Modeling Enterprise Collaboration

Enterprise
Collaboration
Architecture

EDOC

A tutorial on applying Model Driven Architecture using the OMG Enterprise Collaboration Architecture with UML

Introductions

DataAccessTechnologies
Where Business Meets Technology

Cory Casanave
cory-c@enterprise-component.com

Primary author of "CCA" in EDOC

What is the Enterprise Collaboration Architecture?

- ECA is a "profile of UML", a way to use UML for a specific purpose - it is an OMG standard
  - That purpose is *modeling enterprise systems.*
- You can also think of this as a "modeling framework" for enterprise computing
- ECA is part of the "Model Driven Architecture" (MDA) initiative of the OMG
  - Using precise modeling techniques as part of the development lifecycle to speed development and provide technology independence
- ECA has been adopted by the OMG as part of the EDOC RFP.

The Agile Enterprise

Has a competitive advantage in its capability to embrace collaboration and change

Integrating Enterprises, People & Systems - Worldwide

Enabling **e**

Using Internet Technologies

Integrating Enterprises, People & Systems - Worldwide

- **Business Requirements**
  - Virtual Enterprises
  - Enterprise Integration (EAI)
  - Supply-chain automation (B2B)
  - Customer Integration (B2B)
  - Web deployment (B2C)
  - Internet Marketplace (B2C)

- **Collaboration and Integration**

The dynamic reality

❚ The information system must facilitate;
- ❚ Rapid realization of business goals
- ❚ Integration of independent processes and systems
- ❚ Multiple and Changing
  - *business requirements*
  - *business processes*
  - *technologies*
  - *standards*
  - *enterprise boundaries*
  - *partners*

Technology Stew

❚ Web services
❚ .NET
❚ C'
❚ XML
❚ EAI
❚ Active Web pages
❚ EJB
❚ Java Beans
❚ Java
❚ Corba

❚ MQ-Series
❚ C++
❚ SQL
❚ Cobol
❚ IMS
❚ CICS
❚ …

Technology is transient, but we must embrace and adapt to it to provide meet current requirements

Problems to be solved

- Integration Nightmare
- Infrastructure, Version & Vendor lock-in
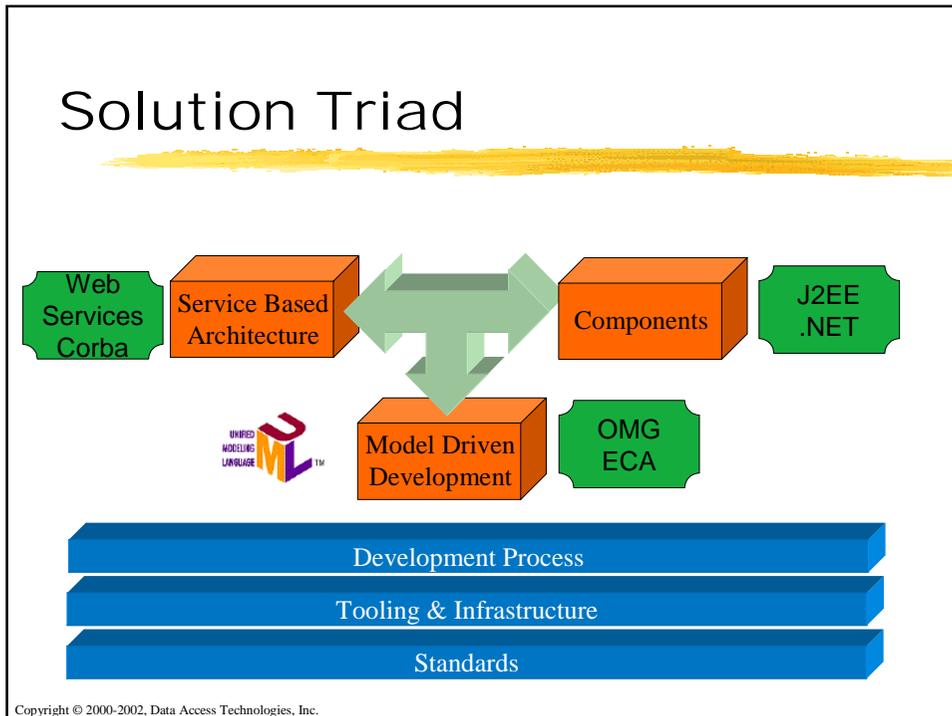- Complex, divergent and manual development and deployment processes

- *Typical solutions require buy-in (Lock-in) to expensive, pervasive and proprietary infrastructure*

Goals

- A scalable and robust enterprise architecture
- Loosely coupled enterprise components
- Enable rapid provisioning of business solutions
  - Simple, reproducible processes supporting reuse
- Technology & vendor independence
- Enable the integration and collaboration of multiple;
  - Business units (internal and external)
  - Customers
  - Suppliers
  - Systems
  - Technologies

Solution Triad

Web Services Corba

Service Based Architecture

Components

J2EE .NET

UNIFIED MODELING LANGUAGE

Model Driven Development

OMG ECA

Development Process
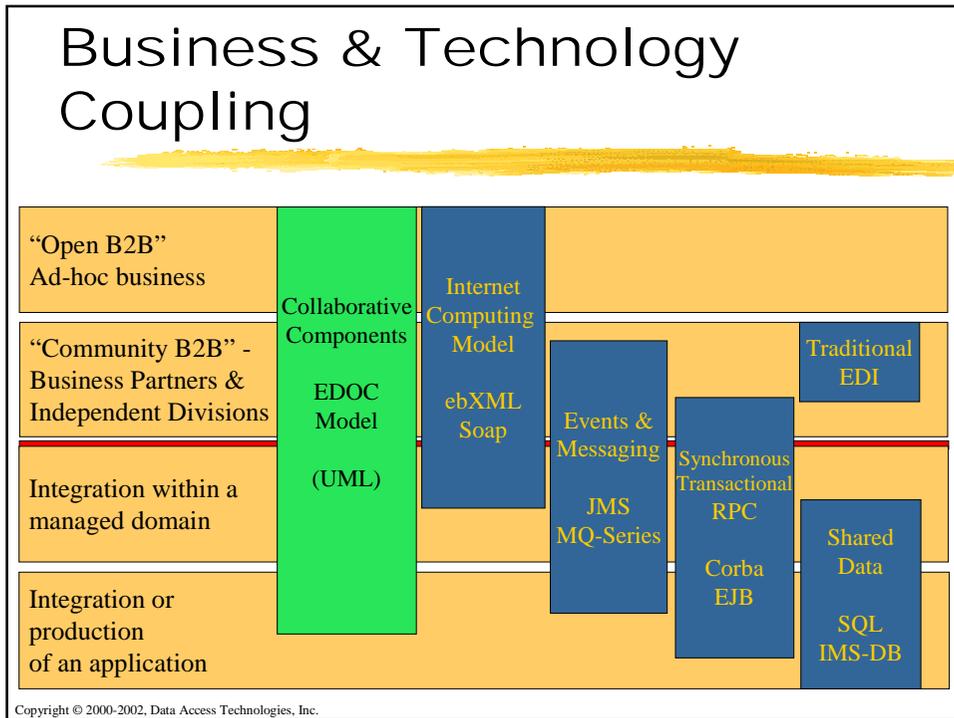
Tooling & Infrastructure

Standards

The new center

- The strategic core of you systems must be the business its self
- Only technology independent business focused models will survive the transience of technology and lock-in
- These models can become *part of your source code*, driving enterprise applications
- Enabler: Model Driven Architecture (MDA) with EDOC-ECA

Extreme Modeling

Business & Technology Coupling

	Collaborative Components EDOC Model (UML)	Internet Computing Model ebXML Soap				
"Open B2B" Ad-hoc business						
"Community B2B" - Business Partners & Independent Divisions					Traditional EDI	
Integration within a managed domain			Events & Messaging JMS MQ-Series	Synchronous Transactional RPC Corba EJB		
Integration or production of an application					Shared Data SQL IMS-DB	

Collaborative Vs. Managed business processes

❚ The enterprise collaboration architecture is intended for collaborative business processes

 ❚ People, departments, systems and enterprises working together within a defined process

❚ Contrast with the managed business process

 ❚ A central authority directs the activities of others

❚ Both have their place

 ❚ Collaborative - B2B, EAI

 ❚ Managed - Fine-grain business processes - factories

Collaborative Business Processes

Independent actors working within a defined process

No central authority
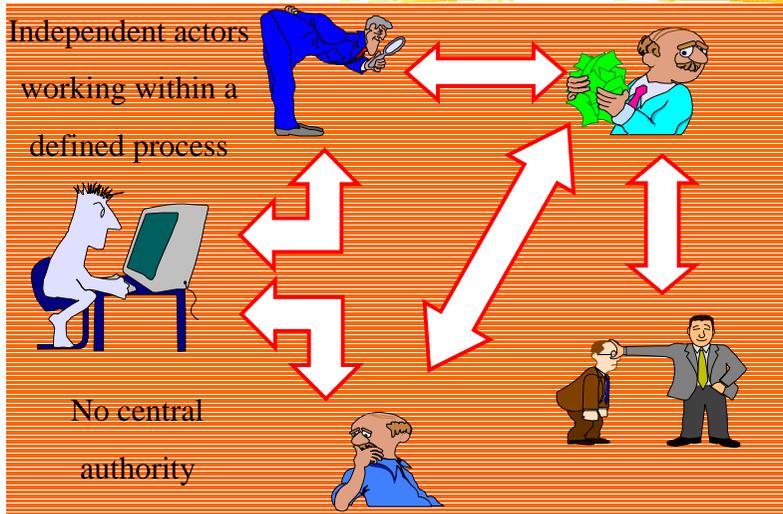
Managed Business Processes

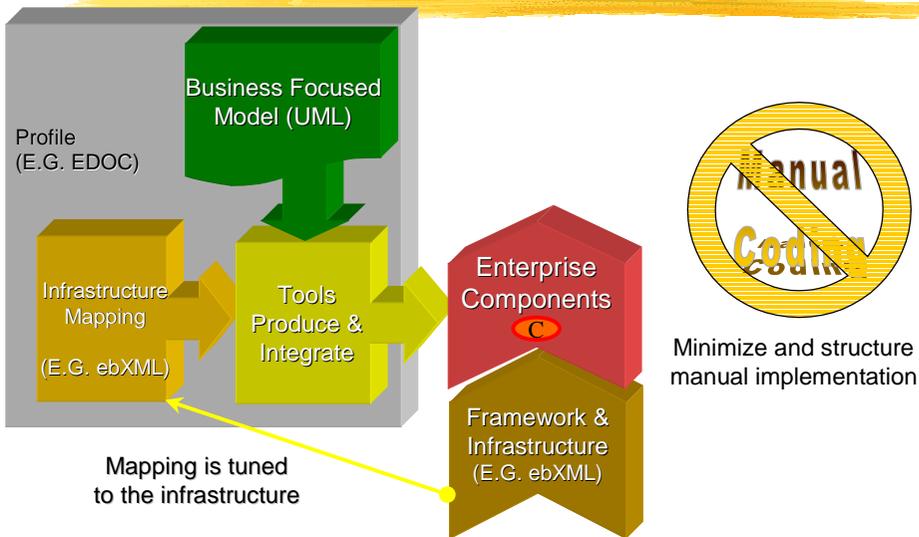Someone or something keeps track of everything
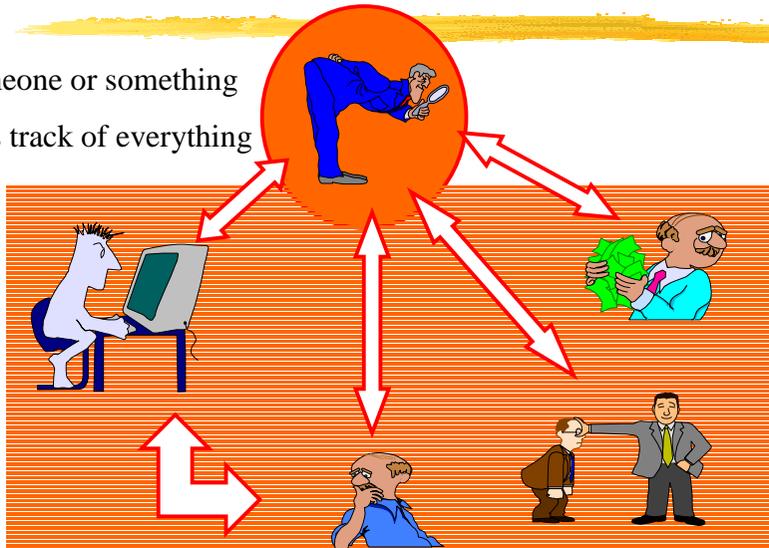
Collaboration is Key

- Collaboration is a key differentiation and key cost center (Healthcare Example)
  - Customer Collaboration
  - Claim processing
  - Disputes
  - Physician Collaboration
  - Payer Collaboration
  - Hospital Collaboration
  - Broker Collaboration
  - Government Collaboration
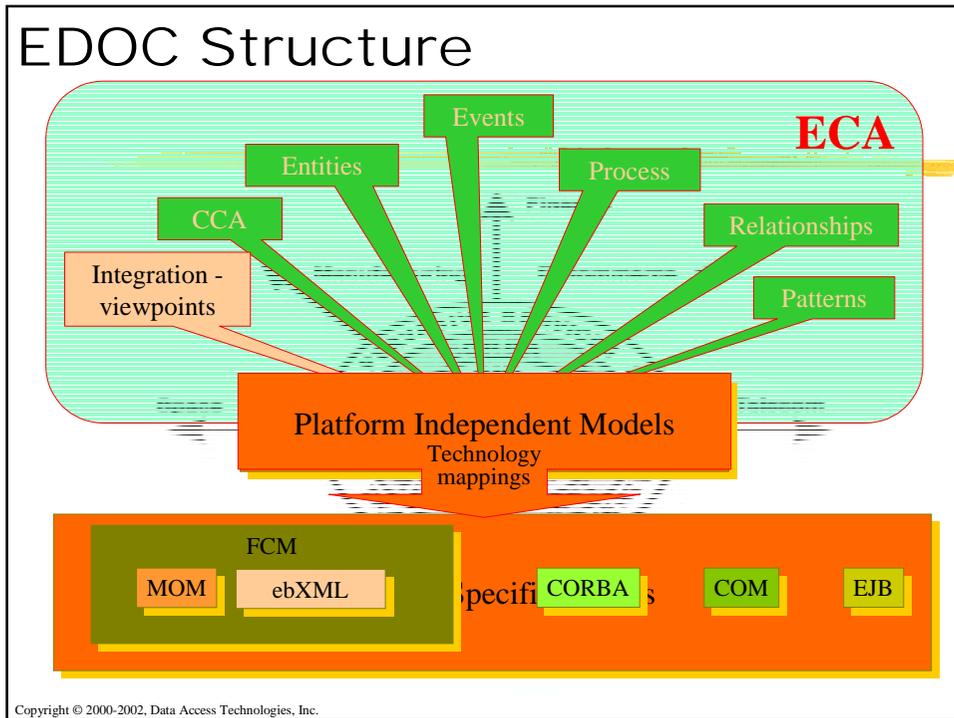 - Employee Collaboration
  - Others...

