Realizing the Benefits of MDA

OptimalJ by Compuware
May 2003 OMG Workshop

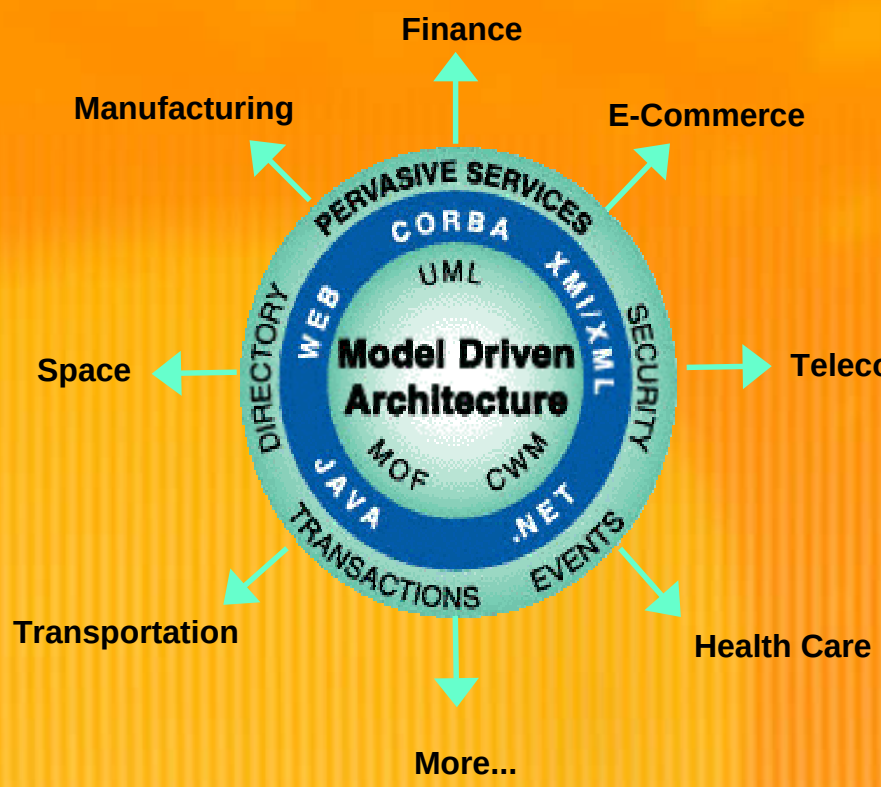
CompuwareCorporation

Chris Sirosky
Product Manager - OptimalJ

Business Challenges
In Java/J2EE application development

- **Rapidly respond to business change – Time to market**
- **Increase developer productivity – "Do more with less"**
- **Enforce the use of best practices, standards and guidelines**
- **Stay current with new versions of technology**
- **Leverage existing investments**
- **Maximize application quality and reliability – Minimize risk**

