



# **Model-Driven Architecture™ and Web Services**

David S. Frankel  
Chief Consulting Architect

# The Sisyphus Syndrome



“There is no question that if Web services are to take off as smoothly as vendors hope, a significant chunk of the more than 20 million programmers in the world will have to write to UDDI, WSDL, XML and SOAP.”

...“Web Services” InfoWorld, vol. 23, issue 11, March 12, 2001, page 39

# This Won't Scale



- Technologies for Web services are in flux
  - Several different ways of stacking them
- XML evolved to XML Schema: What's next?
- SOAP and WSDL are new and evolving
- Will port 80 be what it was intended to be when it is multiplexing myriad Web service messages?
- These technologies are bound to change
- Having Web service developers program directly to them invites rapid obsolescence
  - And is too labor intensive

# The Forest for the Trees



- The information, and the services that use and create the information, are central
- Exposing them over the Internet is the breakthrough concept
- XML, SOAP, WSDL, UDDI, etc., are important, but must be kept in perspective
  - The process of producing individual Web services must be as independent of these technologies as possible

# 1st Generation Web Service Integration



- Projecting Java, CORBA, or COM objects and components as Web services
  - Tools generate the WSDL, SOAP, XML etc.
    - and generate the 3GL code that binds them to middle/back tiers
  - Meets criteria for independence
- But Web services will be coarser grained than many of the already-existing objects and components
  - Web services as compositions of more primitive functionality
- Thus, Web services will have to be *designed*