The system integrates multiple collaborations

Automated Model Driven Architecture

Profile (E.G. EDOC)

Business Focused Model (UML)

Infrastructure Mapping (E.G. ebXML)

Tools Produce & Integrate

Enterprise Components
C

Framework & Infrastructure (E.G. ebXML)

Mapping is tuned to the infrastructure

Manual Coding

Minimize and structure manual implementation

EDOC Structure

Events

ECA

Entities

Process

CCA

Relationships

Integration - viewpoints

Patterns

Platform Independent Models
Technology mappings

FCM

MOM   ebXML   Specifi   CORBA   COM   EJB

Enterprise Architecture

How to slice and dice

Loose Coupling

- Loose coupling is the ability for independent parts of systems to be built and evolve independently
- Tightly coupled systems
  - Prevent change (the next legacy system)
  - Cause lock-in
  - Become unmanageable
  - Prevent reuse
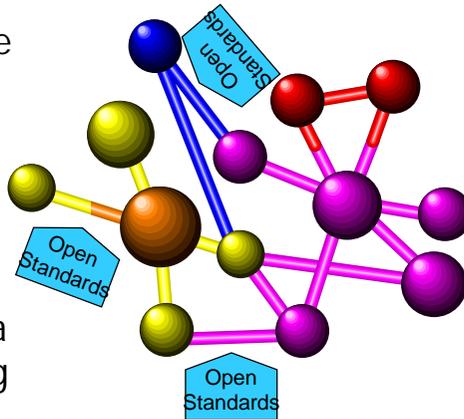- Quality architecture is essential for loose coupling

Architecture Goals

- Create a system from loosely coupled "enterprise components" that can evolve independently
- Provide well defined interfaces and interaction points between these enterprise components
- Make each enterprise component a reusable asset that can serve many business processes
- Build the information system as a community of interacting enterprise components
- Utilize open standards such as XML, EJB and Corba to integrate the enterprise components
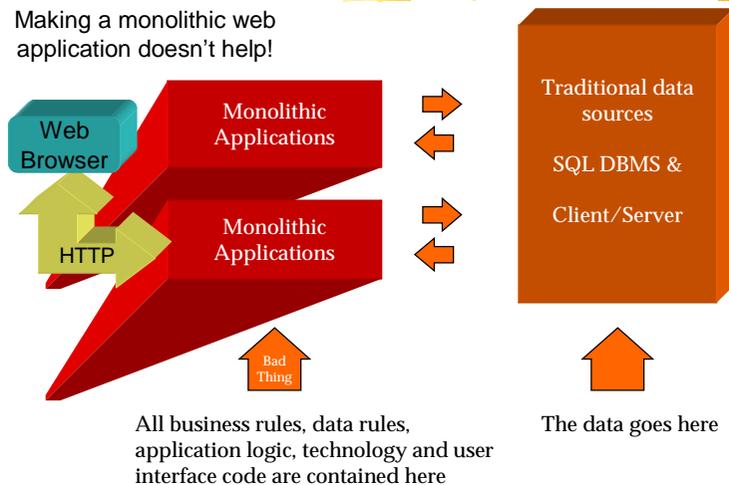
Enterprise Components

- Enterprise Components must be independent
- While being able to interoperate with each other
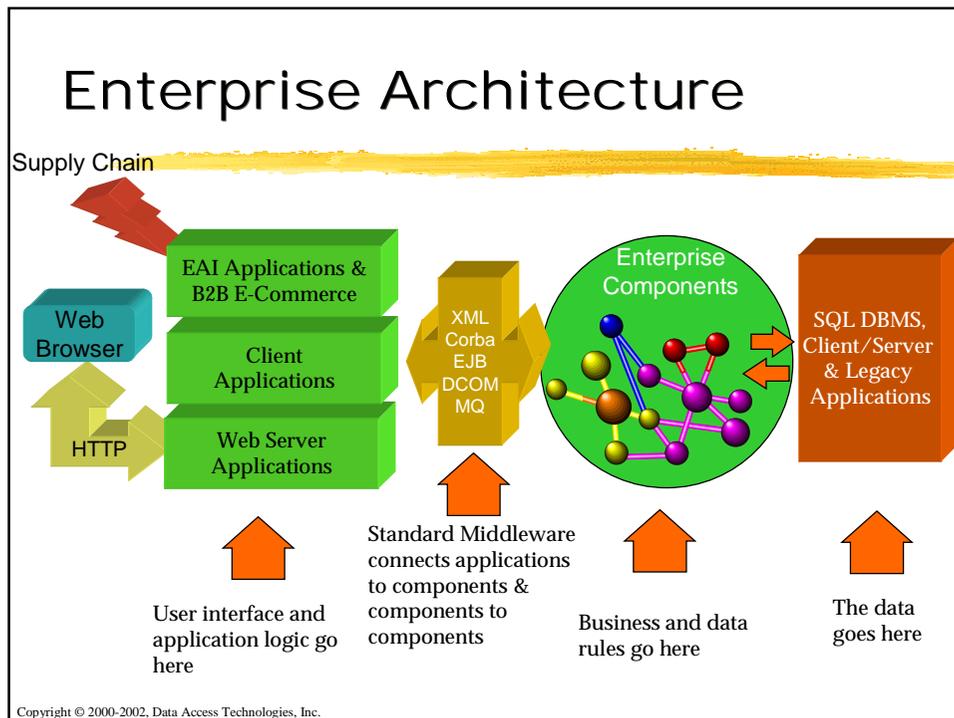- Making the information system a lattice of cooperating components

Poor Monolithic Architecture

Making a monolithic web application doesn't help!

Web Browser

HTTP

Monolithic Applications

Monolithic Applications

Traditional data sources

SQL DBMS & Client/Server

Bad Thing

All business rules, data rules, application logic, technology and user interface code are contained here

The data goes here

Enterprise Architecture

Supply Chain

Web Browser

HTTP

EAI Applications & B2B E-Commerce

Client Applications

Web Server Applications

XML Corba EJB DCOM MQ

Enterprise Components

SQL DBMS, Client/Server & Legacy Applications

User interface and application logic go here

Standard Middleware connects applications to components & components to components
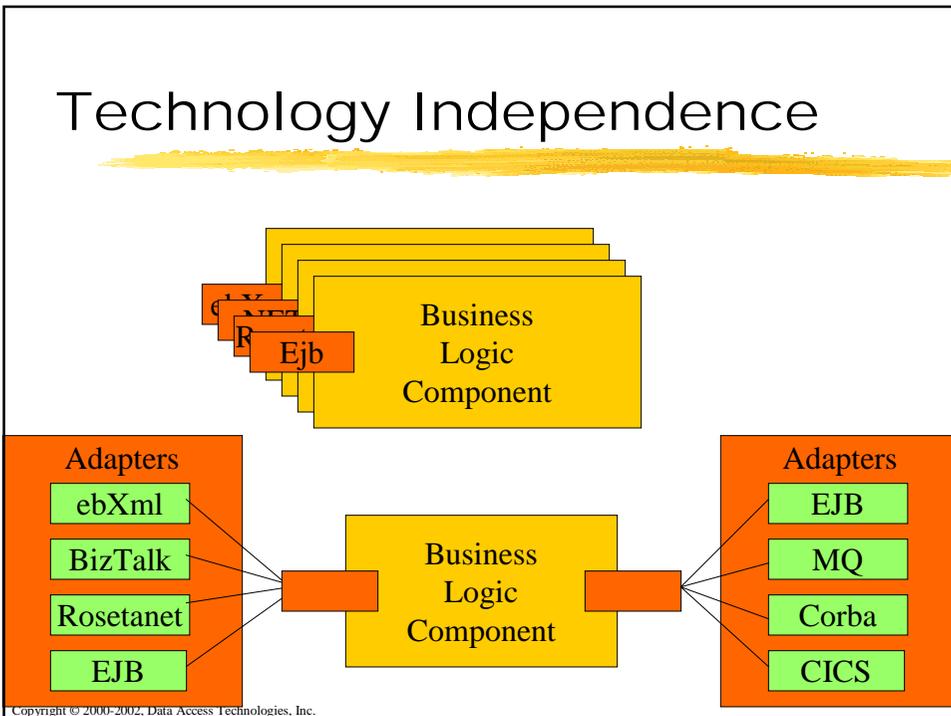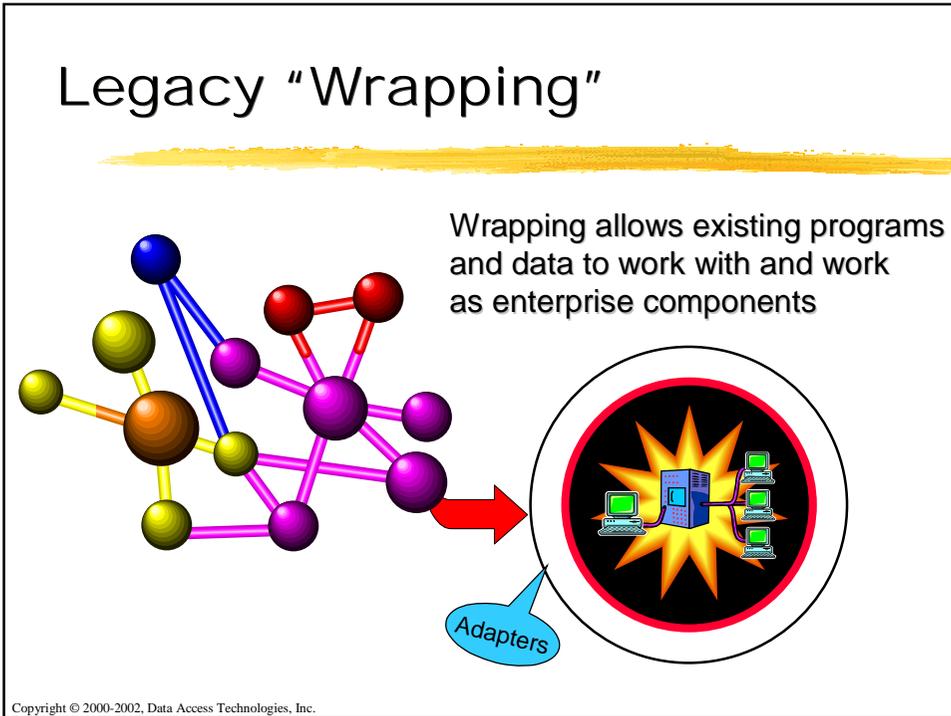
Business and data rules go here
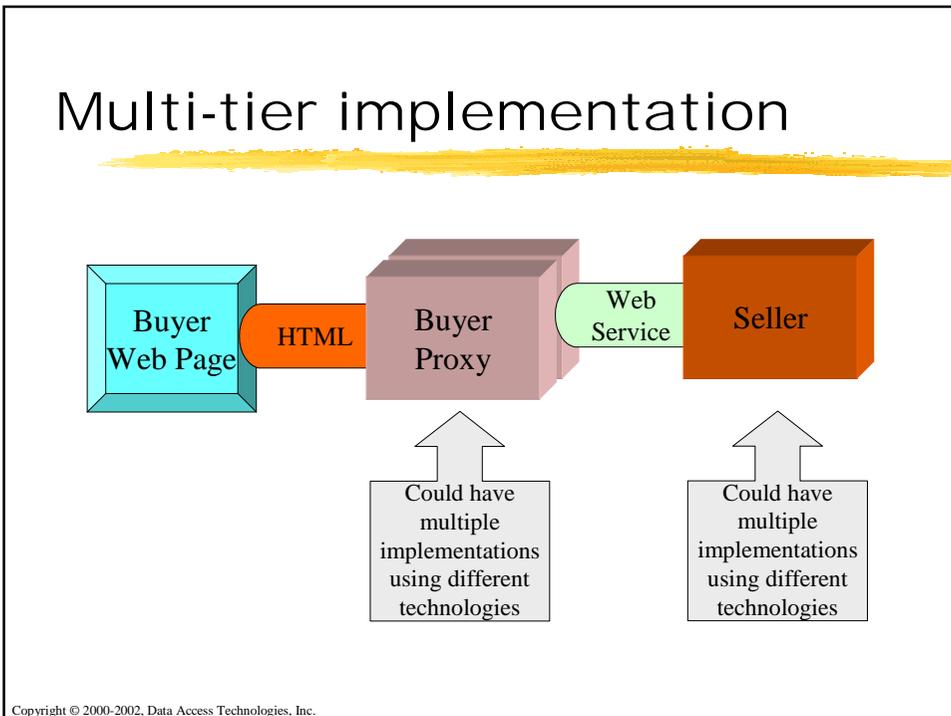
The data goes here