Object Management Group
Model-Driven Architecture

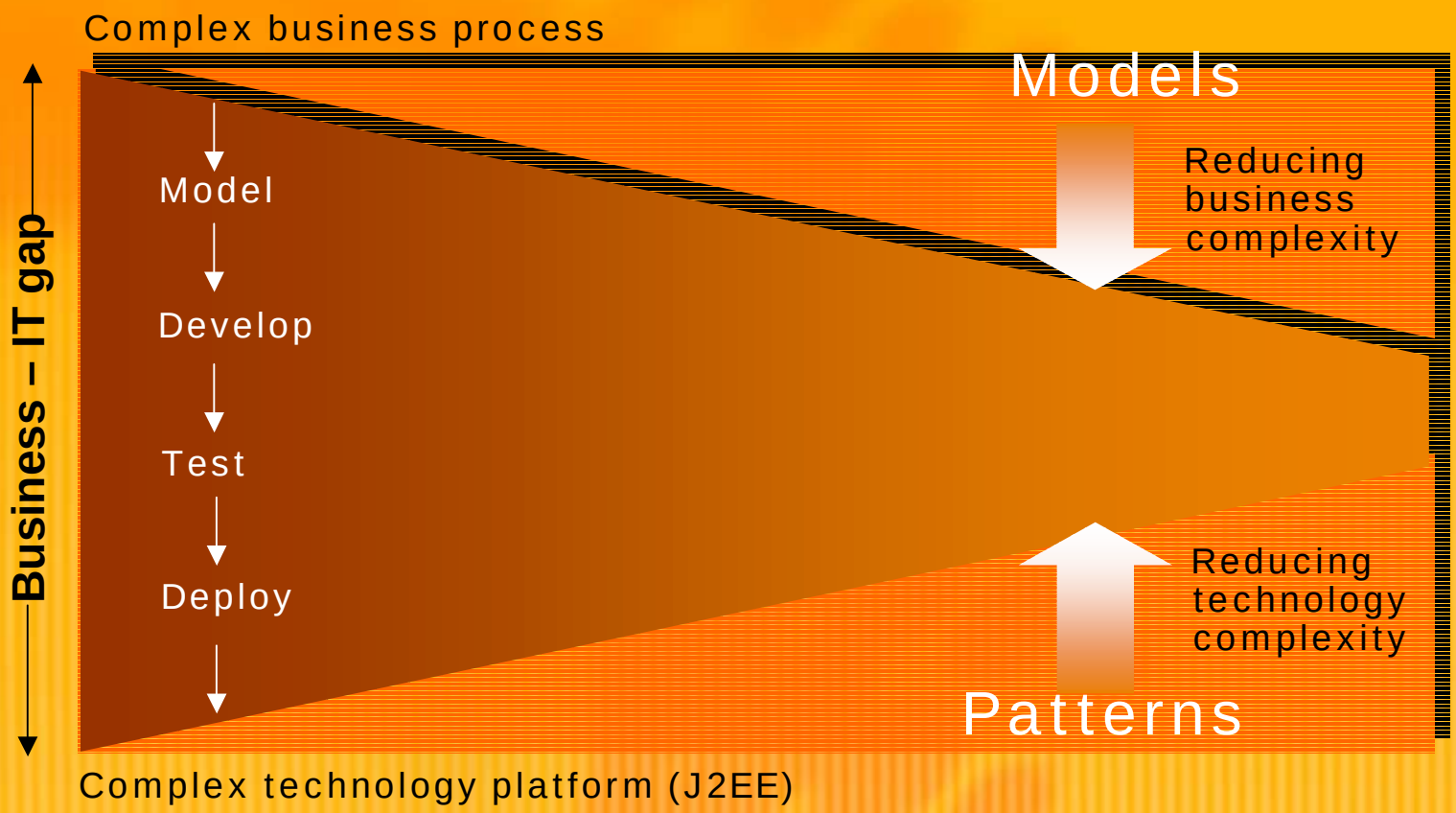


- **A new way to specify and build systems**
  - **Based on modeling with UML**
  - **Modeling instead of programming**

"MDA is about moving to the NEXT level – instead of offering another set of API's, it offers an opportunity to model those API's generically – and then generate code for accessing the API's you choose (or get stuck with). More importantly from the view of the app developer, you model *your application* generically, ignoring the plethora of APIs" *Richard Soley*

Closing the Gap

The need for Models and Patterns



A New Development Paradigm
Model-driven pattern-based

- **Companies that want to maintain or increase their future competitive edge will need to begin evaluating, planning for and migrating development staff to at least one of the two alternative and more efficient forms of development, model-driven pattern-based (MDPB) or component assembly and orchestration (CAO).**

- **Organizations using a model-driven or pattern-based application development framework containing a large inventory of business components have the potential to be five to ten times more productive and responsive than those that do not**

Giga Information Group®

John Meyer
Sr. Industry Analyst

Gartner Inc.

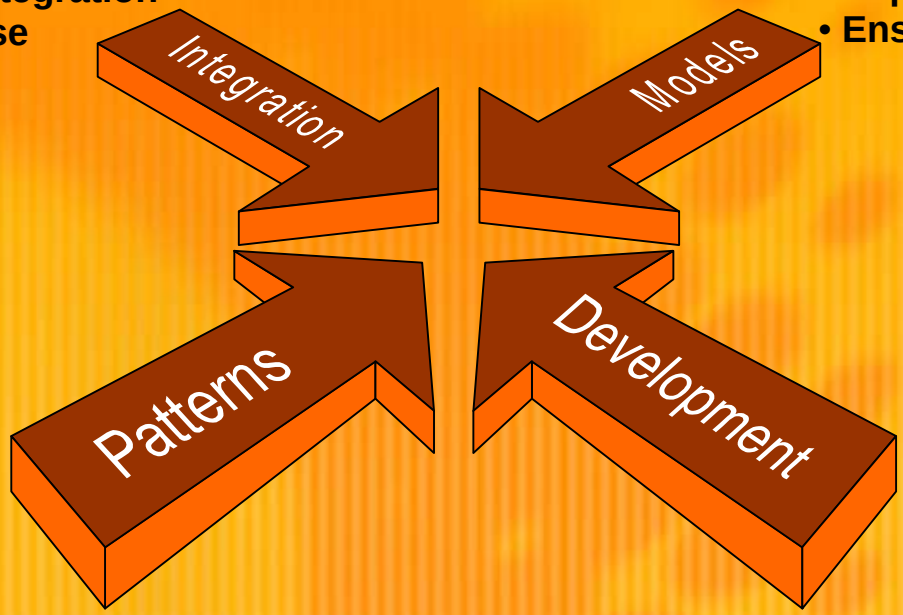
Michael Blechar
VP Internet and e-Business Technologies

