
    Bank.Customer.Address?,
    XML.extension*,
    Bank.Customer.account*)?
...
    
```

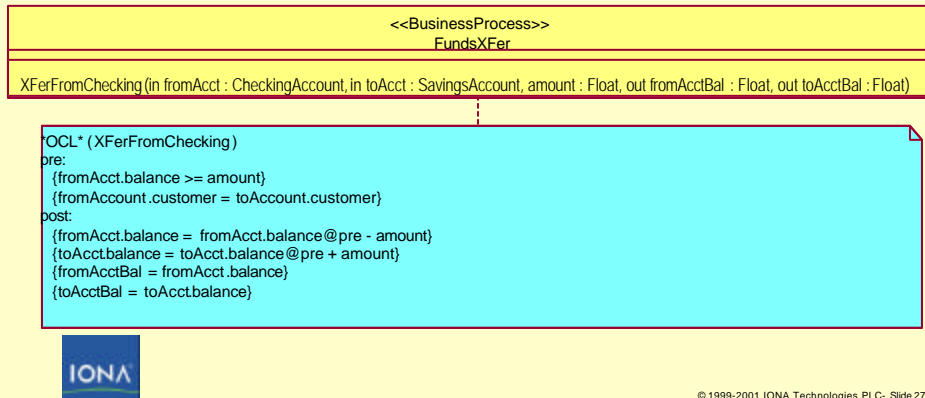
© 1999-2001 IONA Technologies PLC. Slide 25

## XMI DTD Generator Available

- Free IBM Alphaworks XMI Toolkit
  - <http://www.alphaworks.ibm.com/tech/xmitoolkit>

## What About Information Processing?

- Use UML to model in a platform-independent fashion
- Make full use of UML's ability to express semantics!
- Map to one or more of the following:
  - EJB Session Bean
  - SOAP operation
  - CORBA operation
  - XML over MQ Series or JMS



© 1999-2001 IONA Technologies PLC. Slide 27

## Topics

- Covered
  - Separating Business Logic from Platform Concerns
  - Formal modeling
  - *Relevant Industry Standards Activities*
  - Meta Object Facility (MOF)

## Relevant Standards In Progress

- UML Profile for Enterprise Distributed Object Computing (EDOC)
  - Standard UML profile for platform-independent business component models
- UML Profile for Event Driven Architectures in EAI
  - Standards for how to use UML to model event-driven EAI in a platform-independent fashion
  - Integrated with UML Profile for EDOC, which also defines standards for modeling business processes in a detailed fashion
  - Should be ready July, 2001
- UML Profile for CORBA: Currently in FTF
  - IONA a co-submitter
- UML Profile for EJB
  - Sun JCP JSR #26
  - Just releast
  - IONA on panel of experts and a substantive contributor



© 1999-2001 IONA Technologies PLC. Slide 29

## Model-Centric Architecture and Other Key eCommerce Standards

- SOAP
  - The information payload can be XMI-compliant
  - IBM doing this with many of its SOAP initiatives
- UDDI
  - Unfortunately, model-centric approach not taken for registration and discovery API payloads.
  - However, the actual services are SOAP services, and *their* payloads can be XMI-compliant.
- ebXML
  - Has adopted the model-centric approach. Domain DTDs for payloads will be generated from UML entity models. ebXML/XMI alignment work in progress.
  - Using UML activity and collaborations diagrams for modeling business processes
- Rosetta Net
  - Business processes are modeled but not with UML
  - DTDs for information payload not being generated from models



## Benefits of Model-Driven Architecture Summary

- **The platform-independent model is the repository of the key business semantics**
  - UML is much more expressive than XML, EJB, IDL, etc.
  - Invariant rules for entities
  - More explicit relationship specification
    - Strong aggregation, weak aggregation, no aggregation
    - Explicit multiplicities
  - The platform-independent model is relatively stable as platform shifts
- **Platform-specific designs can be more precise than otherwise**



© 1999-2001 IONA Technologies PLC. Slide 31

## Topics

- **Covered**
  - Separating Business Logic from Platform Concerns
  - Formal modeling
  - Relevant Industry Standards Activities
  - *Meta Object Facility (MOF)*