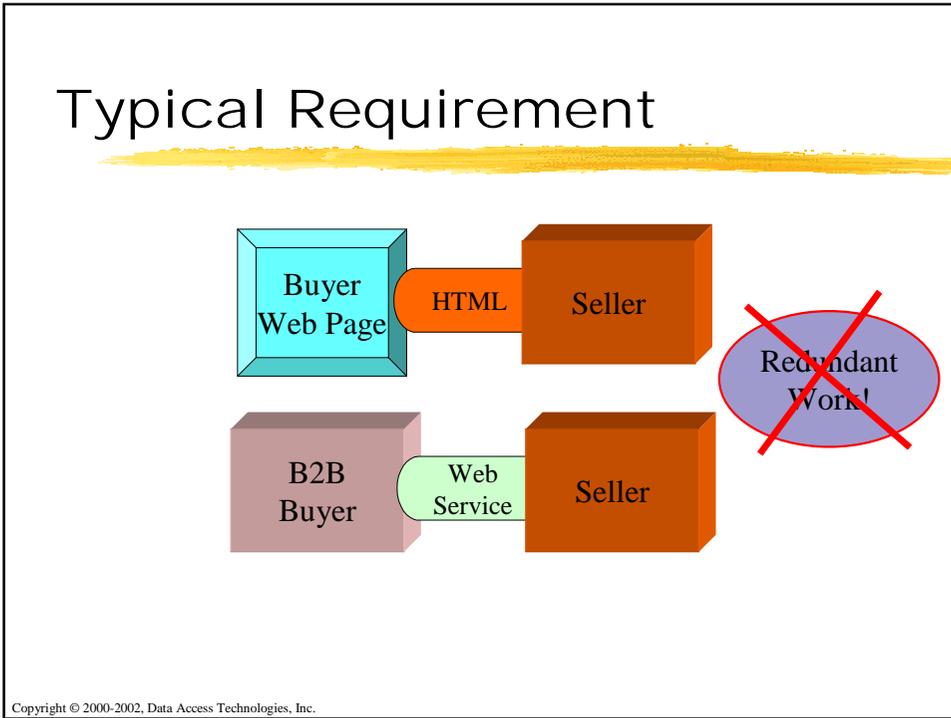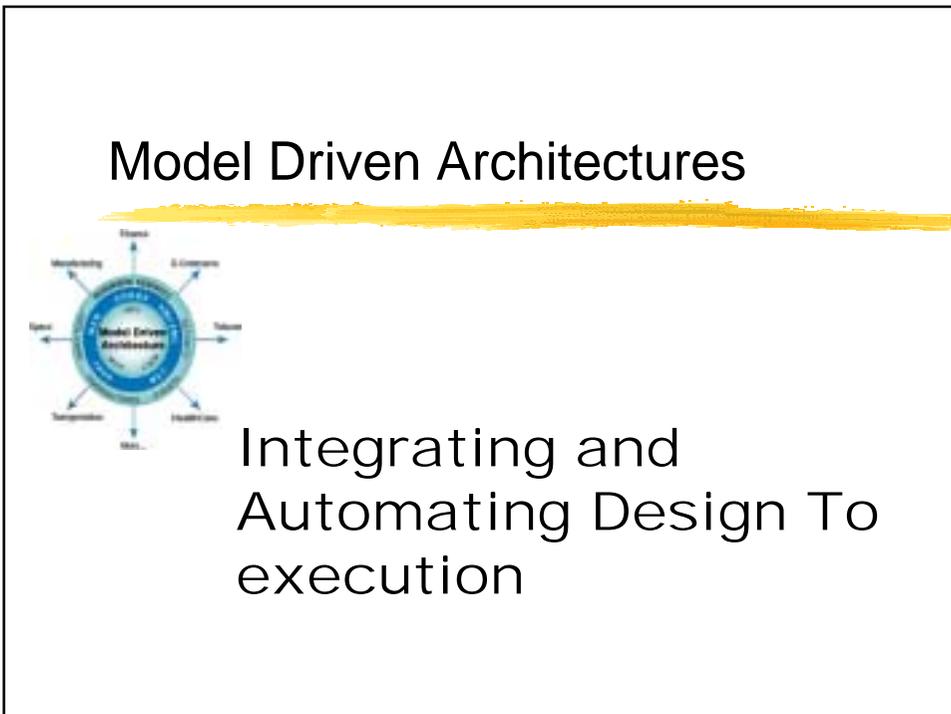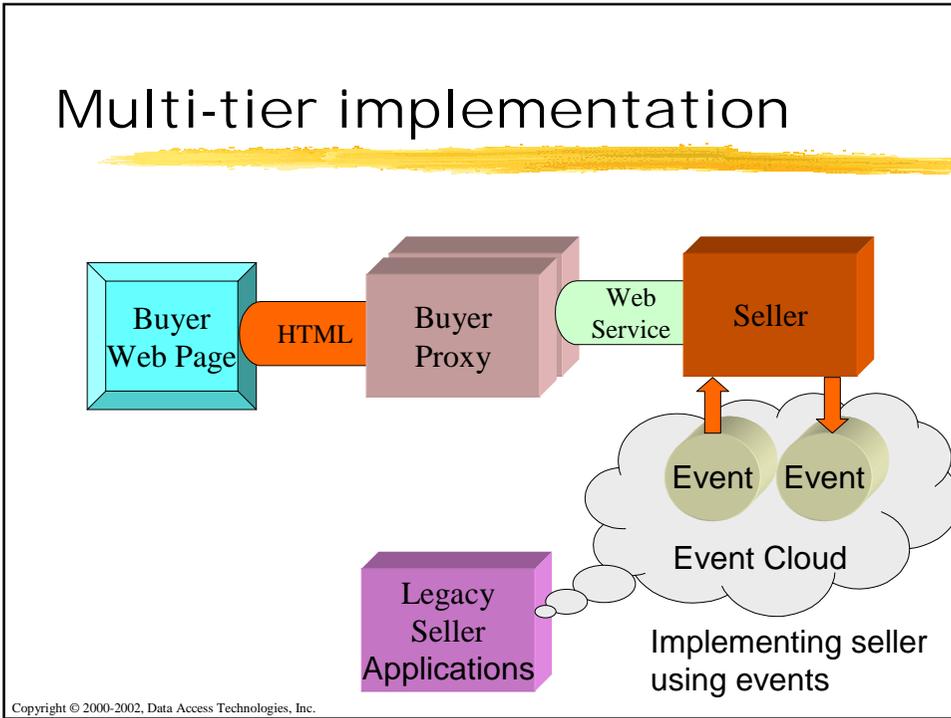Copyright © 2000-2002, Data Access Technologies, Inc.

"Wrapping" Legacy Applications and Data

▌ Enterprise Components are defined in terms of their external contract; implementation may use existing applications
  ▌ Can "call" existing application
  ▌ Can read and write legacy DBMS
  ▌ Can use "screen scraper" (Last resort)
▌ Legacy applications can appear as enterprise components but may not be implemented as components

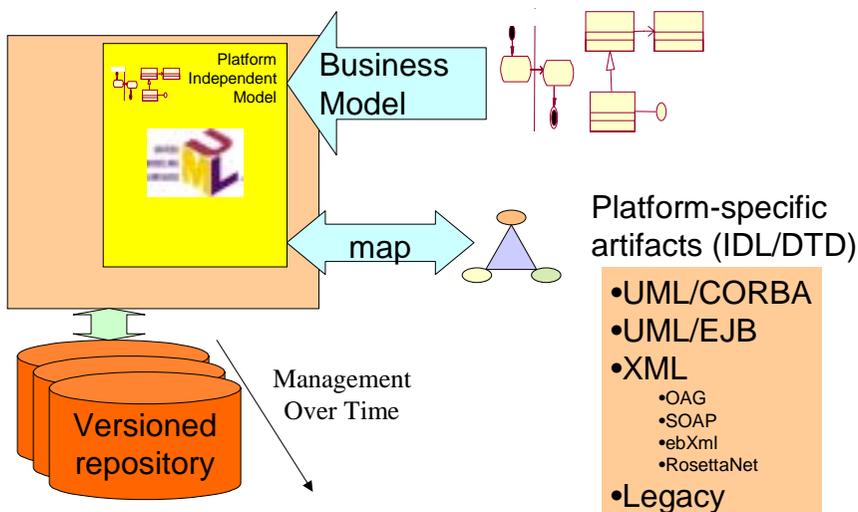Copyright © 2000-2002, Data Access Technologies, Inc.

Legacy "Wrapping"

Wrapping allows existing programs and data to work with and work as enterprise components

Adapters

Technology Independence

Business Logic Component

ebXml
Rosetanet
Ejb

Adapters				Adapters
ebXml		Business		EJB
BizTalk		Logic		MQ
Rosetanet		Component		Corba
EJB				CICS

Typical Requirement

Buyer Web Page → HTML → Seller

B2B Buyer → Web Service → Seller

Redundant Work! (crossed out)

Multi-tier implementation

Buyer Web Page → HTML → Buyer Proxy → Web Service → Seller

Could have multiple implementations using different technologies

Could have multiple implementations using different technologies

Multi-tier implementation

Buyer Web Page — HTML — Buyer Proxy — Web Service — Seller

Legacy Seller Applications

Event Event

Event Cloud

Implementing seller using events

Model Driven Architectures

Integrating and Automating Design To execution

MDA Overview

- Modeling has evolved
  - Its not just pictures anymore
- Use high level UML models made precise with profiles
- With technology specific mappings
- Define and produce substantial parts of the executable system

Models and mapping

Platform Independent Model

Business Model

map

Management Over Time

Versioned repository

Platform-specific artifacts (IDL/DTD)
- UML/CORBA
- UML/EJB
- XML
  - OAG
  - SOAP
  - ebXml
  - RosettaNet
- Legacy

Ways to map

- Generate code and other artifacts
  - Usually required to adapt to legacy
  - Emulates the manual process
- Assemble and configure existing generic and specific components
  - Both efficient and dynamic
- Interpret or otherwise "animate" the high-level model
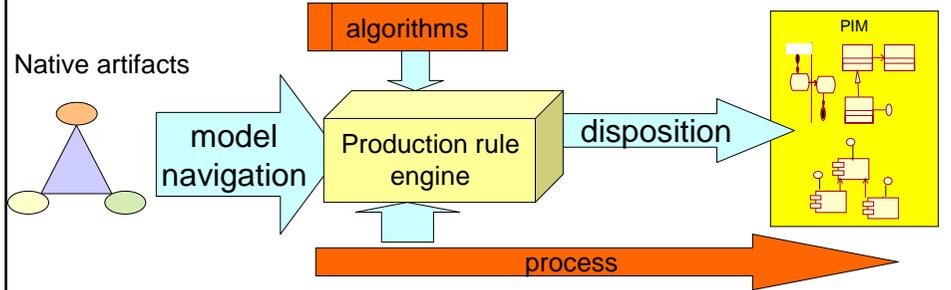 - High level & dynamic but may have performance issues

Example Generated Artifacts

Implementation Artifacts (EJB Examples)

Class Objects

Jars,Wars,Ears

Java Source

Stubs, Skeletons, Helpers, Holders, Interfaces

Homes, Managers, Primary Keys

SQL

Descriptors

BeanInfoEditors.

Documentation

Serialization, Persistence Management

Business Object Implementation Logic

M0/M1 XMI/DTD

Artifact generation involves multiple tools
• EJB Container provider;Deployment tools;Packagers;
• java development tools(IDE);persistence provider;…
Typical 10-20 per PIM Classifier
0-20% manual override

Reverse Engineering

algorithms

PIM

Native artifacts

model navigation → Production rule engine → disposition →

process →

- Native meta-model is platform-specific
  - XML DTD/Schema; java introspection; SQL tables; legacy model; etc.
- Map navigates the native meta-model, populates PIM
- Limited semantic recovery
  - Information and middleware models work best

MDA and Components

Platform Independent Model

Business Model →

Assemble & Execute → Components
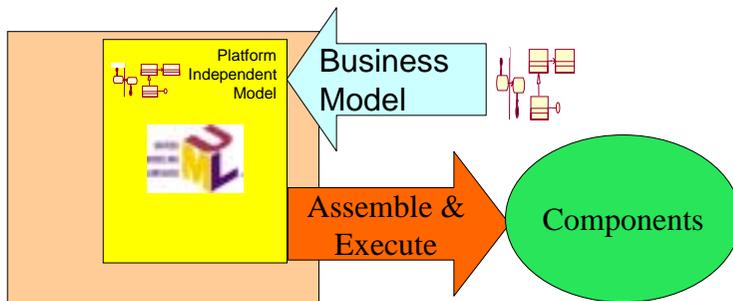
Enabled but not required

What can we automate?

Model

Mappings

Technology Infrastructure

Process & Data Integration rules

Enterprise Collaboration

Programmer

Internal business logic
Legacy integration
30% to 5%
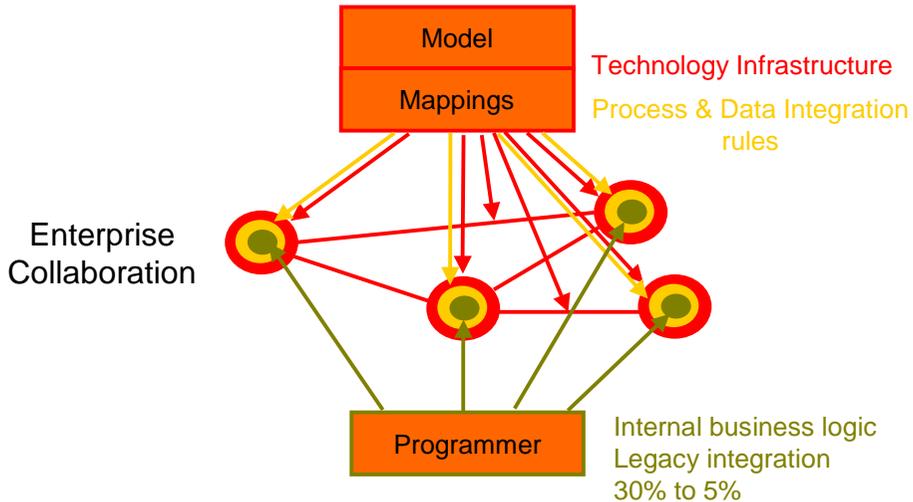
Summary of MDA benefits

- Isolates domain specifications from platform details
  - Reduces complexity
  - Preserves domain model semantics
  - Increases stability and lifetime
  - Generates to platform/legacy of choice
- Decreased development time
  - fast iterative development
  - separation between the engineering and business requirements
- Increased quality.
- Builds on industry directions