OptimalJ

Development & Integration with Patterns & Models

Integration to:
- **Reduce integration complexity**
- **Accelerate integration**
- **Promote reuse**

Models to:
- **Reduce business complexity**
- **Rapidly respond to change**
- **Ensure reuse**

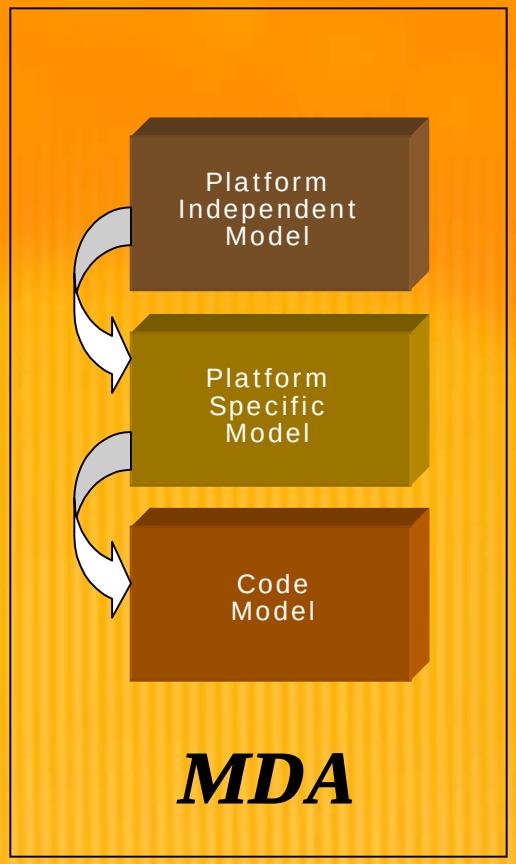


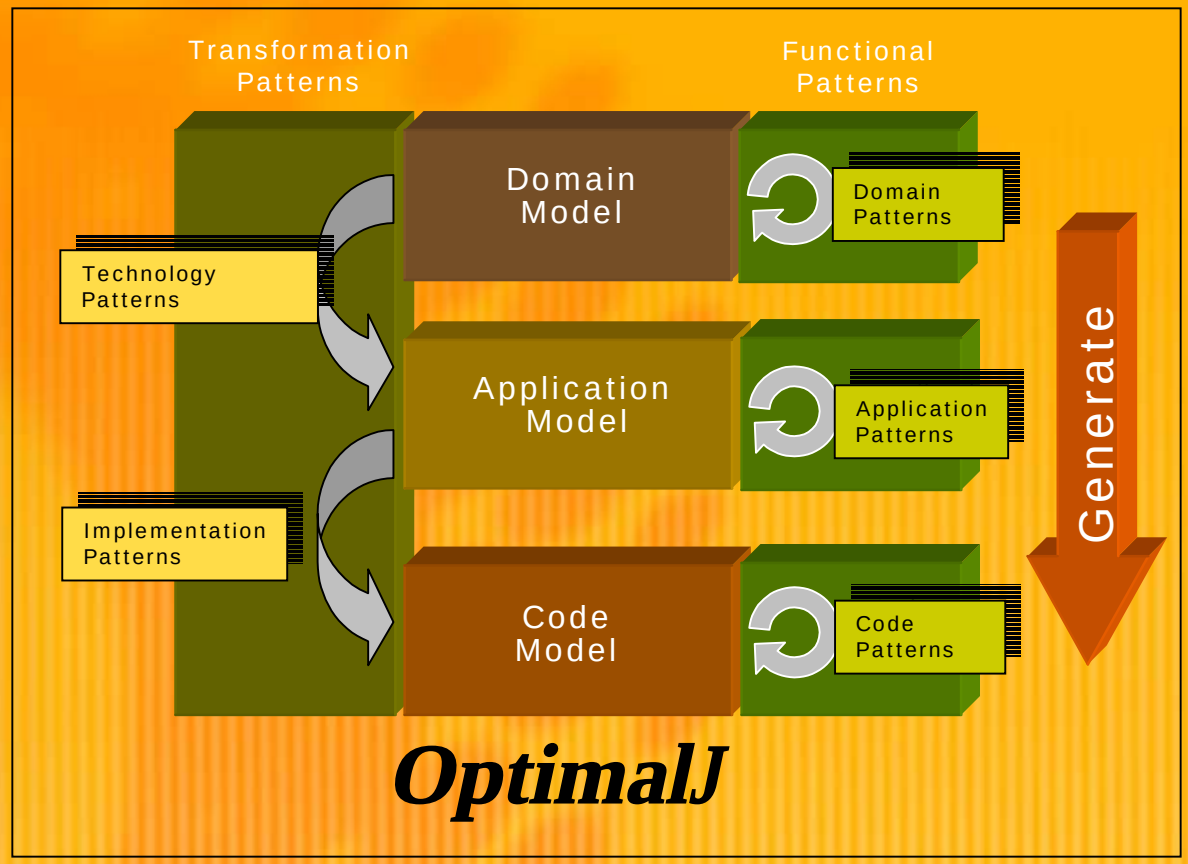
Patterns to:
- **Reduce technology complexity**
- **Accelerate development**
- **Enforce standardization**