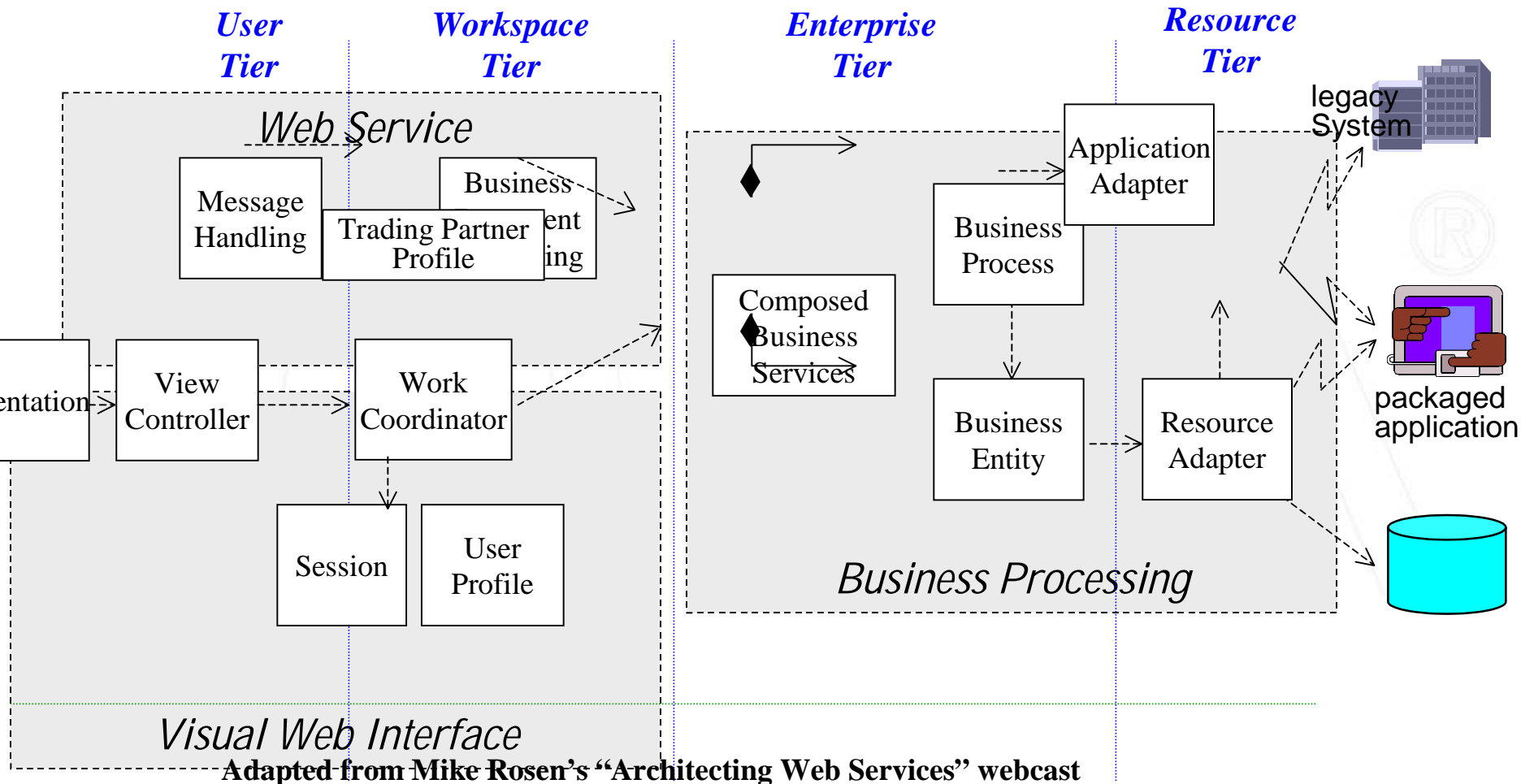
# 2nd Generation



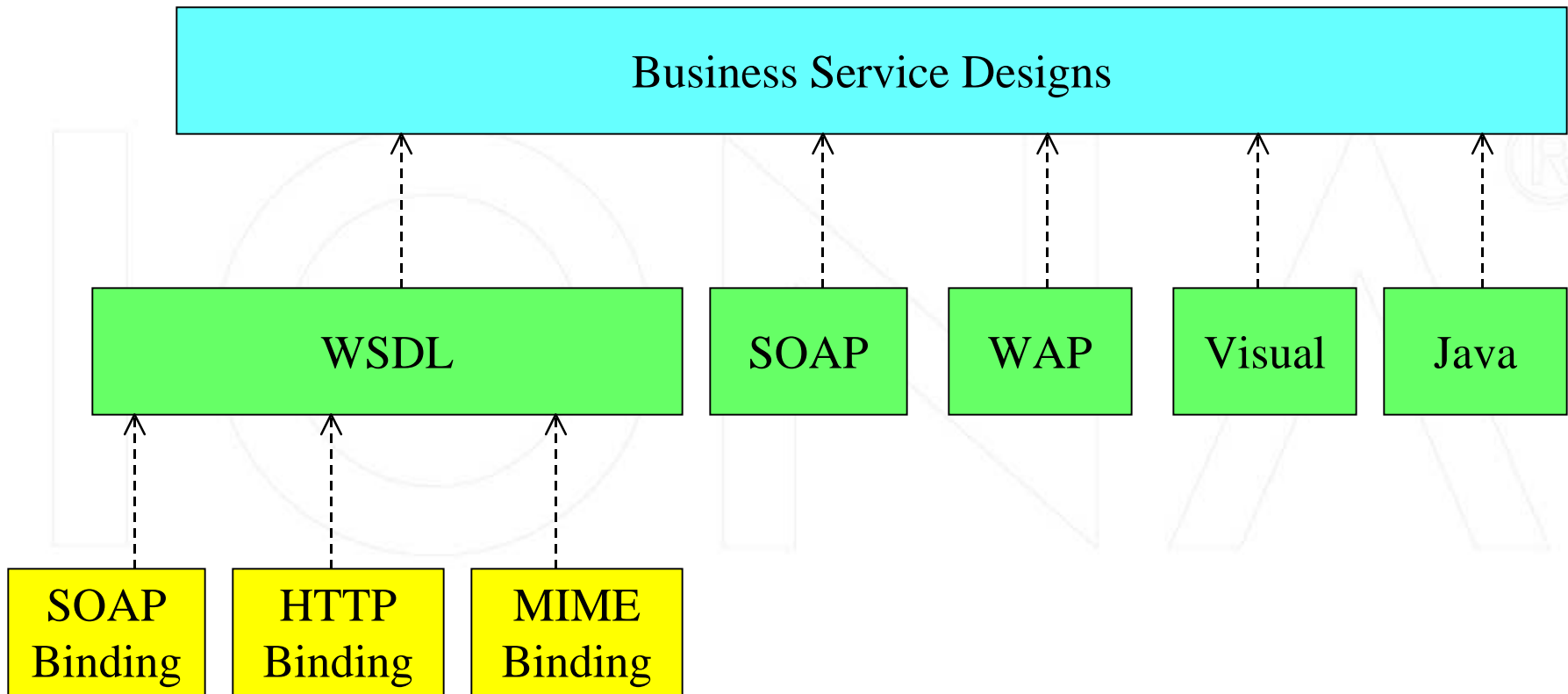
## Web Service Integration

- Web service design, driven by
  - business requirements
  - need to minimize network traffic
- Design vocabulary should provide information and service abstractions independent of XML, WSDL, SOAP, UDDI
- Trend toward tools that can automate production of XML, WSDL, SOAP, UDDI, and 3GL implementation code from the design input
  - But must support fine tuning by engineers familiar with these technologies

# Architecture



# Maximizing Reuse Over Different Technology Stacks



# Raising the Level of Abstraction



- Part of general trend
- Already well-established for front and back ends
  - WYSIWYG GUI modeling and data modeling
  - Hand coding no longer predominates
  - But tuning allowed
- Early Web applications wired Web front end directly to back end
  - Some companies avoided building intermediate tiers
  - Web services and B2B require intermediate tiers to expose coarse grained business services
  - Abstraction to allow reuse of the coarse grained business services via various technologies

# Unified Modeling Language™ (UML™)



- Standard, managed by the OMG
- Suitable for expressing information and services
- Widely accepted
- Basis for OMG's Model-Driven Architecture (MDA™)

# Informal UML Models



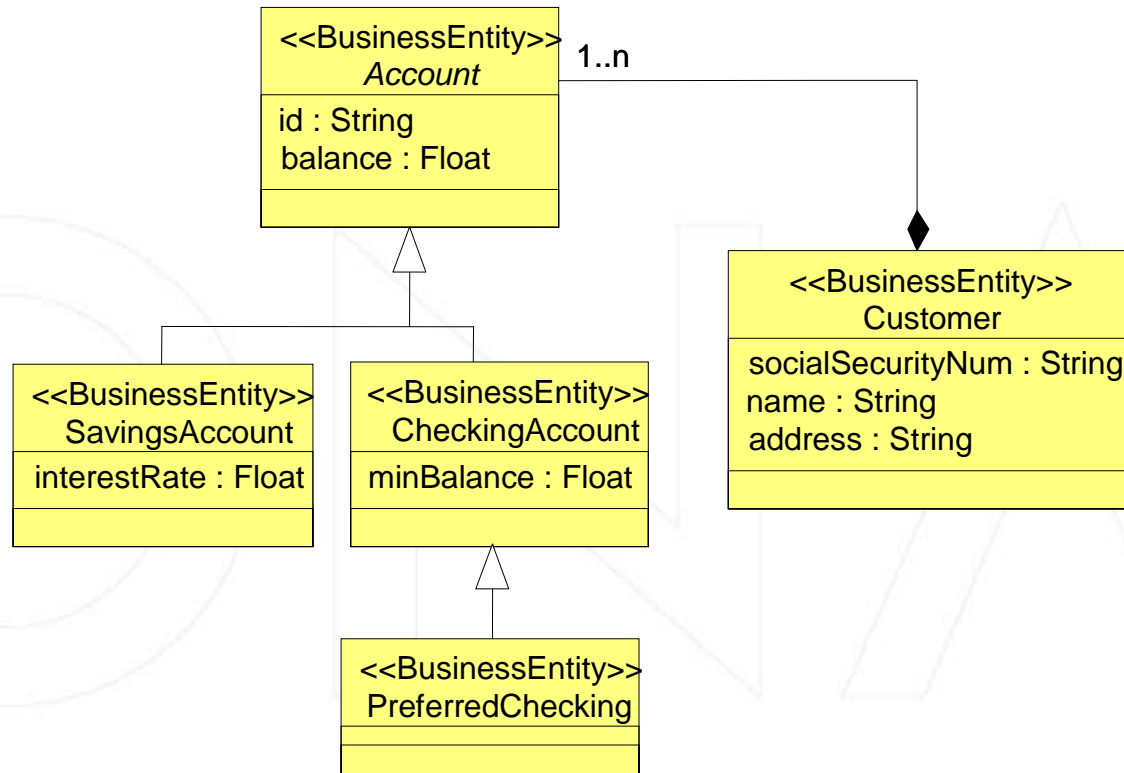
- Informal modeling
- Used to sketch out basic concepts
- Advantage over typical box and line diagrams because shapes and line types have specific meanings
- Important way to use UML, but can't drive code generators and dynamic execution engines
  - Analogously, informal text can't be compiled and executed like 3GL text