Users

Domain
Specifications

MDA
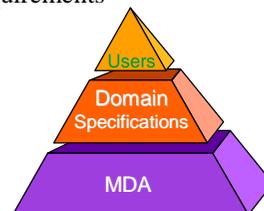
Collaborations and Roles

Conceptual Foundation

Portions copied with permission of Trygve
Reenskaug - Taskon

OORAM

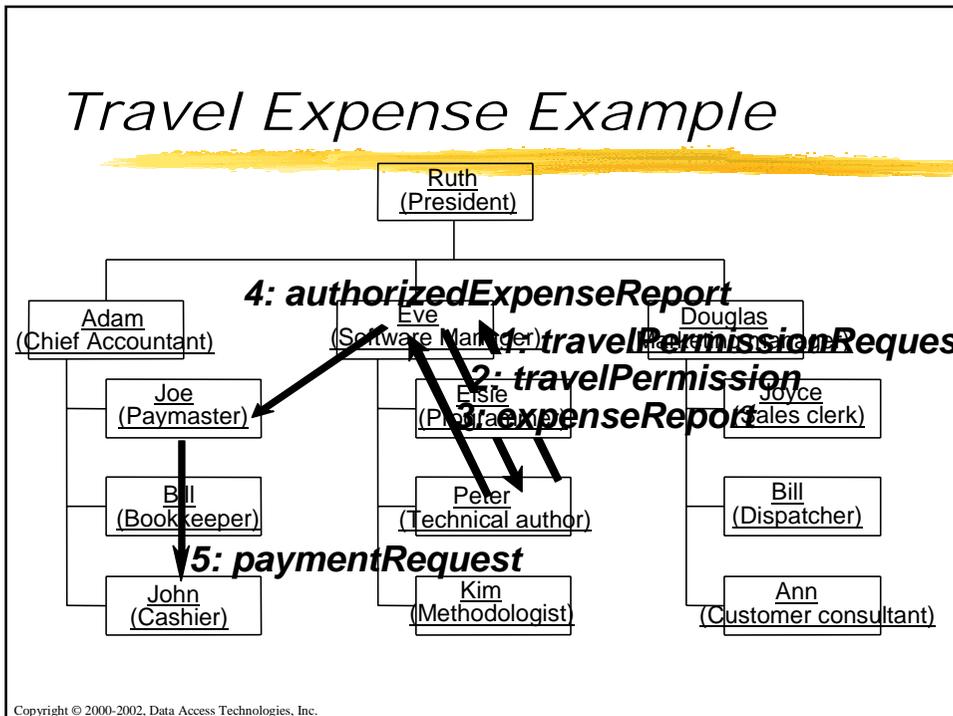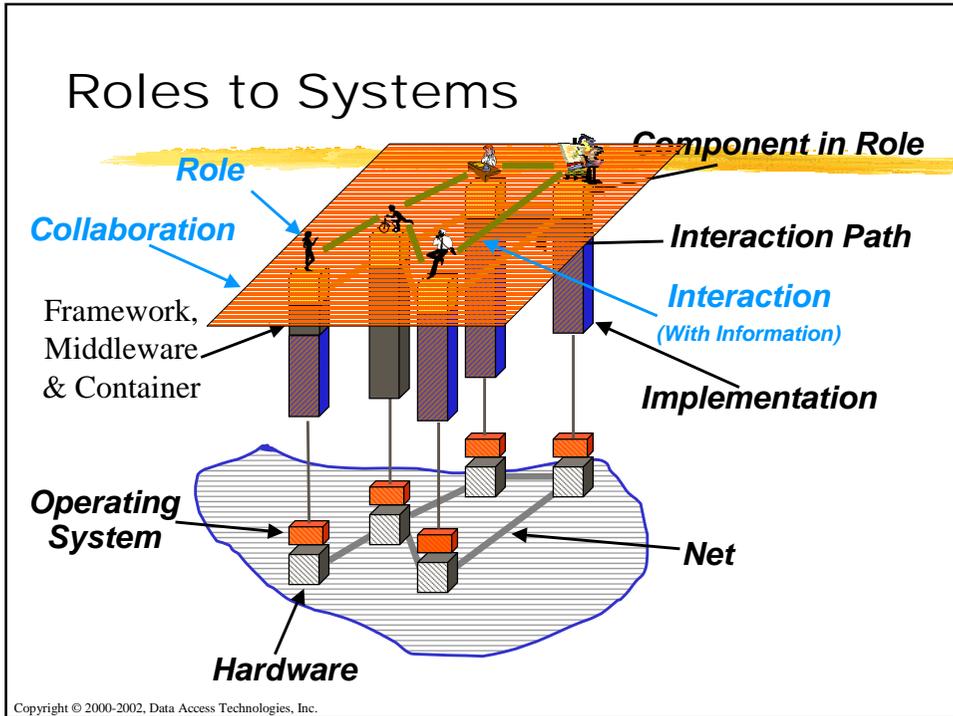(http://www.ifi.uio.no/~trygver)

History

| OORAM Object Oriented Role Analysis | UML Collaborations | Enterprise Collaboration Architecture |

Influence

The Connected Enterprise
Content and Communication

Police Dispatcher Role

Digital Map
Census Data
Police Records
House Drawings
Aerial Photos

Copyright © 2000-2002, Data Access Technologies, Inc.

Multiple roles in a collaboration

Copyright © 2000-2002, Data Access Technologies, Inc.

Roles to Systems

Role

Collaboration

Component in Role

Interaction Path

Interaction
(With Information)

Framework,
Middleware
& Container

Implementation

*Operating
System*

Net

Hardware

Travel Expense Example

Ruth
(President)

Adam
(Chief Accountant)

Eve
(Software Manager)

Douglas
(Personnel manager)

4: authorizedExpenseReport

1: travelPermissionRequest

2: travelPermission

Joe
(Paymaster)

Elsie
(Programmer)

Joyce
(Sales clerk)

3: expenseReport

Bill
(Bookkeeper)

Peter
(Technical author)

Bill
(Dispatcher)

5: paymentRequest

John
(Cashier)

Kim
(Methodologist)

Ann
(Customer consultant)

Diagram
Travel Expense Model

Ruth
(President)

Adam
(Chief Accountant)

/ Authorizer (ger)

Douglas
(Marketing manager)

/ Paymaster

Elsie
(Programmer)

Joyce
(Sales clerk)

/ BookKeeper

/ Traveler (r)

Bill
(Dispatcher)

John
(Cashier)

Kim
(Methodologist)

Ann
(Customer consultant)

Objects --> ClassifierRoles

Collaboration Diagram

Traveler

Authorizer

Book
Keeper

Paymaster

Travel Permission Decision
Travel Expense System

| Traveler | Peter | Period | week 11 | Planned cost | USD 2000.- |

Purpose Attend TOOLS Europe 2001

Current plan for Peter

activity 1
activity 2
activity 3

week   10   11   12 13 14 15

Budget + commitments

```
Item    Budget Committed
Travel 10,000     4,000
```

| Permit | | Reject |

Synthesis - Components play several roles

ProjectManager

ProjectParticipant

ProjectManager

ProjectParticipant

Jane
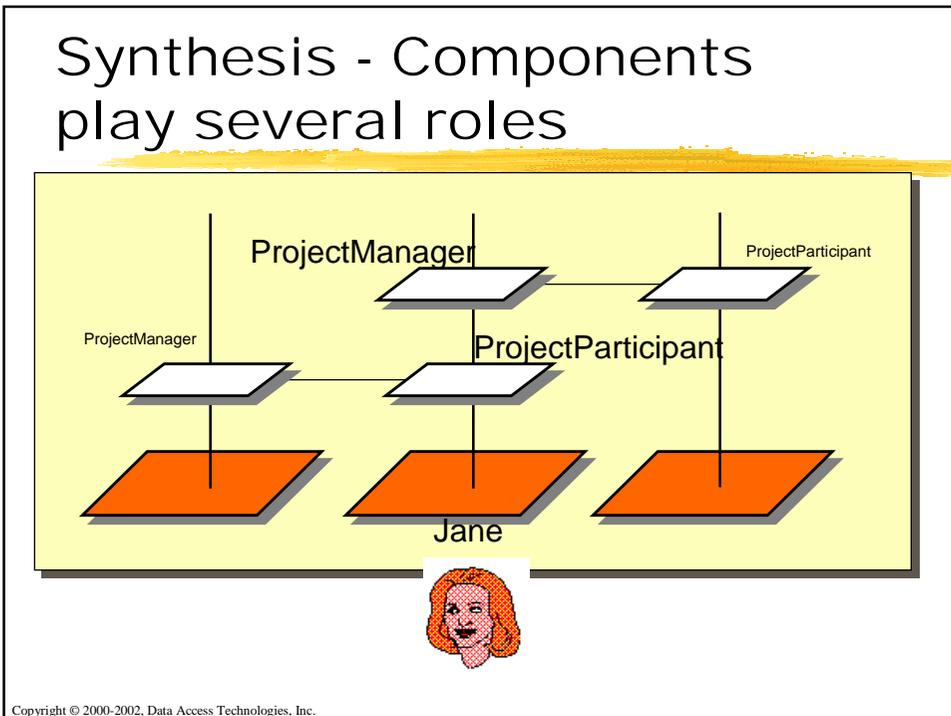
Discussion Point

- What are other ways of representing business semantics?
- What role do these play?
    - Business Processes
    - Information Models (DBMS)
    - Events
    - Workflow
    - Internet
- What is used internally?

Standards for Global Internet Computing

UNIFIED MODELING LANGUAGE™   EDOC

.NET   BPML

XLANG

ebXML
Creating A Single Global Electronic Market

OASIS

XML

WSDL

SOAP

XML-Schema

XML Standards

- XML Schema & DTD
  - Description and packaging of data
- WSDL
  - Specification a services, operations and flows available via that service
- Soap
  - Basic messaging and packaging
  - Extensions for Soap-RPC with WSDL
  - May be extended to support collaborative messaging

Vision

- EDOC (a UML Profile)
  - Provide an architecture for open collaborative computing
  - Simplify the development of component based distributed systems by means of a modeling framework, based on UML 1.4
  - Provide a platform independent, recursive collaboration based modeling approach supporting multiple technologies.
  - Embrace Model Driven Architectures (MDA) – Provide design and infrastructure models and mapping
- ebXML
  - Creating a single global electronic market
    - Includes process specification, transport and repositories
    - Collaboration Oriented

JSR 159

- Java Process Components
- Uses ECA (CCA) as the baseline component model
- Asynchronous document exchange
- Component composition
- A J2EE technology mapping for ECA
- Will make J2EE an excellent integration and web services platform

ebXML & EDOC

ebXML   EDOC

Transport

Distribution   Collaboration   Components

Repository   Process Model

Runtime

Enterprise Integration

Information Model

MDA

ECA as the normal form

MDA Mappings

EDOC-ECA

The standard way to model and tool for multiple technologies

Web Services (WSDL)

ebXML (BPSS)

J2EE (Java RMI)

.NET

MOM (MQ-Series)

Summary of points thus far

- We must enable the emerging Internet Computing Model
  - Loosely coupled roles exchanging documents based on a contract of collaboration
- Web need interoperability at two levels
  - Messaging for the data
  - Metadata for the contract of collaboration, stored in repositories
- This model of collaborating roles is recursive, extending into the enterprise, into managed domains and into applications
  - Inside the enterprise we want to include resources entities, business events and business processes
- Supporting the open domain has some required parts and can be augmented with a "treasure chest" of tools and infrastructure
- Between EDOC & ebXML we are covering B2B and intra enterprise

Back to EDOC

How to realize global
collaborations

EDOC Component
Collaboration Architecture

CCA

The model of
collaborative
work

The Marketplace Example

Order

Conformation

Process Complete

Mechanics Are Us
Buyer

Acme Industries
Seller

Status

Ship Req

Shipped

Delivered

Physical
Delivery

GetItThere Freight
Shipper

The Seller's Detail

Order

Conformation

Shipped

Order Processing

Shipping

Event

Ship Req

Shipped

Delivered

Receivables
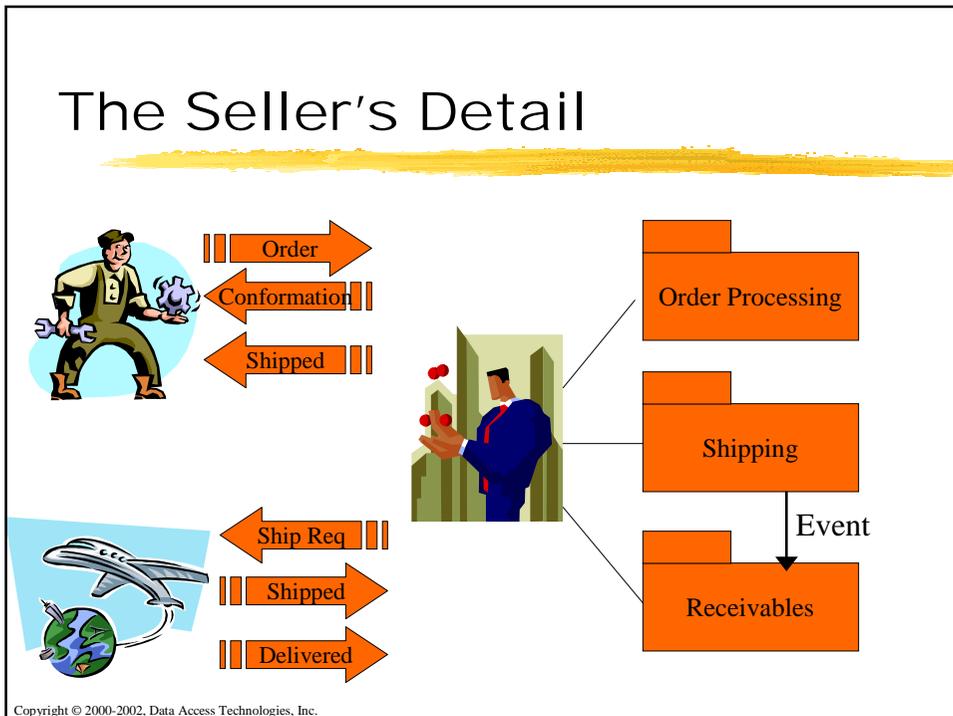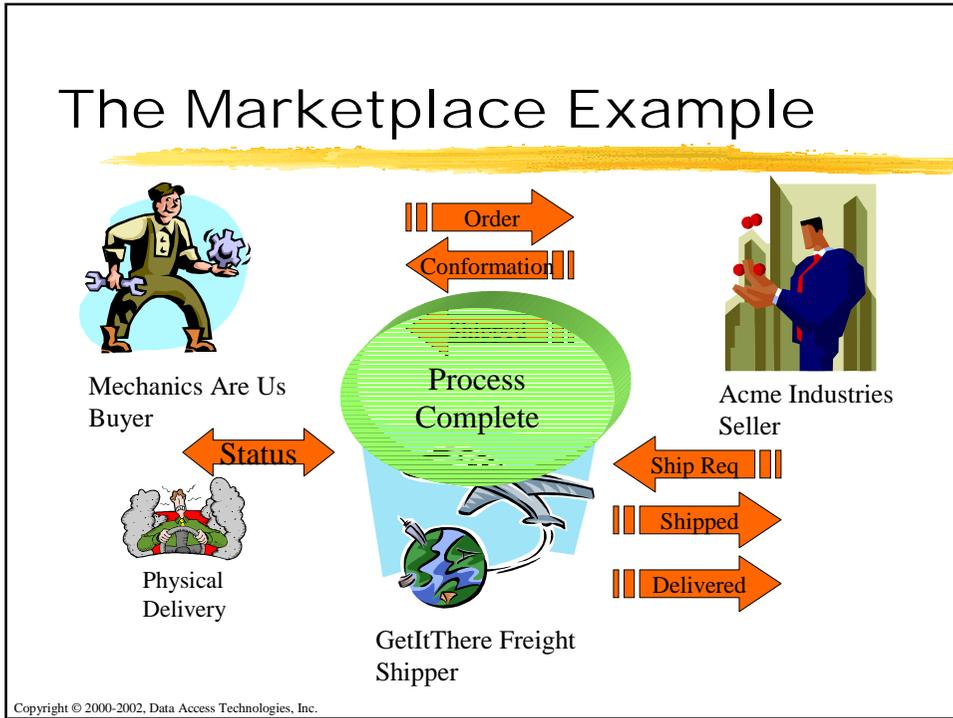
Parts of a CCA Specification

- ▌ Structure of process components and protocols
  - ▌ Process components, ports, protocols and documents
    - ▏ Class Diagram or CCA Notation
- ▌ Composition of process components
  - ▌ How components are used to specify components
    - ▏ Collaboration diagram or CCA Notation
- ▌ Choreography
  - ▌ Ordering of flows and protocols in and between process components
    - ▏ Activity Diagram