Development to:
- **Improve customization**
- **Allow personalization**

Model-driven Development

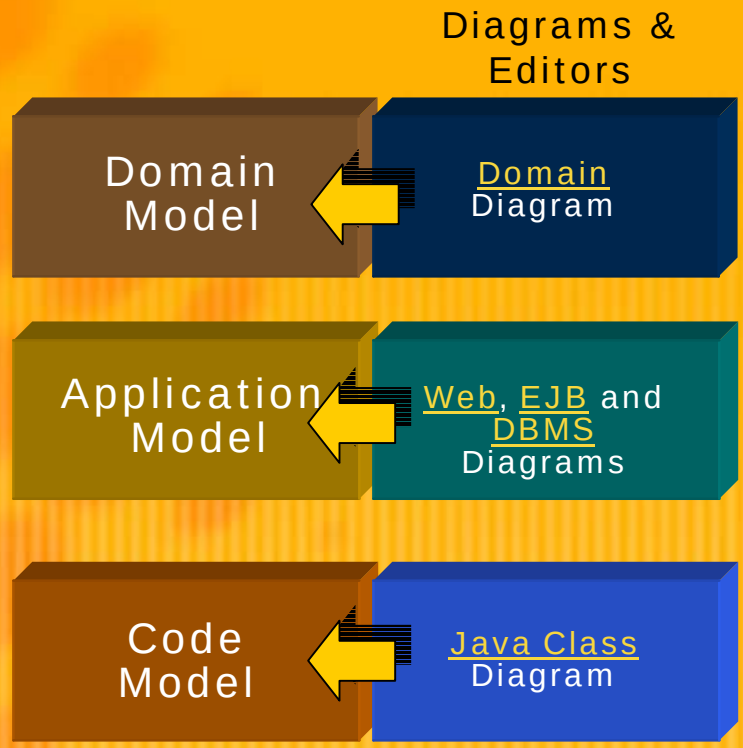
How OptimalJ maps to MDA





Visual Development Paradigm
Models and diagrams

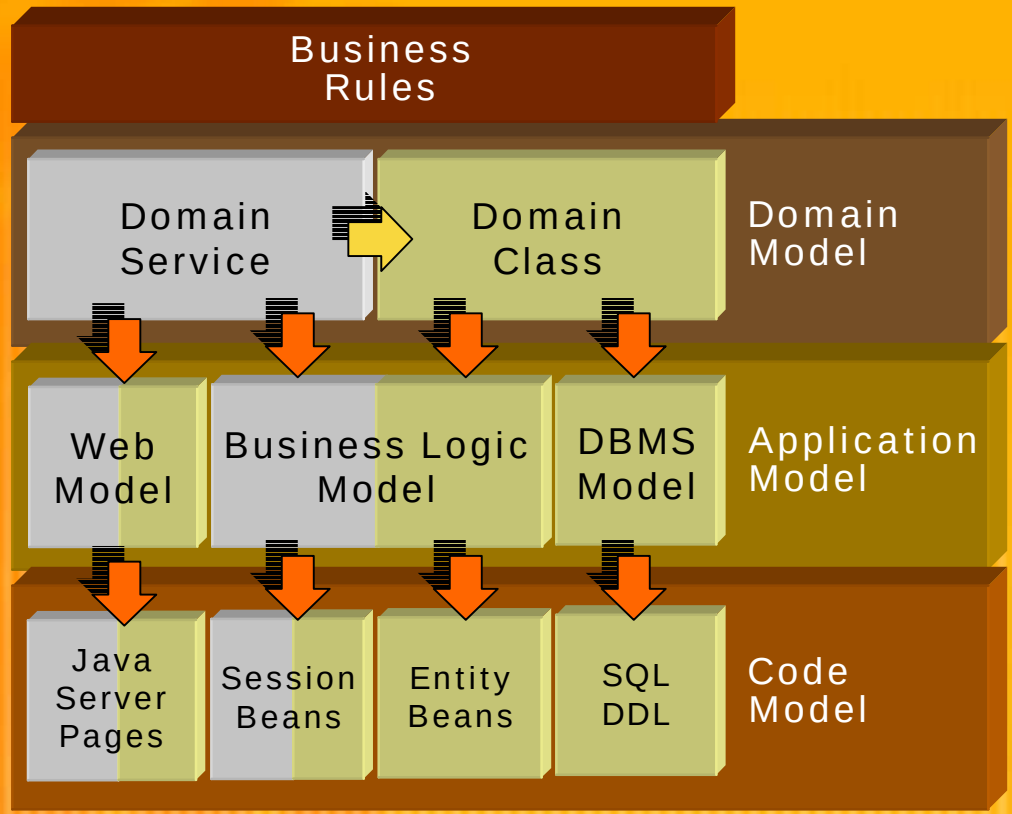
- **Domain Model diagram**
  - Created by the designer
  - Imported from a modeling tool (UML/XMI)
  - Uploaded from an existing database
- **Application models**
  - Generated automatically
  - Improve application navigation