## The M-Word: Metadata

- All of these models become valuable corporate assets that need to be managed effectively
- Along with other kinds of models:
  - Relational db schema
  - Schema transformation rules (important for EAI)
  - Workflow models
  - Various kinds of standard and proprietary configuration and deployment descriptors
- The OMG Meta Object Facility™ provides an architecture for managing all this metadata in an integrated fashion
  - UML, XMI, and CWM (Common Warehouse Metamodel) all are based on the MOF architecture
  - MOF is built on model-centric principles

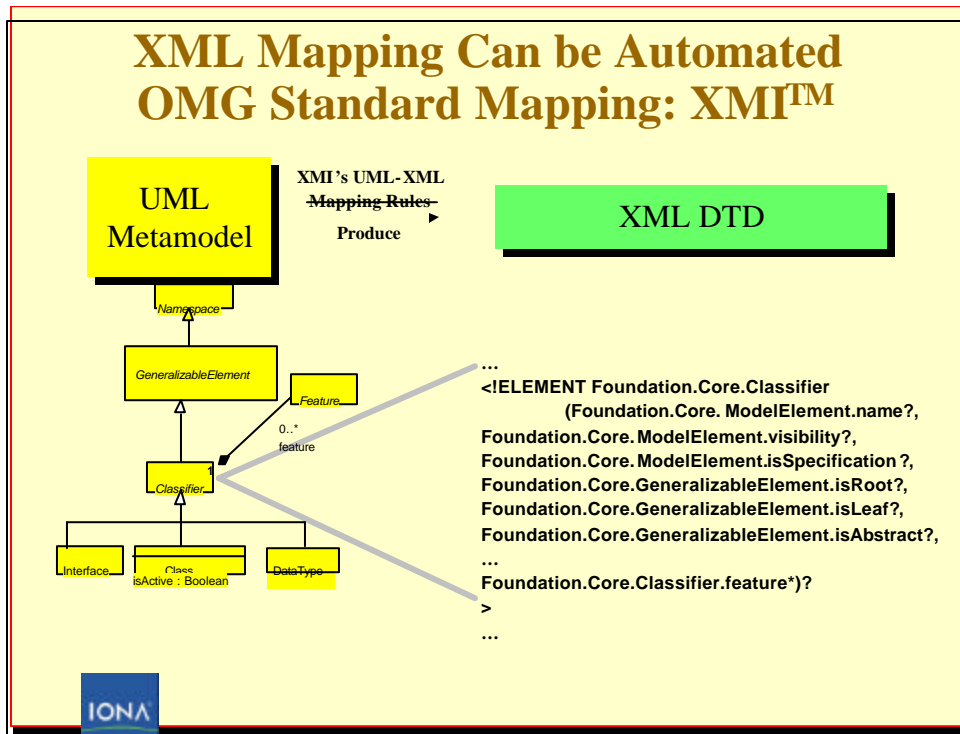
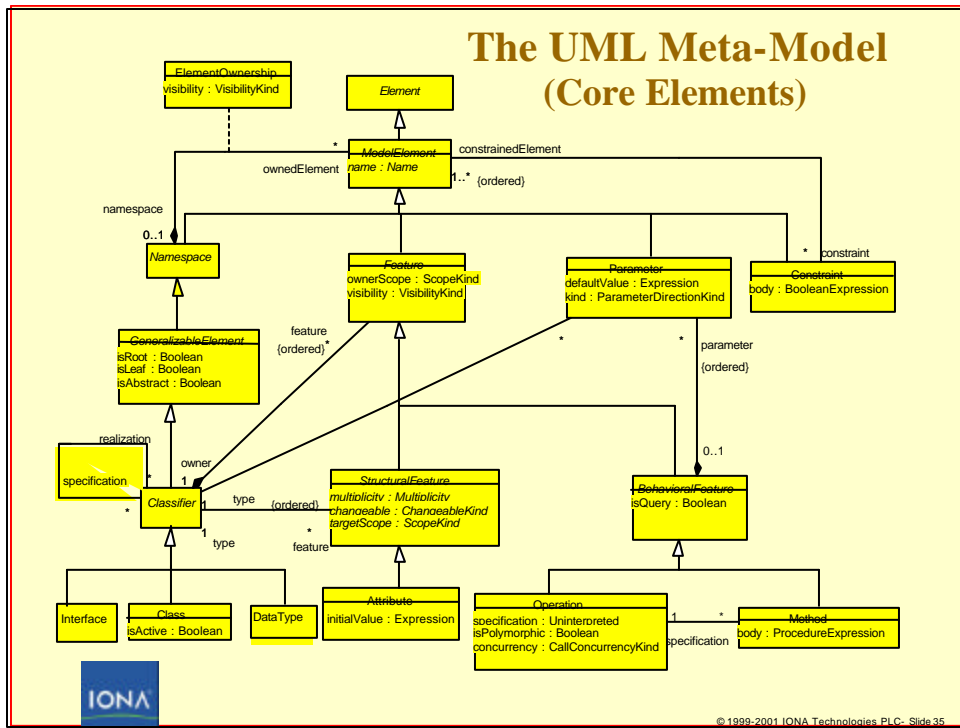


© 1999-2001 IONA Technologies PLC. Slide 33

## The OMG Metadata Architecture Meta Object Facility (MOF)

- Model-Driven
- UML used to model metadata
- XMI DTDs generated for exchanging models
- The official UML DTD was generated automatically from a UML model of UML itself (i.e. from the UML metamodel)
  - Implemented by Rational Rose, Together, and many others
  - Standard for importing and exporting Rose models
- MOF also defines a mapping to CORBA IDL
  - Given a model of a kind of metadata (i.e. a metamodel), produces IDL for a repository holding models of that kind
- Sun is defining a MOF – Java mapping (JSR #40)
  - Given a metamodel, produces Java interfaces for a repository holding models of that kind





## MOF Benefits

- There are several OMG standard kinds of models
  - UML models
  - IDL models (CORBA)
  - Data models (CWM™: Common Warehouse Metamodel)
- XMI DTDs and repository IDL automatically generated
- Tools can also generate the implementation of
  - Reading a model from an XMI stream and depositing the model in a repository
  - Reading a model from a repository and producing an XMI stream for sending
- Such tools are model-driven
  - They are driven by the model of the metadata (i.e. metamodel)
  - Separate hand-coding of the implementations for each metamodel not required
  - They can do this because they understand the mapping rules
  - XMI DTD, IDL, Java interfaces all based on the same metamodel



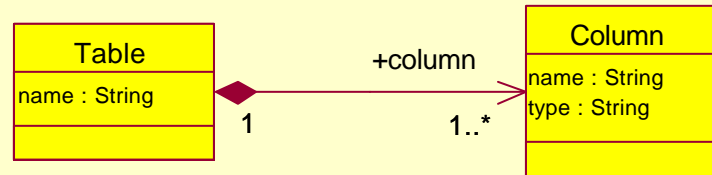
© 1999-2001 IONA Technologies PLC. Slide 37

## CWM Common Warehouse Metamodel

- Standard metamodels for
  - Relational data
  - Hierarchical data
  - OLAP
  - Data Mining
  - Schema transformation
- Submitted by
  - Dimension EDI
  - Genesis Development
  - Hyperion Solutions
  - IBM
  - NCR
  - Oracle
  - UBS
  - Unisys
- Adopted in January, 2000
- Demo of first prototypes at December, 2000 OMG meeting
- XMI DTDs standardize database schema exchange and exchange of transformation rules
- **Schema transformation huge for EAI**
  - **Between two relational schema**
  - **Between a relational schema and a hierarchical schema**
  - **Between a relational schema and an XML DTD**
  - **Etc.**



## Example Simple RDB Metamodel



© 1999-2001 IONA Technologies PLC. Slide 39

## XMI DTD Fragment Generated from the Metamodel