# Formal UML Models

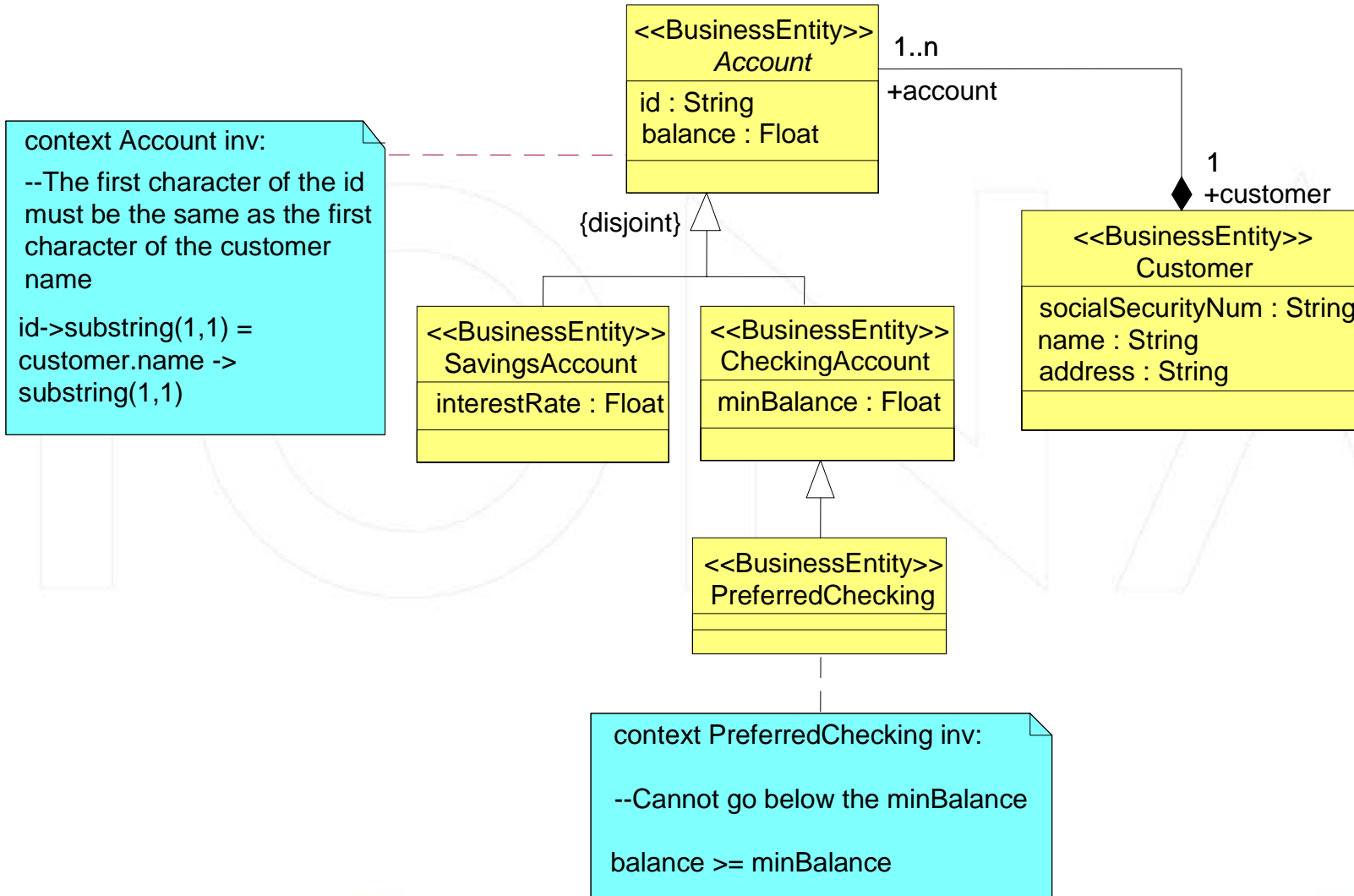


- Precise
  - Precision and detail are *not* the same!
- Computationally complete
  - Missing properties and unresolved references not acceptable
  - 3GL analogy...
    - an incomplete expression such as “a +” does not compile
    - An undeclared identifier does not compile

# Business Information Model Imprecise and Incomplete



# Business Information Model Precise and Complete



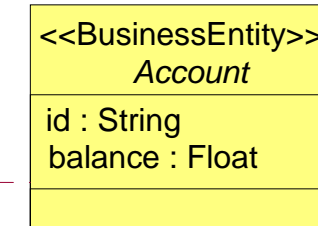
# Business Information Model Precise and Complete