  - Improve application overview and understanding
- **Code Model**
  - One-to-one mapping to Java code
  - Visualization of the code
  - Reengineering supported



OptimalJ

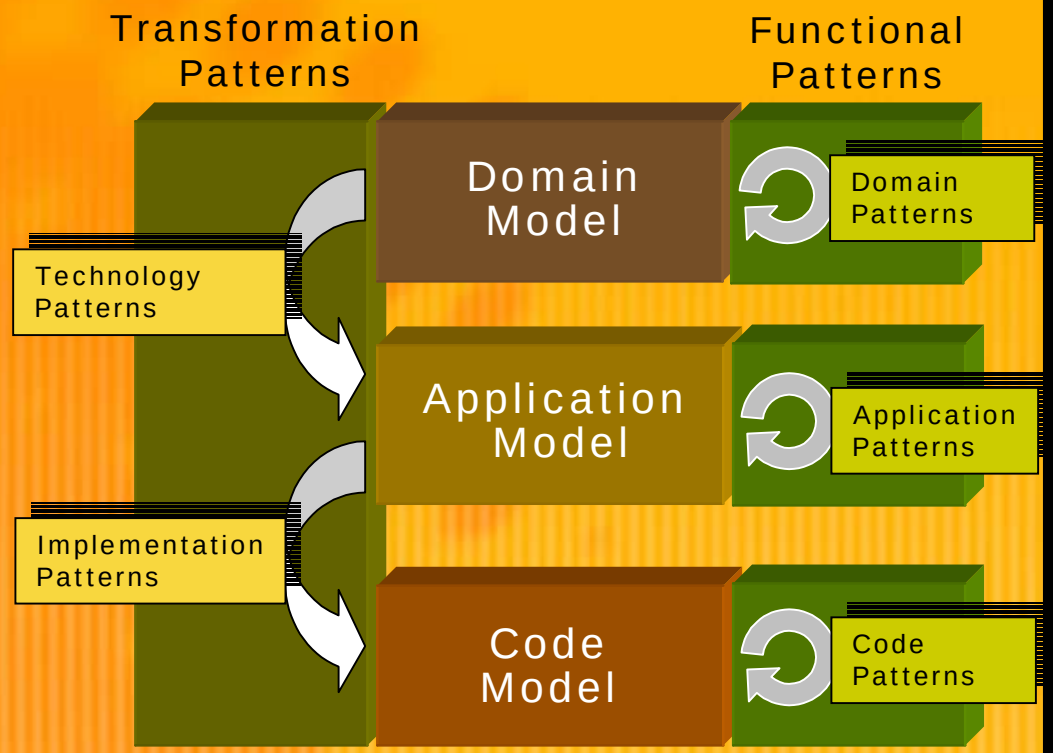
Pattern-driven Application Generation

- **Rapidly generate a J2EE application from a visual model**
- **Produces consistent, high-quality, reliable Java code**
- **Automates tedious, repetitive development tasks**
- **Patterns shield the designers from the complexity of the J2EE platform**