The Community Process

- ▌ Identify a "community process", the roles and interactions in a collaboration
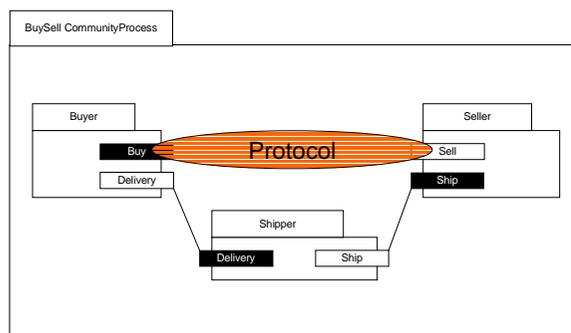
Protocols
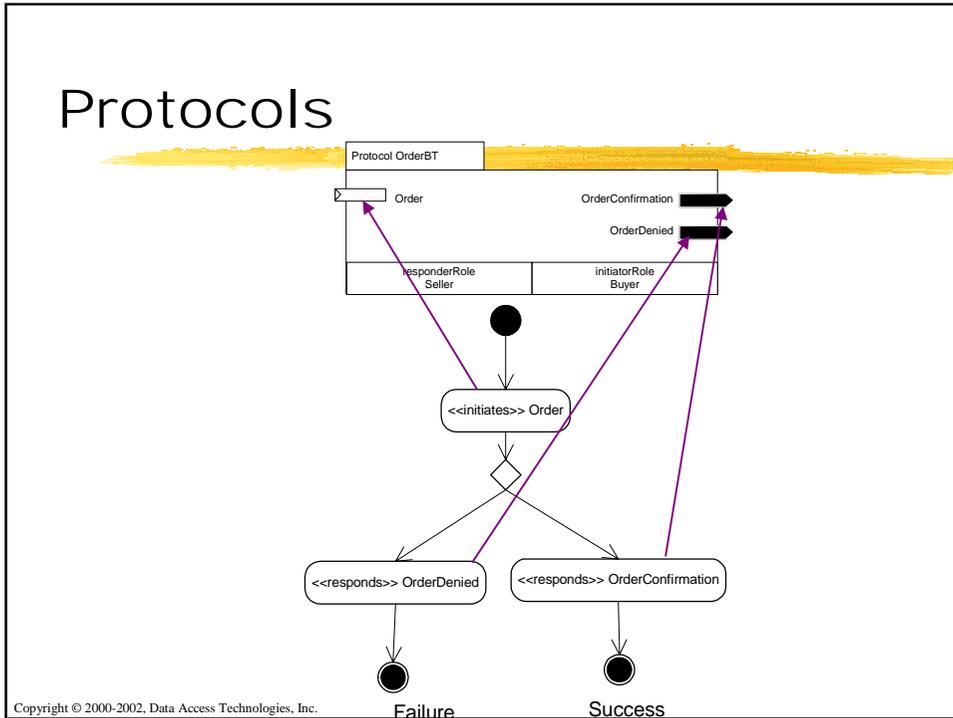
Composition
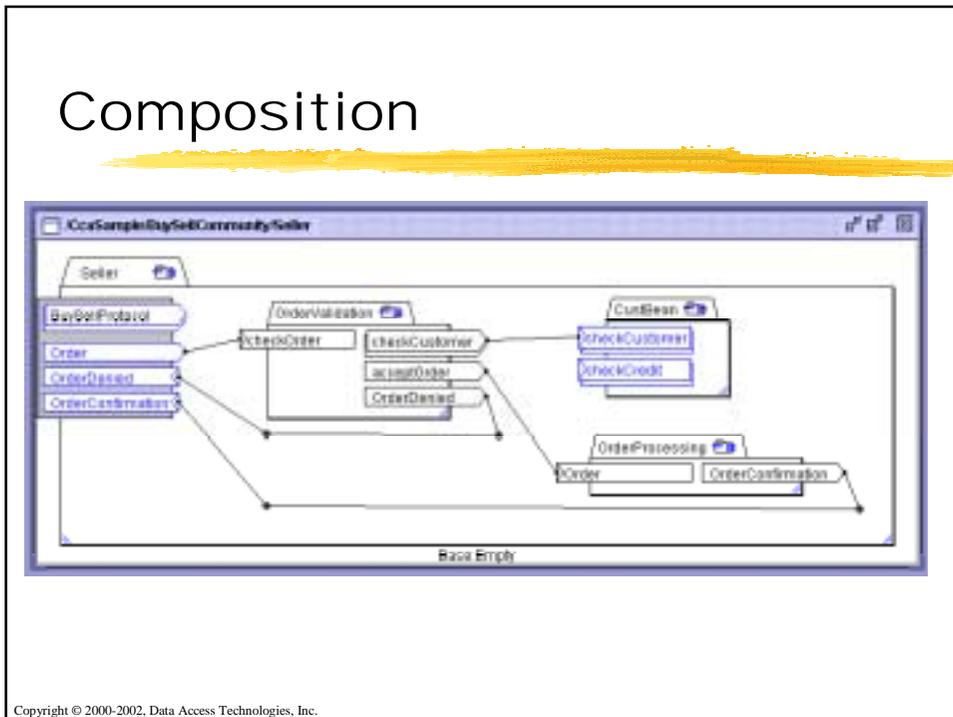
ECA Entity Profile

The model of things

Data Inside a "shared domain"

Adding Entities

- Entities are added to manage entity data
- Entity Roles are managers that provides a view of the same identity in another context
- The Entities have ports for managing and accessing the entities
- Non-entities which are owned by (aggregate into) an entity are managed by the entity

<<Entity>>
CompanyManager

Manage

Manages

«Key»
CompanyKey
+CompanyId : String

«EntityData»
Company
+Name : String
-CompanyId : String

«Key»
AccountKey
+AccountNo : String

«EntityData»
Account
+Name : String
+Balance : Decimal = 0
+AccountNo : long

+Cust +Adr

«EntityData»
Addtress
+Street : String
+City : String
+State : String
+Zip : String

-Manages

<<EntityRole>>
AccountManager

Manage
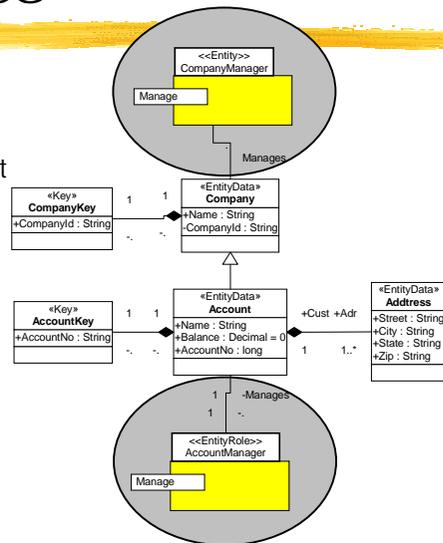
ECA Business Events

The model of when…

Loosely coupled integration within the enterprise and with "aligned" business partners

---

Event Based Business Processes
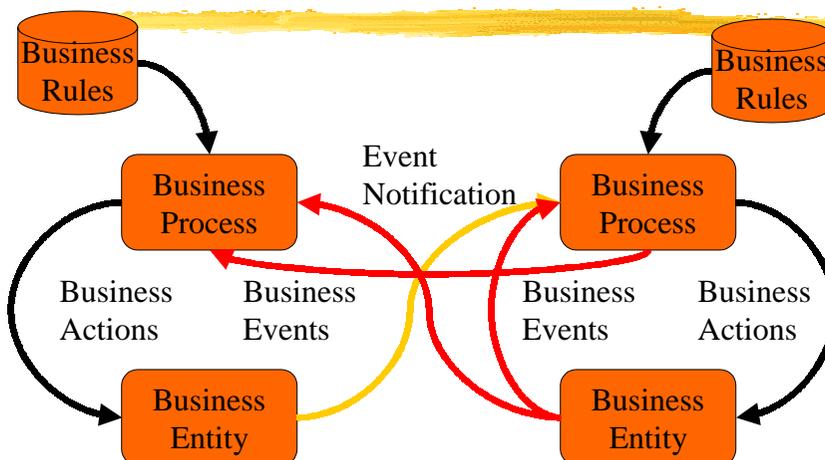
Business Rules

Business Rules

Business Process

Event Notification

Business Process

Business Actions

Business Events

Business Events

Business Actions

Business Entity

Business Entity

Point to point Event Notification

Pub/Sub Event Notification
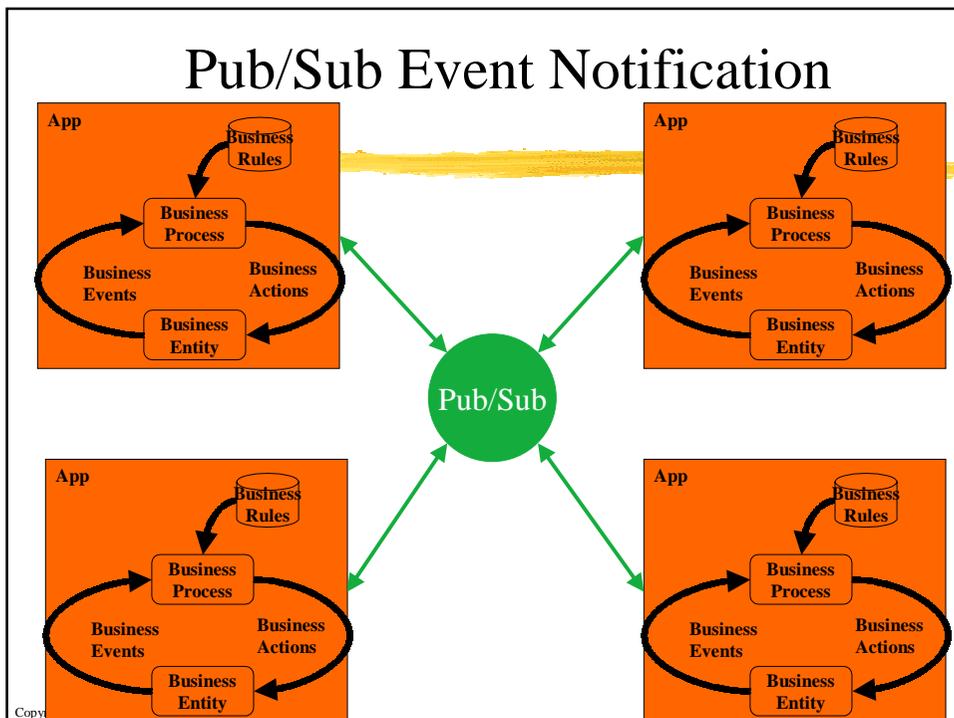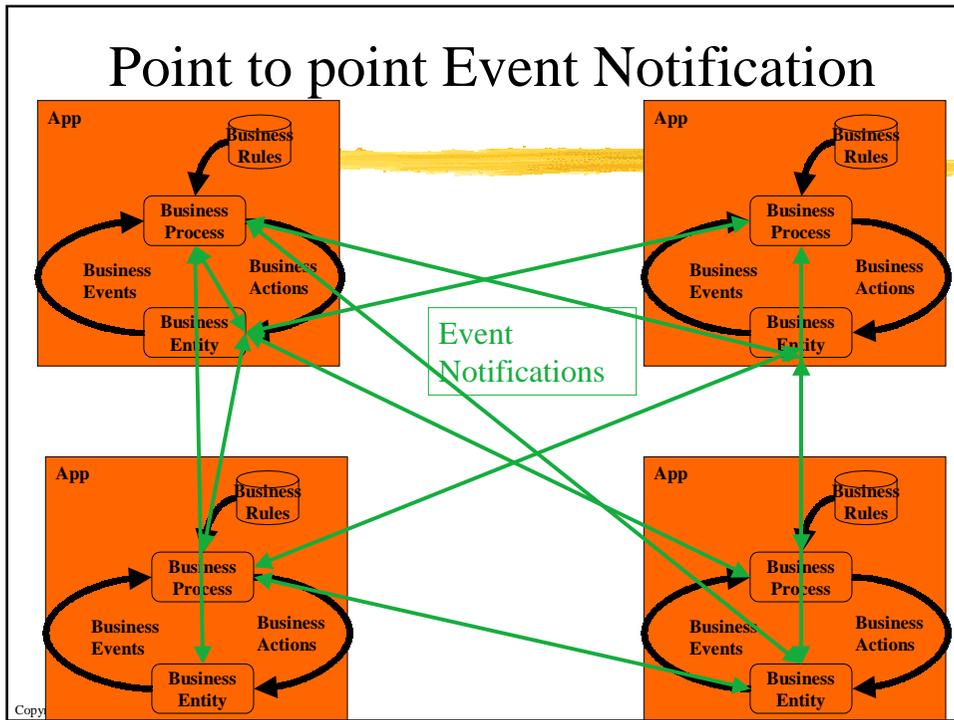
Event Example
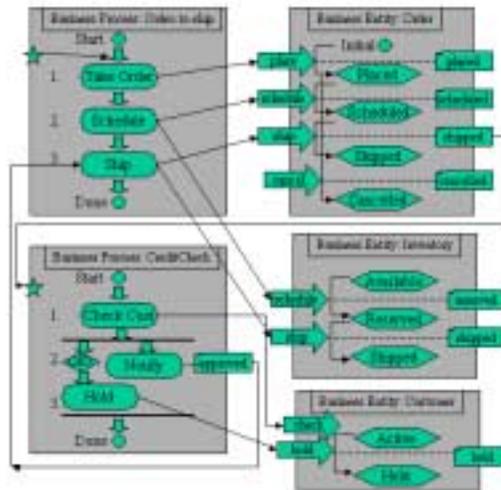
Events or Services

❚ Services - conversation between known parties
  ❚ Good supply chain model
❚ Events - Actions based on notification of events
  ❚ Parties do not know about each other!
  ❚ More loosely coupled
  ❚ Works best inside a managed domain (enterprise)