```

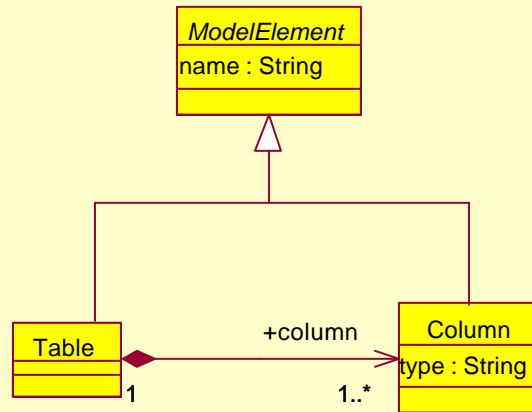
<!ELEMENT Table.column (Column)* >
<!ELEMENT Table.name (#PCDATA | XML.reference)* >
<!ELEMENT Table (Table.name?, XML.extension*, Table.column*)? >
<!ATTLIST Table
  %XML.element.att;
  %XML.link.att;
>

<!ELEMENT Column.name (#PCDATA | XML.reference)* >
<!ELEMENT Column.type (#PCDATA | XML.reference)* >
<!ELEMENT Column (Column.name?, Column.type?,
XML.extension*)? >
<!ATTLIST Column
  %XML.element.att;
  %XML.link.att;
>

<!ELEMENT RDB ((Table | Column)* >
<!ATTLIST RDB
  %XML.element.att;
  %XML.link.att;
>
  
```



## OO Models vs. Non-OO DTDs



© 1999-2001 IONA Technologies PLC. Slide 41

## Entity Declarations Avoiding Repetition in Subtypes