OptimalJ Patterns
Transformation & Functional Patterns

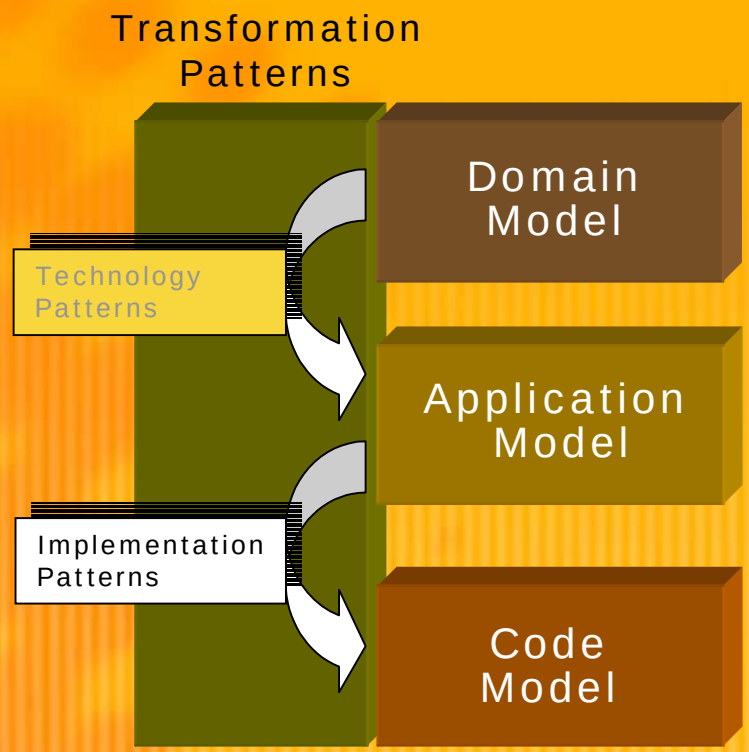
- **Transformation Patterns – Automation**
  - **Reduce complexity**
  - **Automatically transform high level model into lower level model.**
    - **Technology Patterns**
    - **Implementation Patterns**
- **Functional Patterns – Reuse**
  - **Accelerate development**
  - **Reuse best practices or implement standards**
    - **Domain Patterns**
    - **Application Patterns**
    - **Code Patterns (GoF)**



Implementation Patterns

Pattern Editor

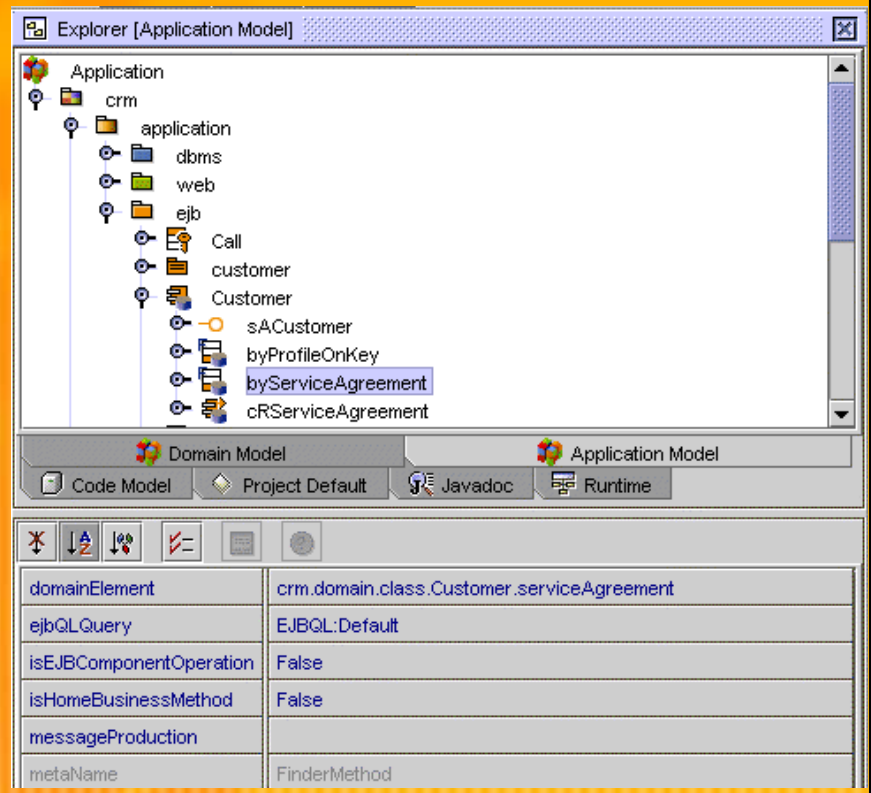
- **Experienced J2EE architects can add, update and delete Implementation Patterns**
- **Allows architects to implement their internal coding standards**
- **Ensures designers/developers implement the standards**
- **Reduces the need for code reviews and training**
- **Automatically transfers expert knowledge to designers/developers**