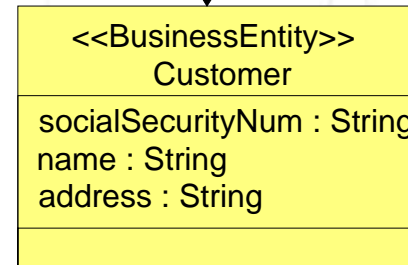
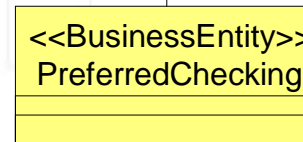
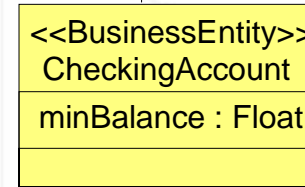
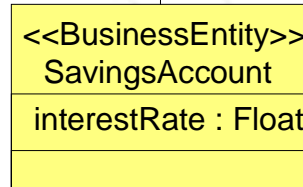
Disjoint means no instance can be an instance of both subclasses.

context Account inv:  
--The first character of the id must be the same as the first character of the customer name  
  
id->substring(1,1) = customer.name -> substring(1,1)

Invariant rules expressed in UML's Object Constraint Language (OCL)



Multiplicity could be 1 or 0..1, must be specified



◆ = composition (a.k.a. strong aggregation)

Composition of Account by Customer formally captures an important business rule: An account cannot be transferred from one customer to another.

context PreferredChecking inv:  
--Cannot go below the minBalance  
balance >= minBalance

# Business Service Model Design by Contract™



<<BusinessService>>  
FundsXFer

XFerFromChecking(in fromAcct : CheckingAccount, in toAcct : SavingsAccount, amount : Float, out fromAcctBal : Float, out toAcctBal : Float)

```
context FundsXFer (XFerFromChecking)
--Pre and post conditions

pre:
  {fromAcct.balance >= amount}
  {fromAccount.customer = toAccount.customer}

post:
  {fromAcct.balance = fromAcct.balance@pre - amount}
  {toAcct.balance = toAcct.balance@pre + amount}
  {fromAcctBal = fromAcct.balance}
  {toAcctBal = toAcct.balance}
```

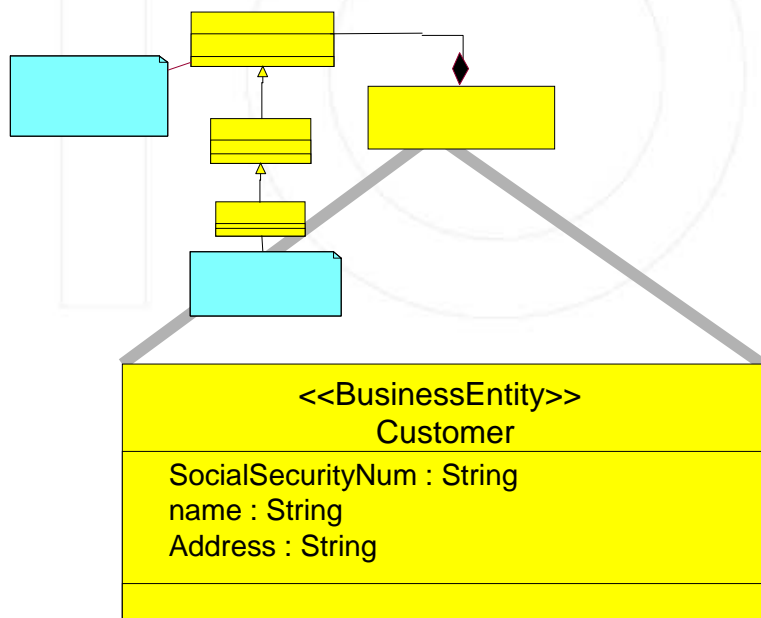
# Mapping the Business Information Model to XML



Platform-Independent Model

XMI's UML-XML Mapping Rules  
Produce

XML DTD (or Schema)



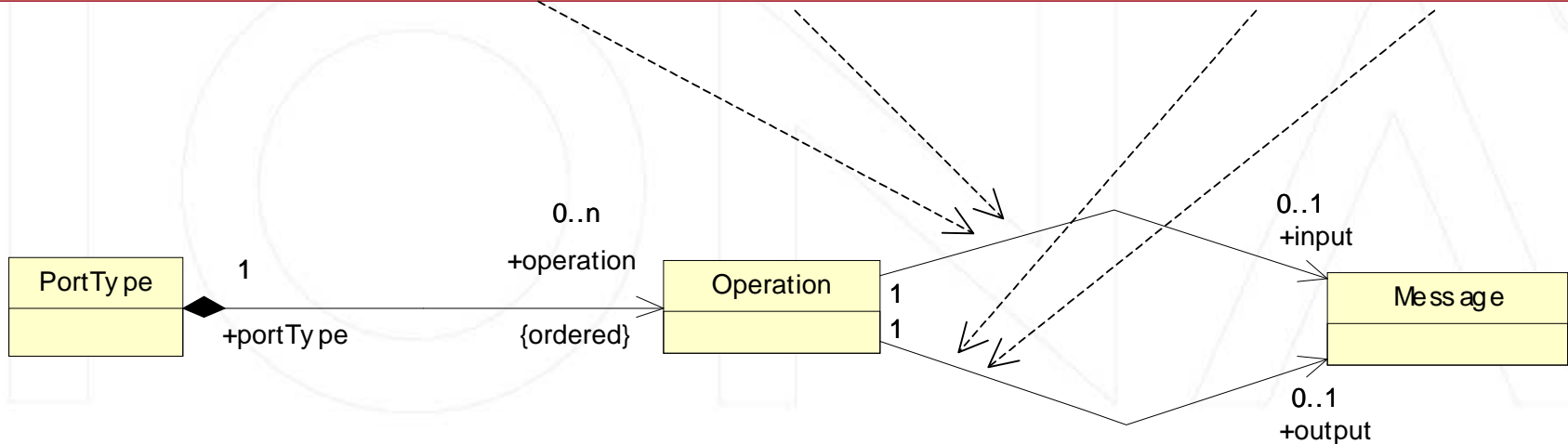
```
...
<!ELEMENT Bank.Customer.SocialSecurityNum (#PCDATA |
XMI.reference)*>
<!ELEMENT Bank.Customer.name (#PCDATA | XMI.reference)* >
<!ELEMENT Bank.Customer.Address (#PCDATA | XMI.reference)* >
...
```