EDOC CCA Example

Buy-Sell Process

Community Process

Buy/Sell Comminity Process

: Buyer

: Buys

: Seller

: Sells
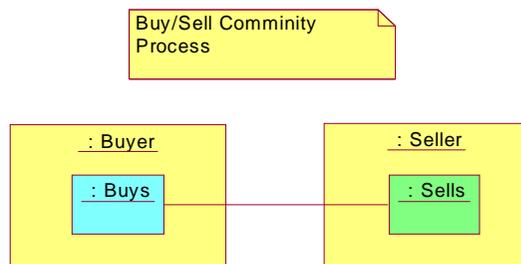
UML Collaboration Diagram

Community Process (CCA)
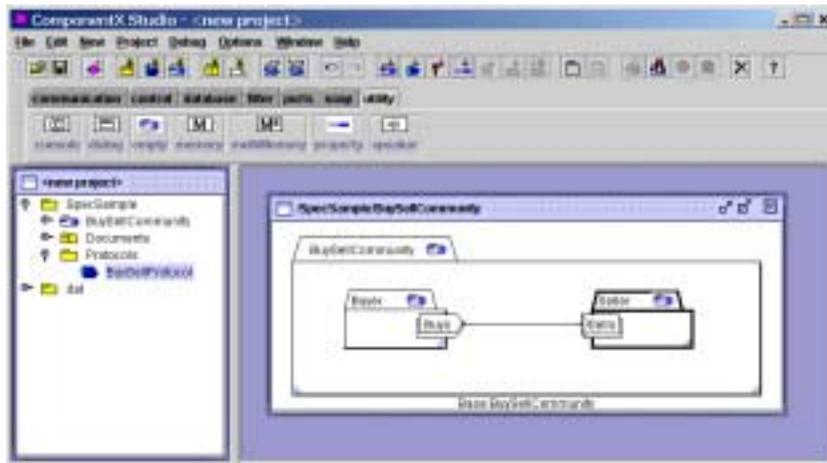
CCA Notation
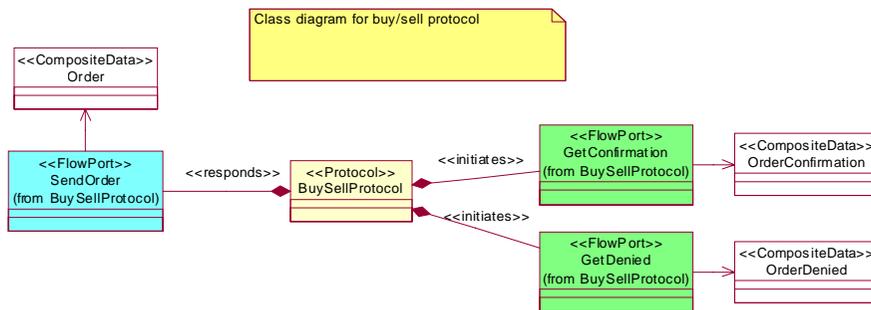
Protocol

Class diagram for buy/sell protocol

```
<<CompositeData>>
Order

<<FlowPort>>
SendOrder
(from BuySellProtocol)   <<responds>>   <<Protocol>>   <<initiates>>   <<FlowPort>>
                                        BuySellProtocol              GetConfirmation   <<CompositeData>>
                                                                     (from BuySellProtocol)   OrderConfirmation

                                                        <<initiates>>   <<FlowPort>>
                                                                        GetDenied   <<CompositeData>>
                                                                        (from BuySellProtocol)   OrderDenied
```

UML Class Diagram
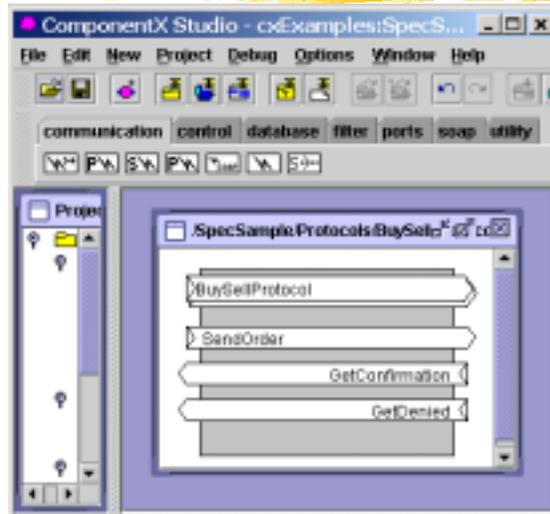
Protocol (CCA)

Define Documents
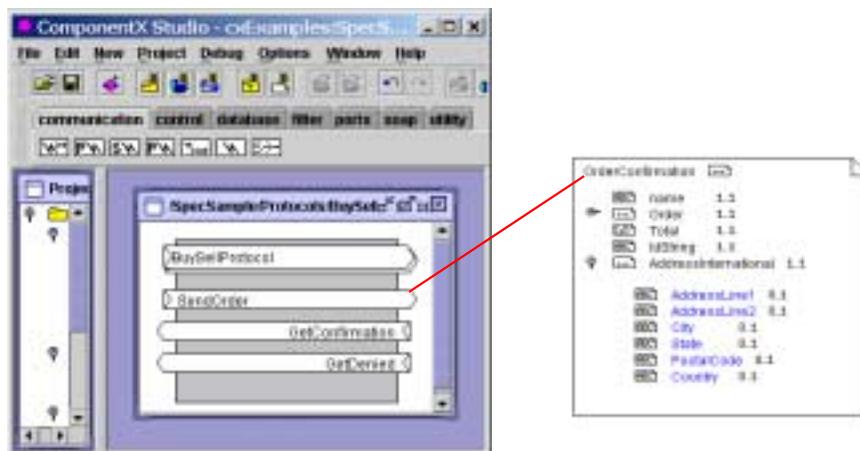
Protocol Choreography
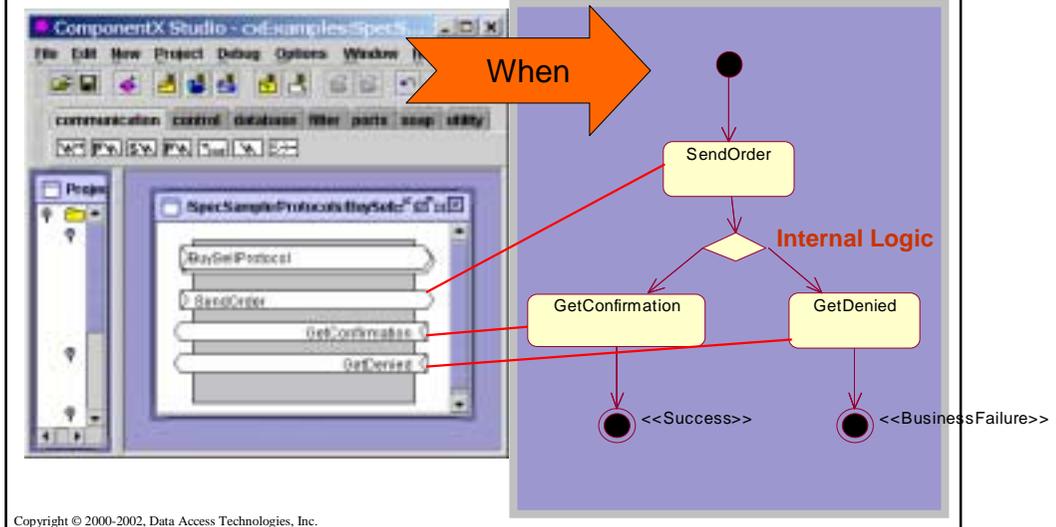
EDOC Exercise - Community

▌ Part one
  ▌ Pick a collaboration from the first exercise
  ▌ Model it with EDOC notation as a community process

▌ Identify an interaction
  ▌ Specify the documents of the interaction
    ▎ What are some of the major elements?
  ▌ Specify any choreography

Validation Component

UML Class Diagram

Validation Component (CCA)

Choreography
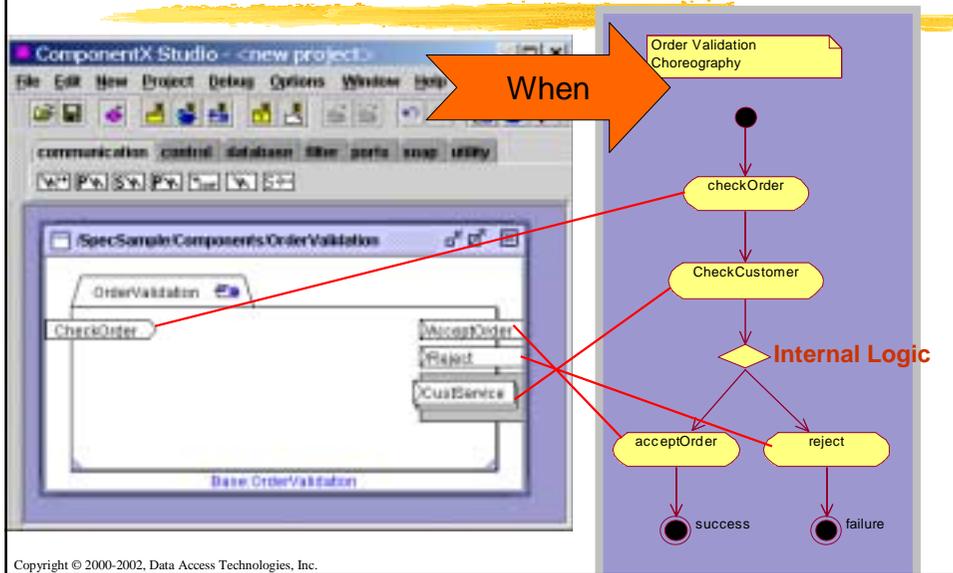
Copyright © 2000-2002, Data Access Technologies, Inc.

Composition (UML Collaboration)
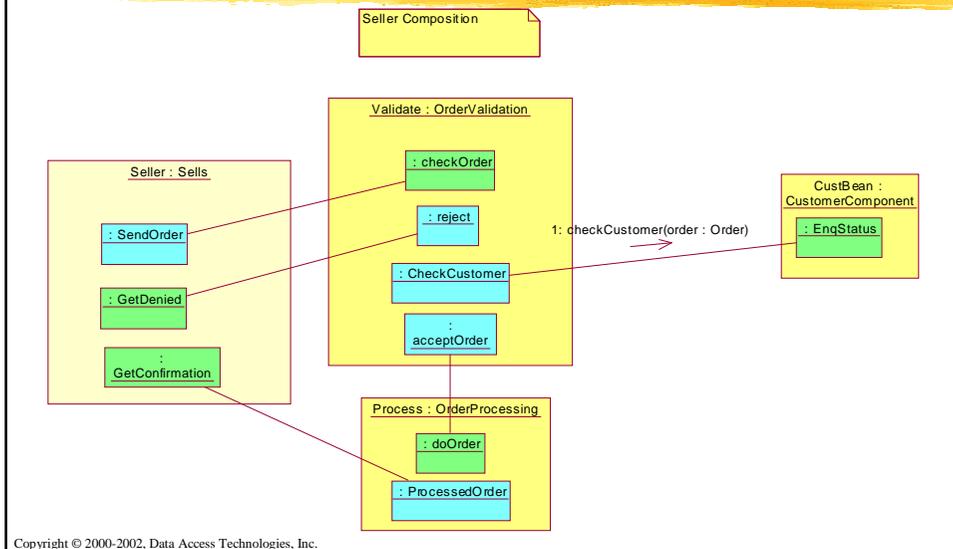
Copyright © 2000-2002, Data Access Technologies, Inc.

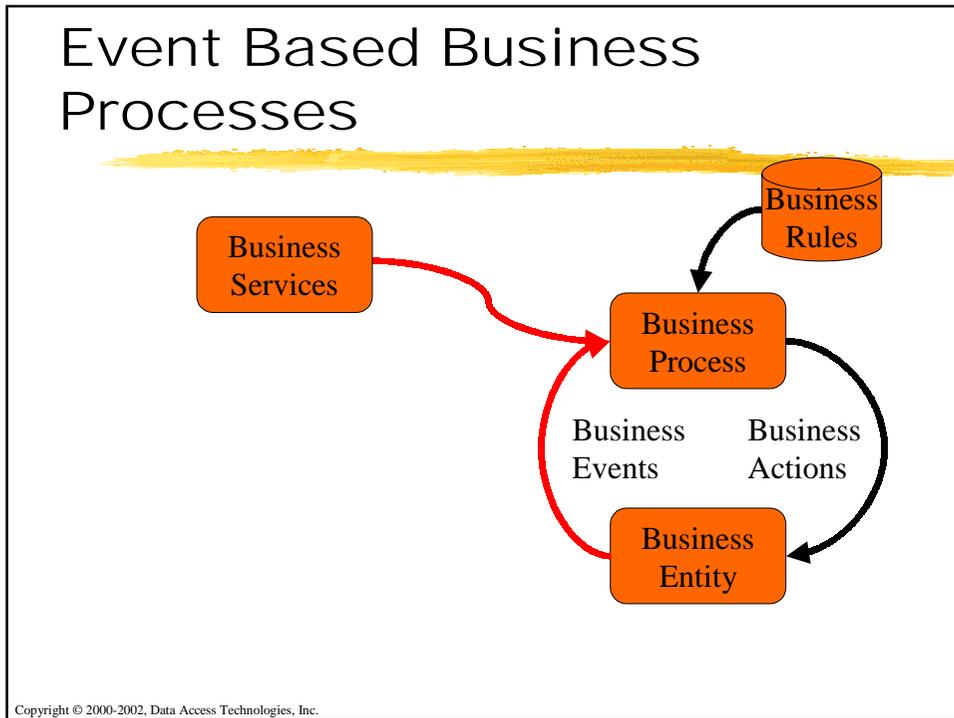# Composition (CCA)
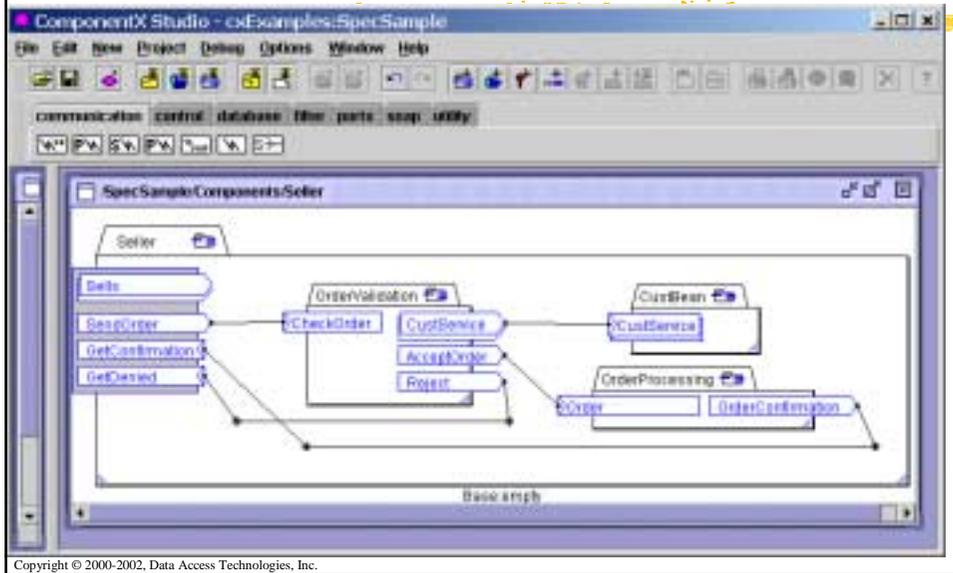
Sample Information Model
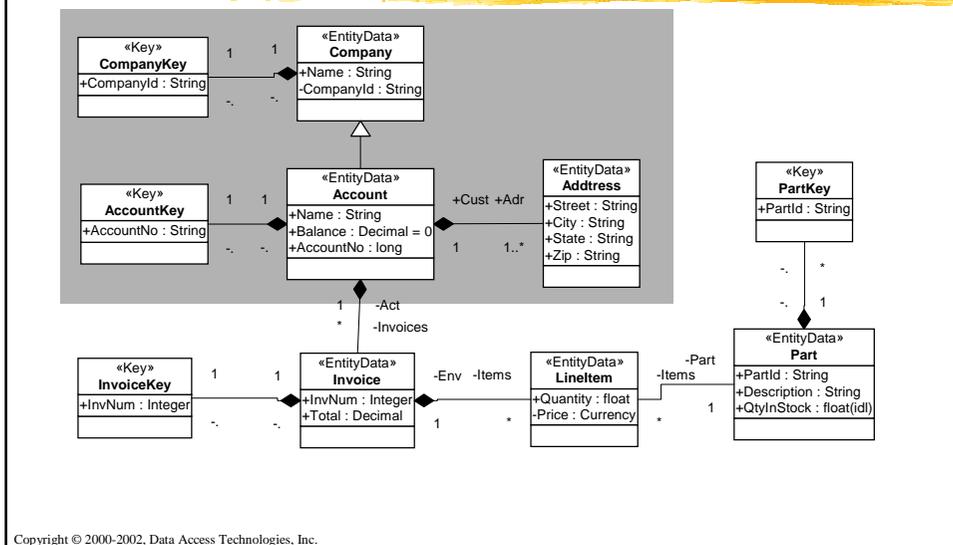
Adding Entities