Pluggable Pattern Architecture

For example upgrading from EJB 1.1 to EJB 2.0

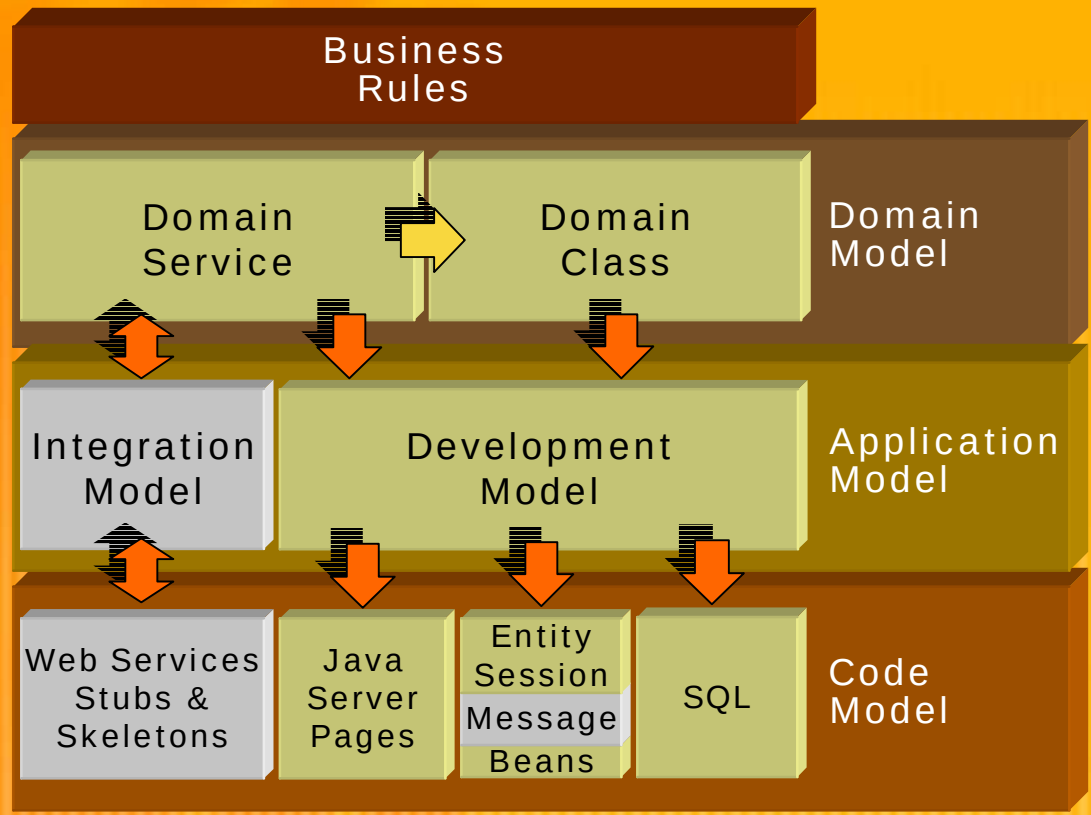
- **OptimalJ's pluggable pattern architecture**
  - **Install OptimalJ's EJB 2.0 pattern**
  - **Regenerate the application**
- **Automatically implemented EJB 2.0 features**
  - **EJB 2.0 attribute declarations**
  - **Container Manager Relationships (CMR)**
  - **EJB Query Language (EQL) on Finder Methods**
  - **Deployment descriptors**
  - **Custom code preserved**