# Mapping the Business Service Model to WSDL



<<BusinessService>>  
FundsXfer

XferFromChecking(**in** fromAcct : CheckingAccount, **in** toAcct : SavingsAccount, amount : Float, **out** fromAcctBal : Float, **out** toAcctBal : Float)

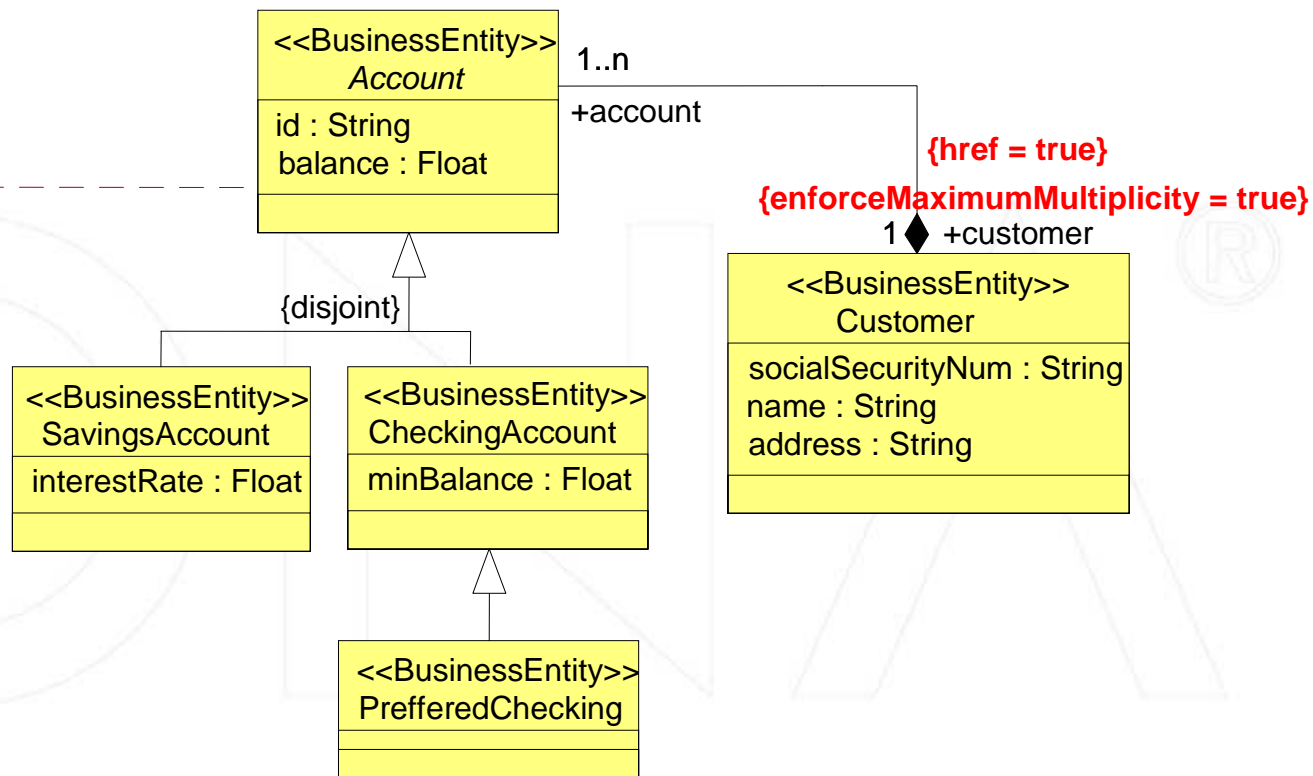


**The message payload format is based on a UML-XML mapping applied to the business information model.**

# Fine Tuning XML Generation Using XMI Parameters



context Account inv:  
 --The id must begin with "A"  
 and be seven characters long  
 id->substring(1,1) = "A" and  
 id->size() = 7



context PreferredChecking inv:  
 --Cannot go below the minBalance  
 balance >= minBalance