- Entities are added to manage entity data
- Entity Roles are managers that provides a view of the same identity in another context
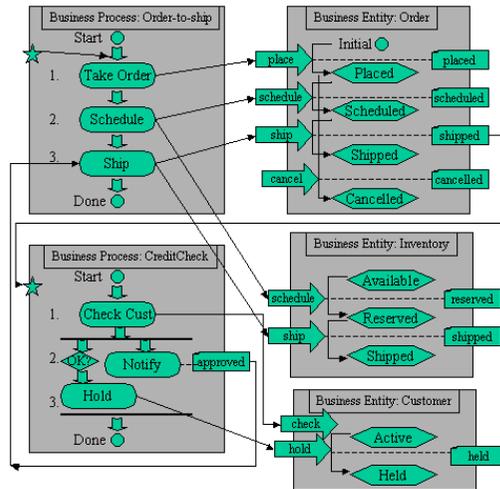- The Entities have ports for managing and accessing the entities
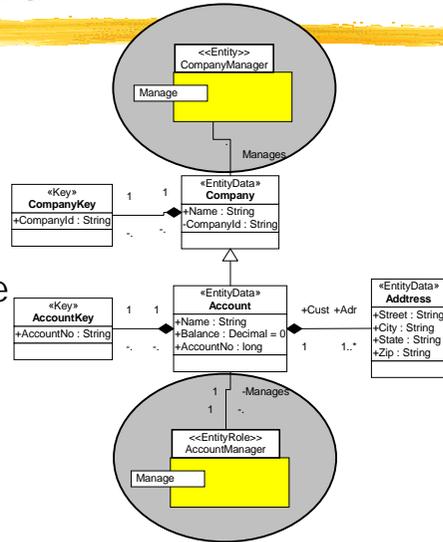
Modeling Events

What to do when something happens

Event Based Business Processes

Business Rules

Business Services

Business Process

Business Events   Business Actions

Business Entity

Copyright © 2000-2002, Data Access Technologies, Inc.

Event Based Business Processes

Business Rules

Business Rules

Business Process

Event Notification

Business Process

Business Actions   Business Events

Business Events   Business Actions

Business Entity

Business Entity
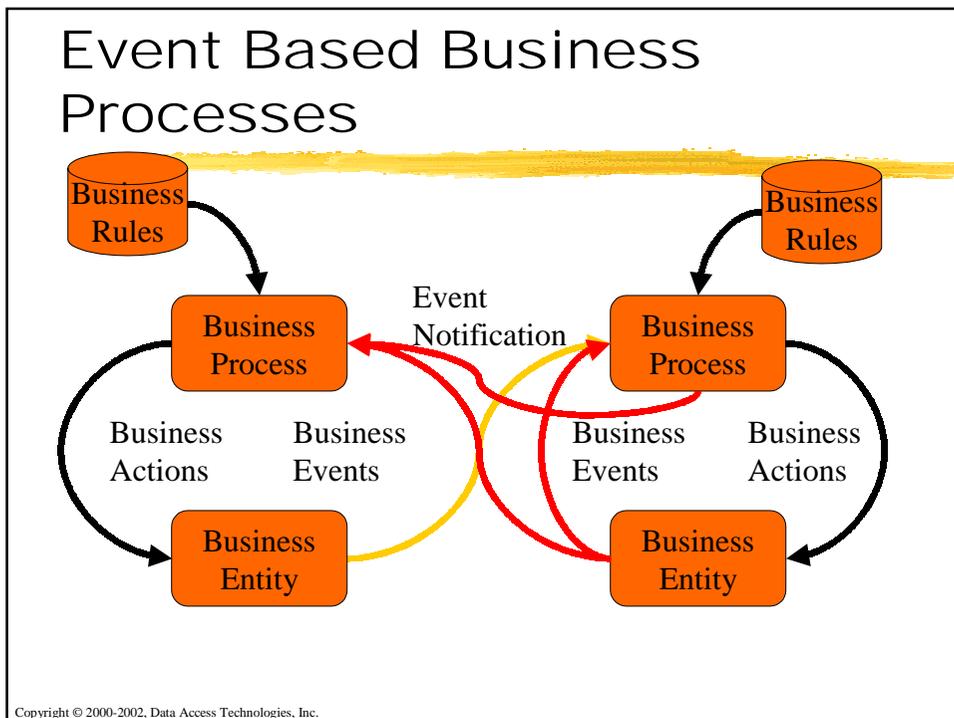
Copyright © 2000-2002, Data Access Technologies, Inc.

Event Example

Does ECA map well to WSDL?

An approach to the mapping

ECA/WSDL mapping

- ECA works well as a modeling framework for WSDL
- How major concepts could map
  - WSDL Port <-> ECA Port
  - WSDL Operation <-> ECA Flow port (one way) or Operation (Two way)
  - WSDL Service <-> ECA Component
  - WSDL Port type <-> ECA Protocol
  - WSDL Message <-> ECA Document type

WSDL/ECA Differences

- WSDL Adds
  - Technology binding and endpoints
- ECA adds
  - Choreography, nested conversations, two-way protocols, nested components.

The WSEC RFP

WSEC

Web Services for Enterprise Collaboration

This RFP has not yet been issued

Solicits proposals for

- ▌ A Mapping from EDOC-Enterprise Collaboration Architecture to WSDL 1.1 with attachments and a SOAP binding.
- ▌ A Mapping from WSDL 1.1 with a Soap binding to the EDOC-Enterprise Collaboration Architecture.
- ▌ Any required extensions to the EDOC-Enterprise Collaboration Architecture to represent WSDL semantics.

ECA Methodology

A simple methodology for creating collaborative bueiness processes

Basic Steps

- Identify roles and organize roles into collaborations
- Define collaboration documents
- Create basic business transactions
- Organize into protocols and events
- Use protocols to define ports on roles
- Drill-down into role detail
- Implement roles
- Configure implementations for deployment with technology specifics
- Deploy

Identifying roles and collaborations
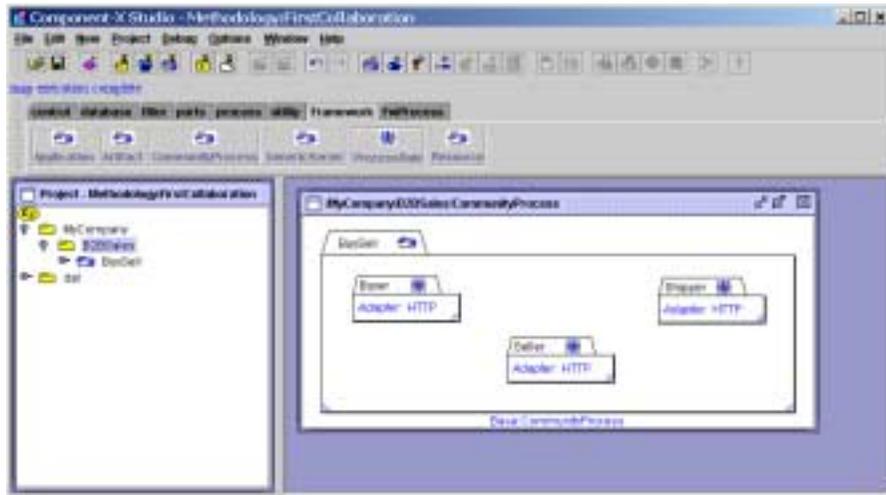
Identify Documents

Distinguish protocols and events
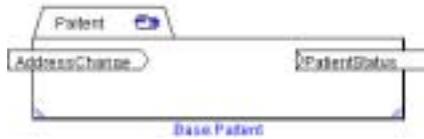
Create Business Transactions
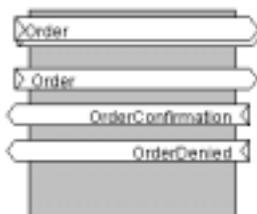
Organize into protocols
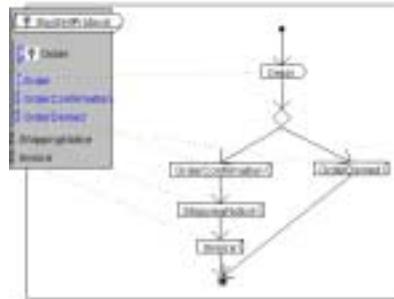
Add ports to complete community process

53

Drill-down
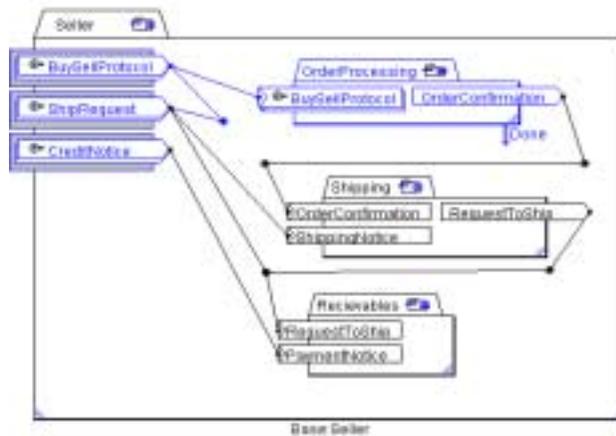
Add implementation

- As component compositions
- In a programming language
- By using an external service

Add technology specifics for deployment
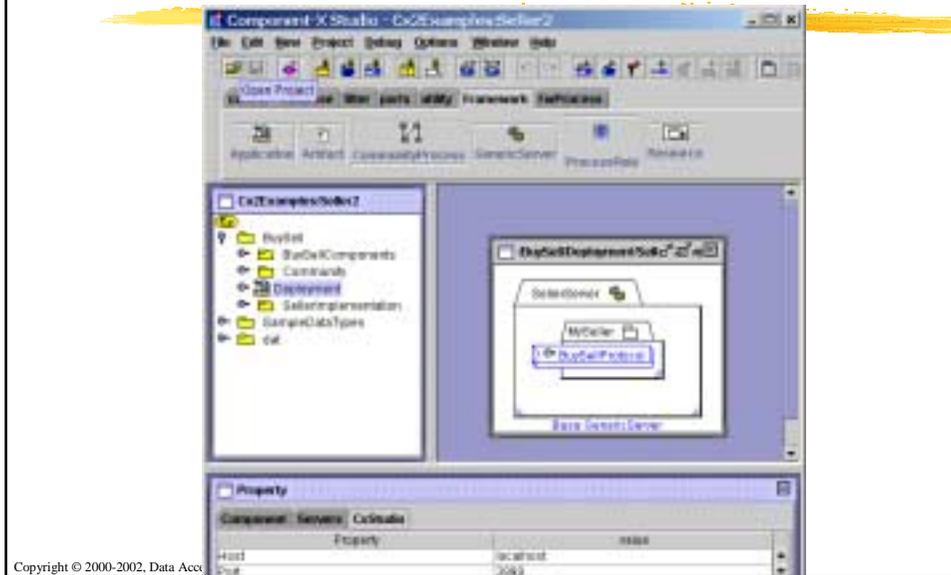
Complete process

- Implemented roles are deployed
- Completing the process

Vision

Building and adapting systems for collaboration, reuse and change

Loosely coupled enterprise architecture

- Independent enterprise components
- Representing business concepts
- Link via open standards

Open Standards

Open Standards

Open Standards

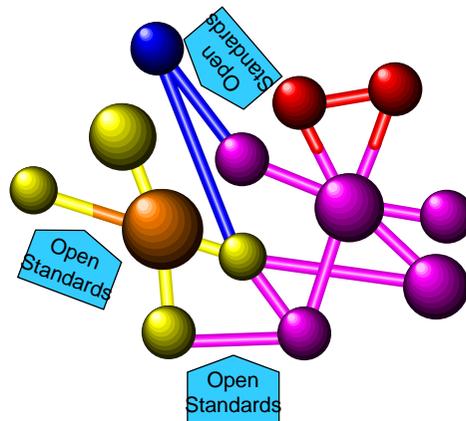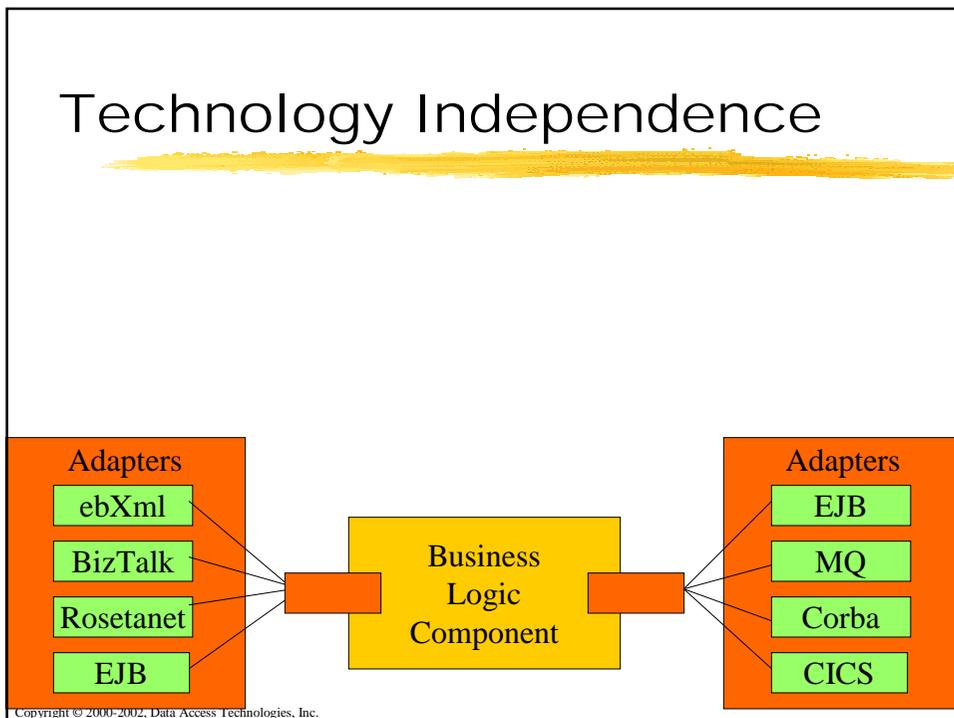# Business Component Marketplace