```

<!-- ===== RDB:ModelElement ===== -->
<!ELEMENT RDB:ModelElement.name (#PCDATA|XML.reference)*>
<!ENTITY % RDB:ModelElementFeatures 'XML.extension |
RDB:ModelElement.name'>
<!ENTITY % RDB:ModelElementAtts '%XML.element.att; %XML.link.att;
name CDATA #IMPLIED'>
<!ELEMENT RDB:ModelElement (%RDB:ModelElementFeatures;)*>
<!ATTLIST RDB:ModelElement %RDB:ModelElementAtts;>

<!-- ===== RDB:Table ===== -->
<!ENTITY % RDB:TableFeatures '%RDB:ModelElementFeatures;'>
<!ENTITY % RDB:TableAtts '%RDB:ModelElementAtts;'>
<!ELEMENT RDB:Table (%RDB:TableFeatures;)*>
<!ATTLIST RDB:Table %RDB:TableAtts;>

<!-- ===== RDB:Column ===== -->
<!ELEMENT RDB:Column.type (#PCDATA|XML.reference)*>
<!ENTITY % RDB:ColumnFeatures '%RDB:ModelElementFeatures; |
RDB:Column.type'>
<!ENTITY % RDB:ColumnAtts '%RDB:ModelElementAtts;
type CDATA #IMPLIED'>
<!ELEMENT RDB:Column (%RDB:ColumnFeatures;)*>
<!ATTLIST RDB:Column %RDB:ColumnAtts;>
  
```



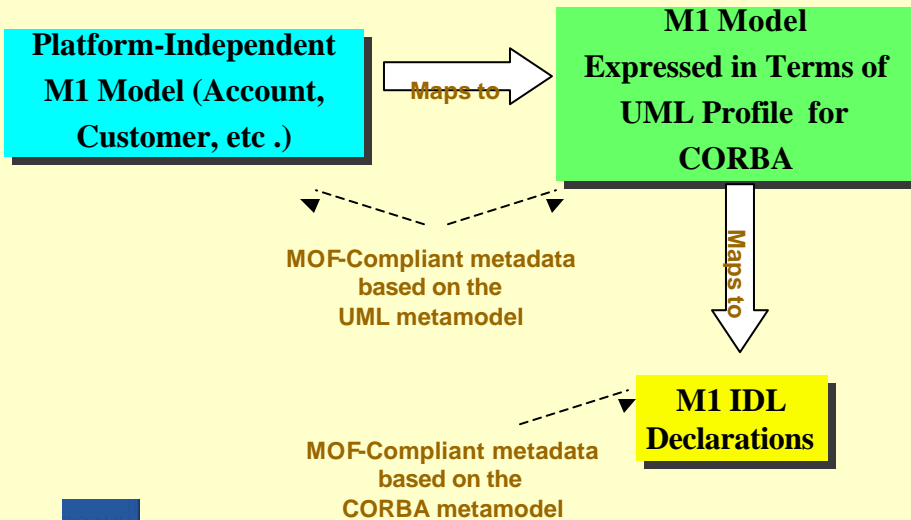
## The Four Meta-Levels Revisited Using XMI at the Instance Level

Layer	Description	Elements
<b>M3</b> meta-metamodel MOF Layer	The infrastructure for a metamodeling architecture. Defines the language for specifying metamodels.	MetaClass, MetaAttribute, MetaOperation Mof::Class, Mof::Attribute, Mof::Operation
<b>M2</b> metamodel (e.g. UML) Layer	An instance of a meta-metamodel. Defines the language for specifying a model. Consists of MetaClasses, MetaAttributes, etc.	Class, Attribute, Operation
<b>M1</b> model (User Object Model or Data Model)	An instance of a metamodel. Models a specific information domain. Consists of meta-objects (i.e. meta-data)	StockShare, askPrice, sellLimitOrder, StockQuoteServer
<b>M0</b> user objects (user data)	An instance of a model.	Acme_Software_Share_98789, 654.56, sell_limit_order, Stock_Quote_Svr_32123



© 1999-2001 IONA Technologies PLC. Slide 43

## A MOF View of Business Components



## This is an Architecture!

- **When a hot new technology arrives...such as LDAP**
  - **We don't design an LDAP schema directly**
  - **Instead, we**
    - Define a mapping from a platform-independent UML model to LDAP
    - Apply the mapping to derive our LDAP schema
    - Apply the XMI mapping to derive an XML DTD
    - Use the DTD to stream information from one directory to another