# UML Profiles

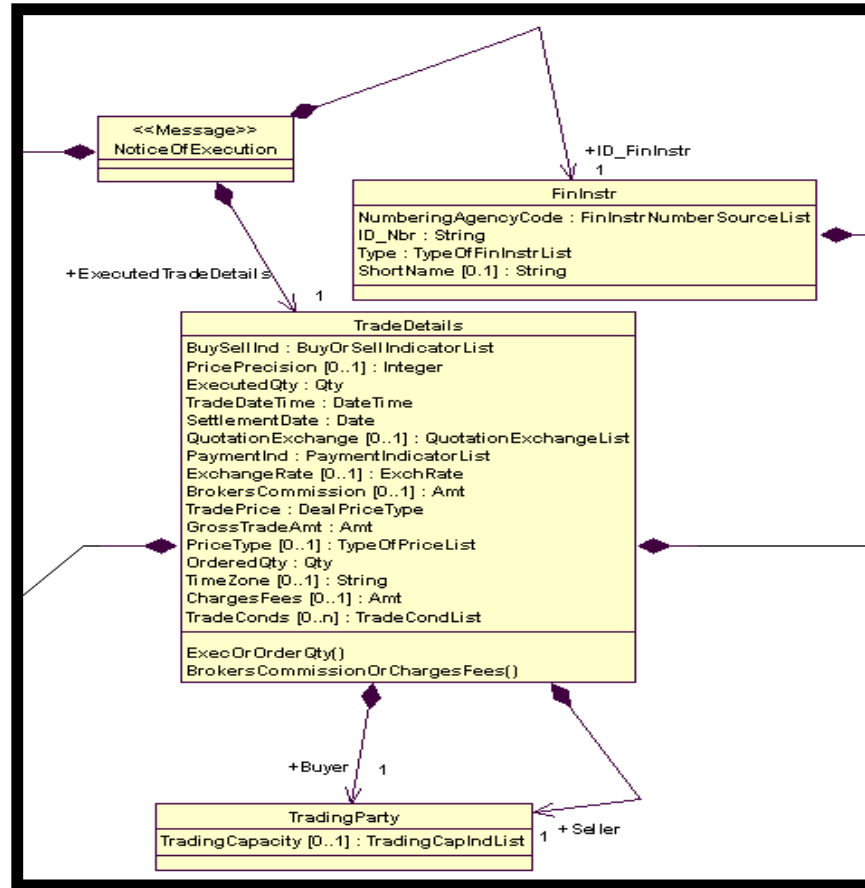


- In order to define a profile...
  - Select a subset of UML
  - Use UML's built in extension mechanisms to extend the subset
    - Stereotypes, e.g. <<BusinessEntity>>
    - Tagged Values, e.g. {href = true}
- Two uses of profiles
  - To model a particular domain, e.g. business information, business services, business collaborations, realtime systems, telecomm, etc.
  - To parameterize mappings to specific technologies

# ebXML



*from*



**Information  
model**

*to...*

**...XML  
representation**

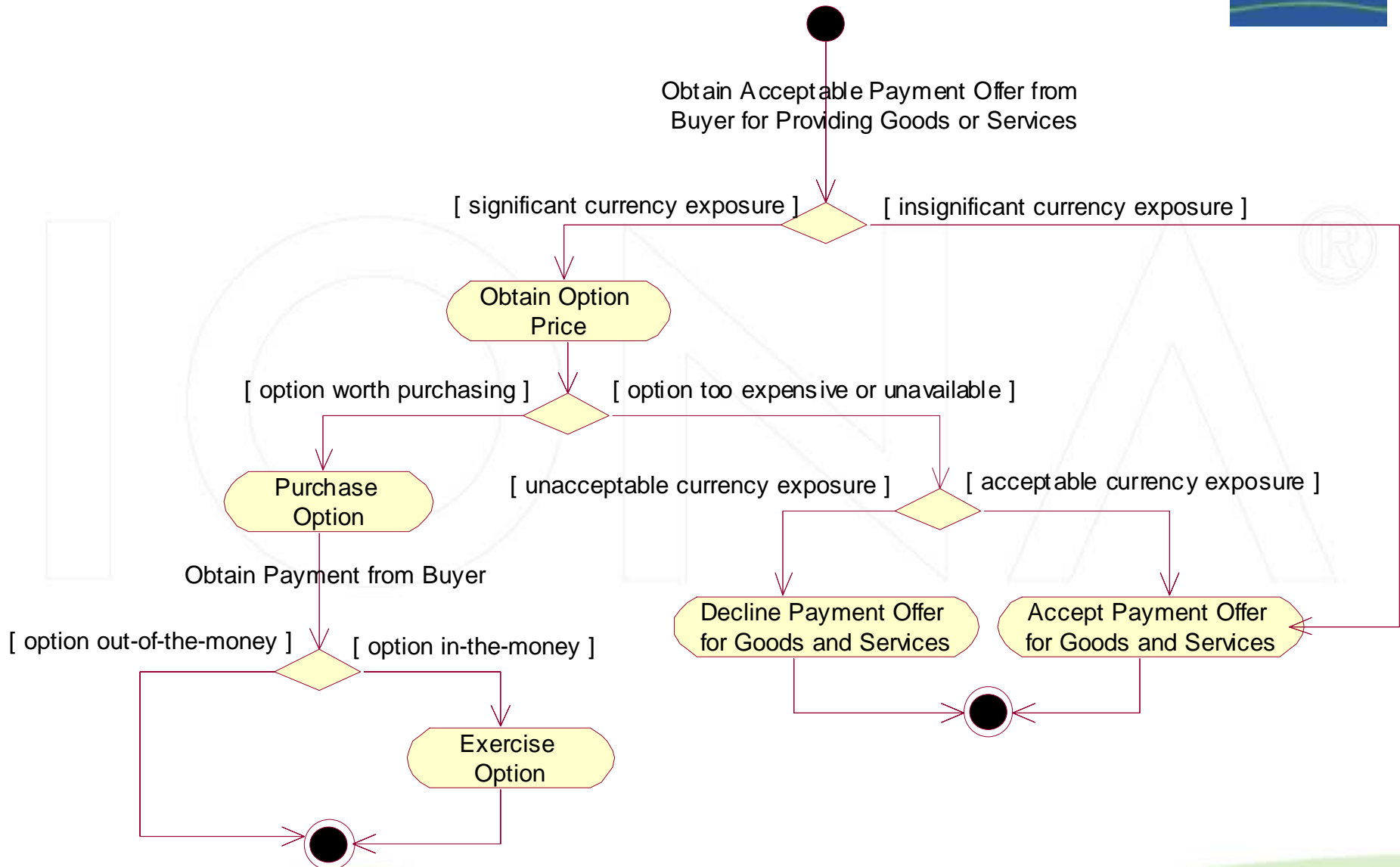
# The Frontier...