- The business component marketplace is projected to be a 10b market in 5 years
- Consider the value of XML components that wrap popular legacy
- New application functionality built from components
- Components for integration and transformation
- XML and web services makes an excellent basis for such components
- Technology components, such as for repositories and DBMS
- Marketplace my be inside the enterprise or commercial

OMG Model Driven Architecture (MDA)

- High level – platform independent models
- Technology Models
- Mapping
  - Custom
  - Standard
- Standard Models produce technology specific standards artifacts

Automated MDA

UML
Design

Profile
(E.G. EDOC)

Infrastructure
Mapping

(E.G. XML)

Tools
Produce &
Integrate

Enterprise
Components

Framework &
Infrastructure
(E.G. XML)

Manual
Coding

Mapping is tuned
to the infrastructure

Technology Independence

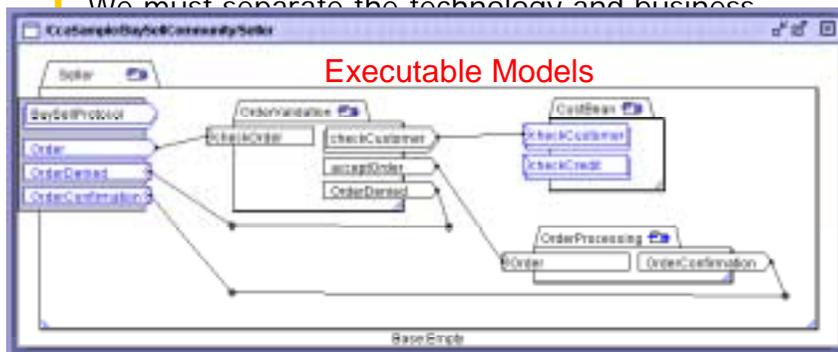| Adapters	Business Logic Component	Adapters
ebXml		EJB
BizTalk		MQ
Rosetanet		Corba
EJB		CICS

High level tooling & infrastructure

- ▌ MUST BE SIMPLE!
  - ▌ We must be able to create better applications faster
  - ▌ We must separate the technology and business concerns, enable the user
- ▌ Tooling + Infrastructure
  - ▌ Executable models are source code
  - ▌ Tooling must be technology aware
  - ▌ Infrastructure must support tooling, not manual techniques
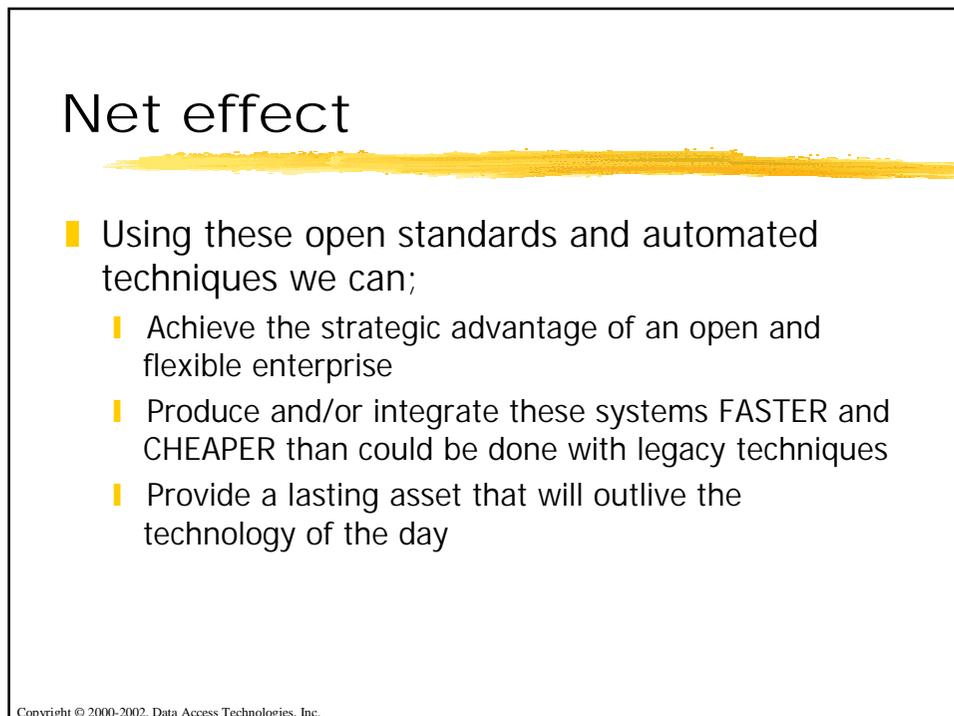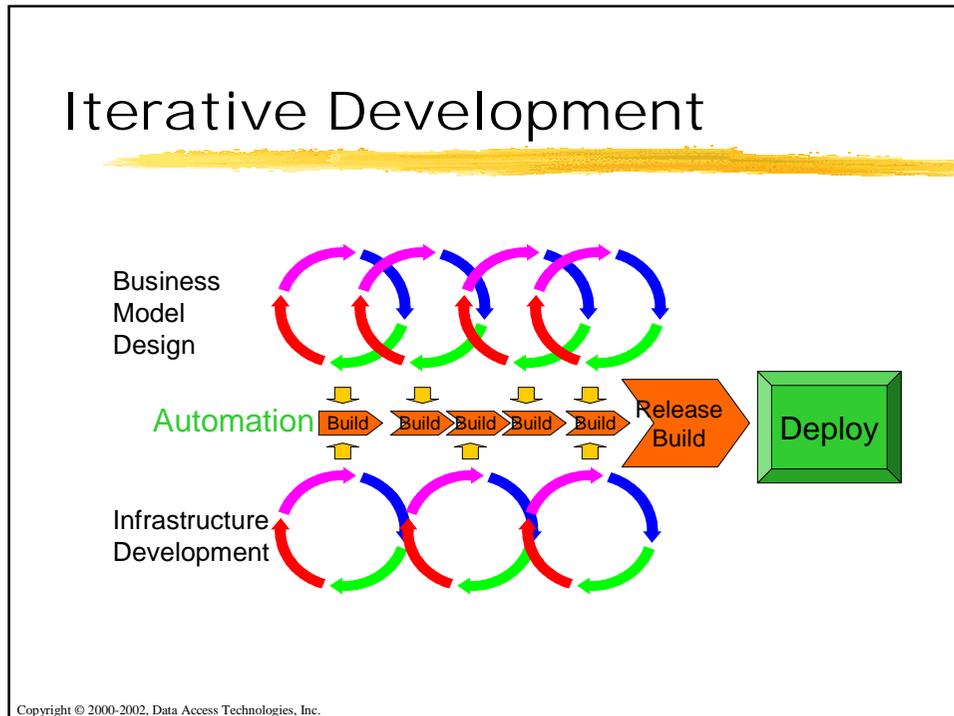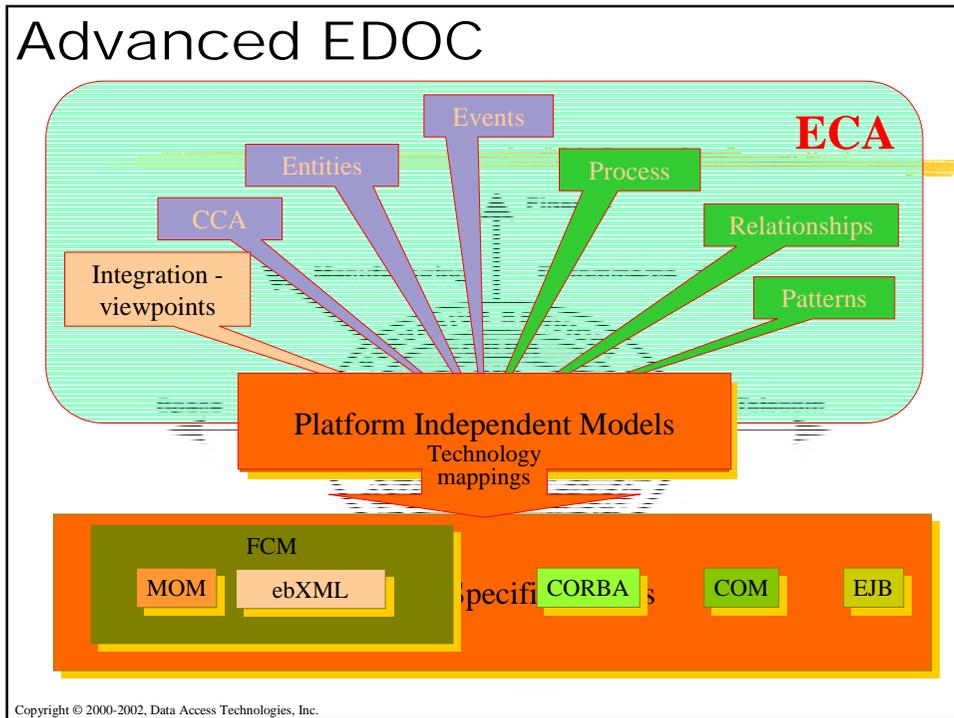- ▌ Model based component architectures

High level tooling & infrastructure

- ▌ MUST BE SIMPLE!
  - ▌ We must be able to create better applications faster
  - ▌ We must separate the technology and business

Iterative Development

Business
Model
Design

Automation Build Build Build Build Build Release Build Deploy

Infrastructure
Development

Net effect

❚ Using these open standards and automated techniques we can;

   ❚ Achieve the strategic advantage of an open and flexible enterprise

   ❚ Produce and/or integrate these systems FASTER and CHEAPER than could be done with legacy techniques

   ❚ Provide a lasting asset that will outlive the technology of the day

Advanced EDOC

Events

ECA

Entities

Process

CCA

Relationships

Integration - viewpoints

Patterns

Platform Independent Models
Technology mappings

FCM

MOM   ebXML   Specifi...   CORBA   COM   EJB

Business Process Profile

how things are coordinated

Business Processes

- Specialize CCA
- Activity-centric view of a Process
- Express
  - Complex temporal and data dependencies between business activities
  - Iteration of activities
  - Alternative required Inputs and Outputs of activities
  - Roles related to performers, artifacts and responsible parties for activities
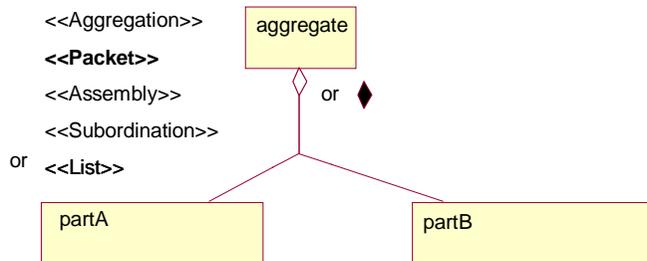
Relationships Profile

Useful associations and dependencies

Relationships Profile

- Enables non-binary aggregations
- Defines useful association and dependency stereotypes

<<Aggregation>>
<<Packet>>
<<Assembly>>   or
<<Subordination>>
or **<<List>>**

aggregate

partA

partB

Patterns Profile

reusing
parameterised
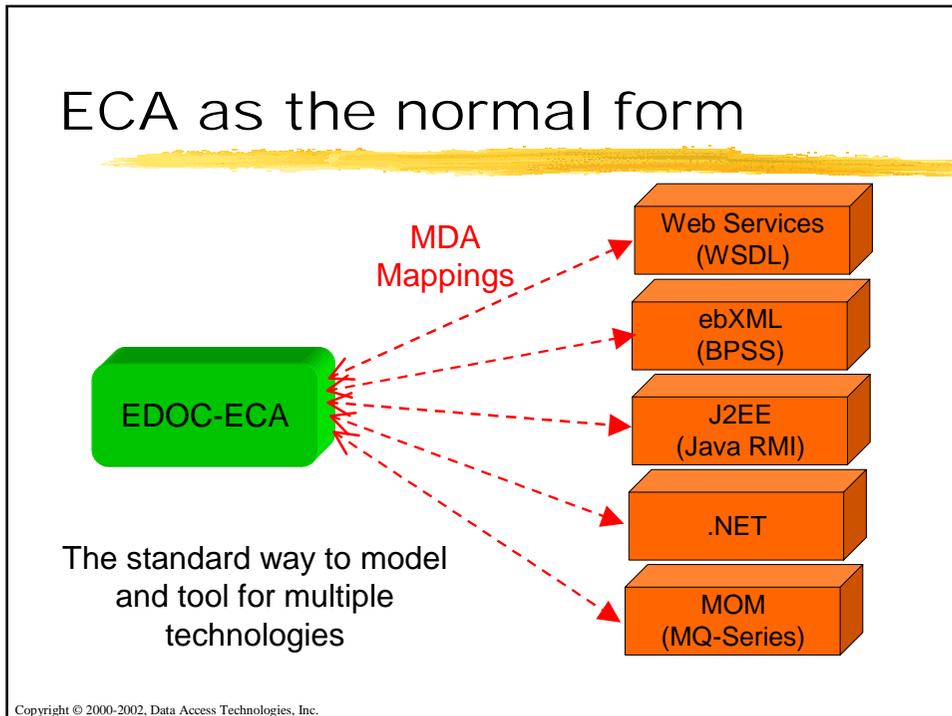designs

Patterns Profile

- Profiles UML Parameterized Collaborations
- Based on Business Function Object Patterns (BFOP)
  - Multi-Layer
  - Based on Catalysis Approach
- Adds stereotypes for
  - Named Patterns
  - Inheritance
  - Composition
  - Pattern Binding with renaming

Platform Specific Modelling

▌ EJB, FCM, MOF

▌ Technology mappings from EDOC to Distributed Component and Message Flow Platform Specific Models

  ▌ EDOC to J2EE/EJB mapping
  ▌ EDOC to CORBA/CCM mapping
  ▌ EDOC Business Process to FCM mapping
  ▌ EDOC Business Process to CORBA mapping

ECA as the normal form

MDA Mappings

EDOC-ECA

Web Services (WSDL)

ebXML (BPSS)

J2EE (Java RMI)

.NET

MOM (MQ-Series)

The standard way to model and tool for multiple technologies

Discussion

❚ How much are we using these kinds of concepts today?

❚ What would be the advantages to going in this direction?

❚ What are the barriers;
  ❚ Technical?
  ❚ Political?

❚ What would be a good first step?

Contact

Cory Casanave

Data Access Technologies

www.enterprise-component.com

cory-c@enterprise-component.com

(305) 234-7077