# Automating Business Processes and Choreographies

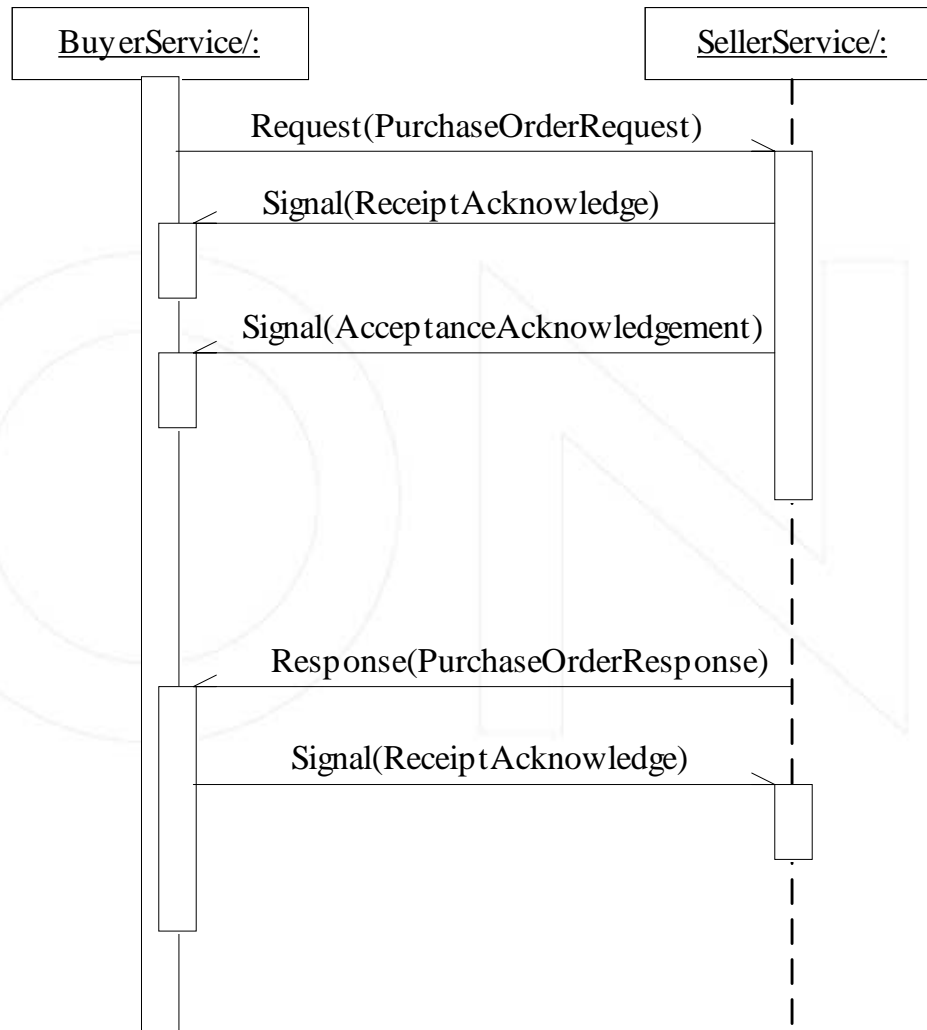


- Here again, there is a mix of technologies
  - ebXML Business Process Specification Schema (BPSS)
  - RosettaNet PIP
  - WSFL, WSEL (IBM)
  - XLANG (Microsoft)
- Here again, the business processes and choreographies are central
- UML Activity and Interaction Models can provide the necessary abstractions
  - With mappings to the various technologies

# UML Activity Model



# UML Interaction Model



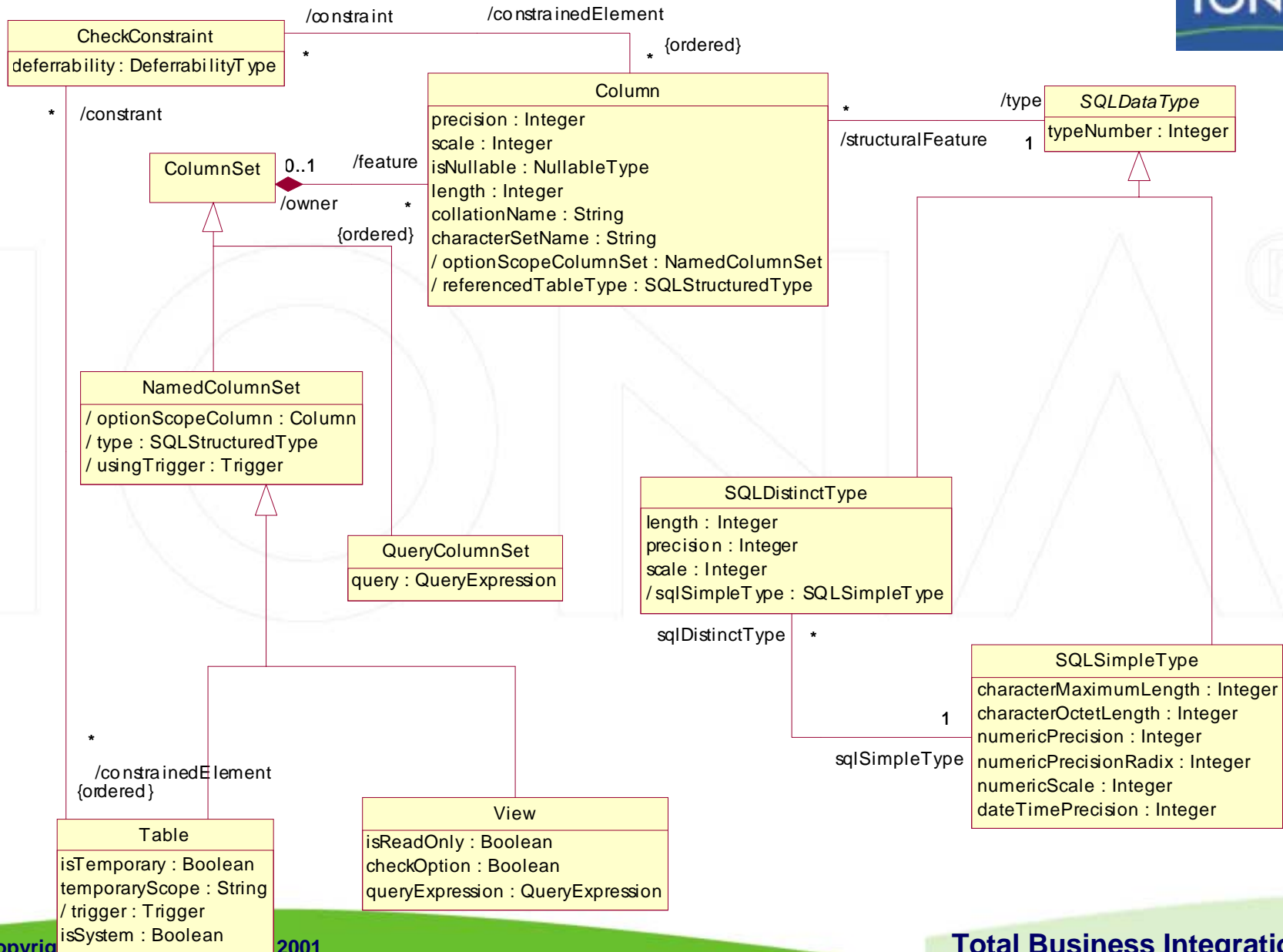
From UN/CEFACT Modeling Methodology (UMM)

# MDA: The Big Picture



- Raising the level of abstraction on a broader scale
- Efforts underway to define UML profiles for
  - Realtime
  - Telecom
  - Component Composition and Assembly
- Meta Object Facility (MOF™)
  - Subset of UML
    - With semantics nailed down precisely
    - For modeling metadata
  - Mappings to XML (XMI), Java (JMI), and CORBA
- Like Web Services Integration, MDA can be overhyped
  - Transition will unfold gradually
  - But will be profound

# MOF Model of Metadata



# MOF Overview

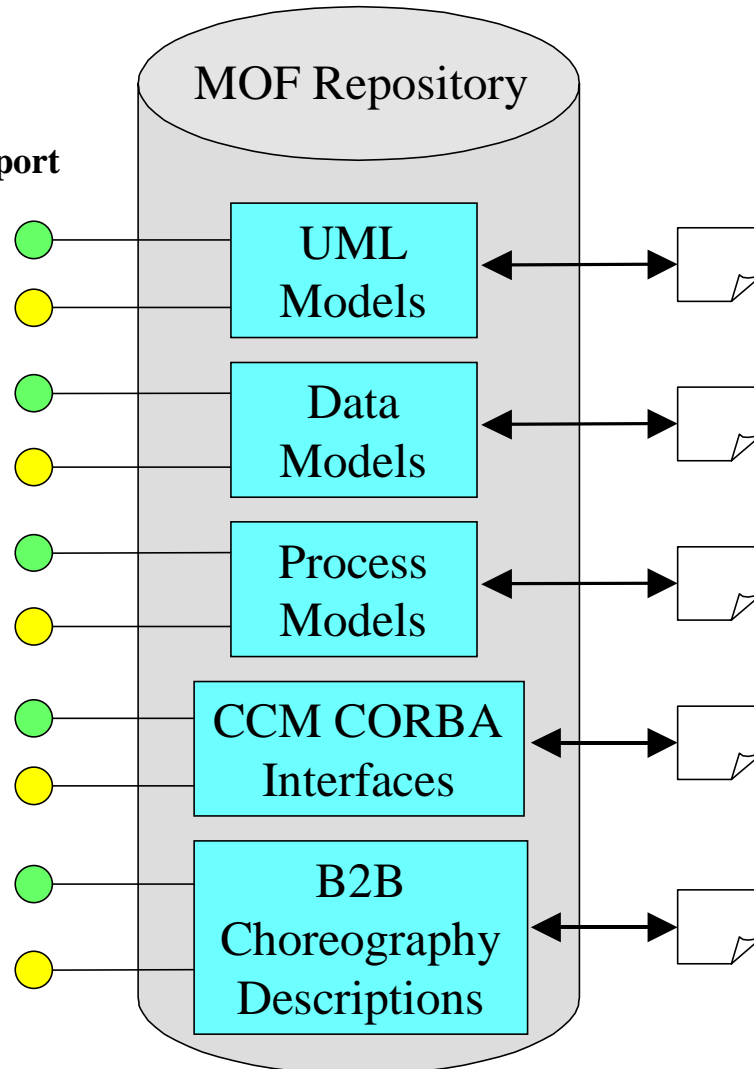


● = MOF CORBA Interfaces

● = MOF Java Interfaces (JMI)

☐ = MOF XML (XMI) Import / Export

Work in progress: MOF-WSDL mapping



# To Learn More



- UML 1.4
  - <http://cgi.omg.org/cgi-bin/doc?ad/2001-02-13>
- MOF 1.4
  - <http://cgi.omg.org/cgi-bin/doc?ptc/01-08-22>
- XMI 2.0
  - <http://cgi.omg.org/cgi-bin/doc?ad/2001-06-12>
- JMI Public Review Release
  - <http://jcp.org/aboutJava/communityprocess/review/jsr040/index.html>
- UML Profile for Schedulability, Performance, and Time (a.k.a. UML Profile for Realtime)
  - <http://cgi.omg.org/cgi-bin/doc?ad/2001-06-14>

# Trademarks



- Design by Contract is a trademark of Interactive Software Engineering
- Model-Driven Architecture, MDA, Unified Modeling Language, UML, XMI, and MOF are trademarks of the Object Management Group (OMG)



IONA<sup>®</sup>

Questions?