Application Integration

Java, .NET and legacy Cobol

- **Integrate with existing Java, .NET and legacy applications**
- **Import WSDL, IDL, Cobol Copybooks or Commarea and Java signatures**
- **Automatically generate Web Services**
- **Automatically generate JCA compliant stubs and skeletons**
  - **Integrate with existing JCA compliant resource adapters such as IBM's CICS**



OptimalJ from Compuware

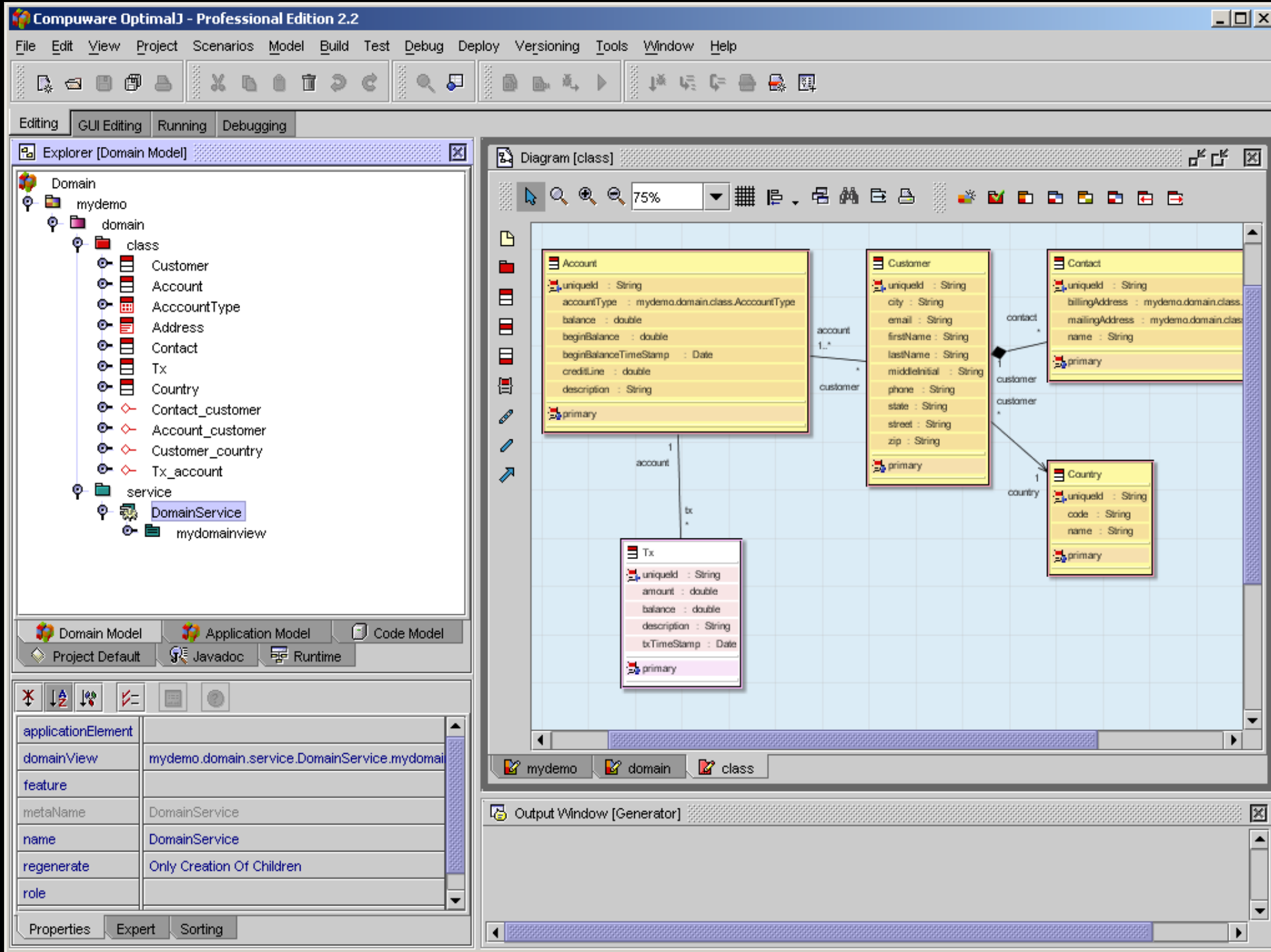
Meeting today's Business Challenges

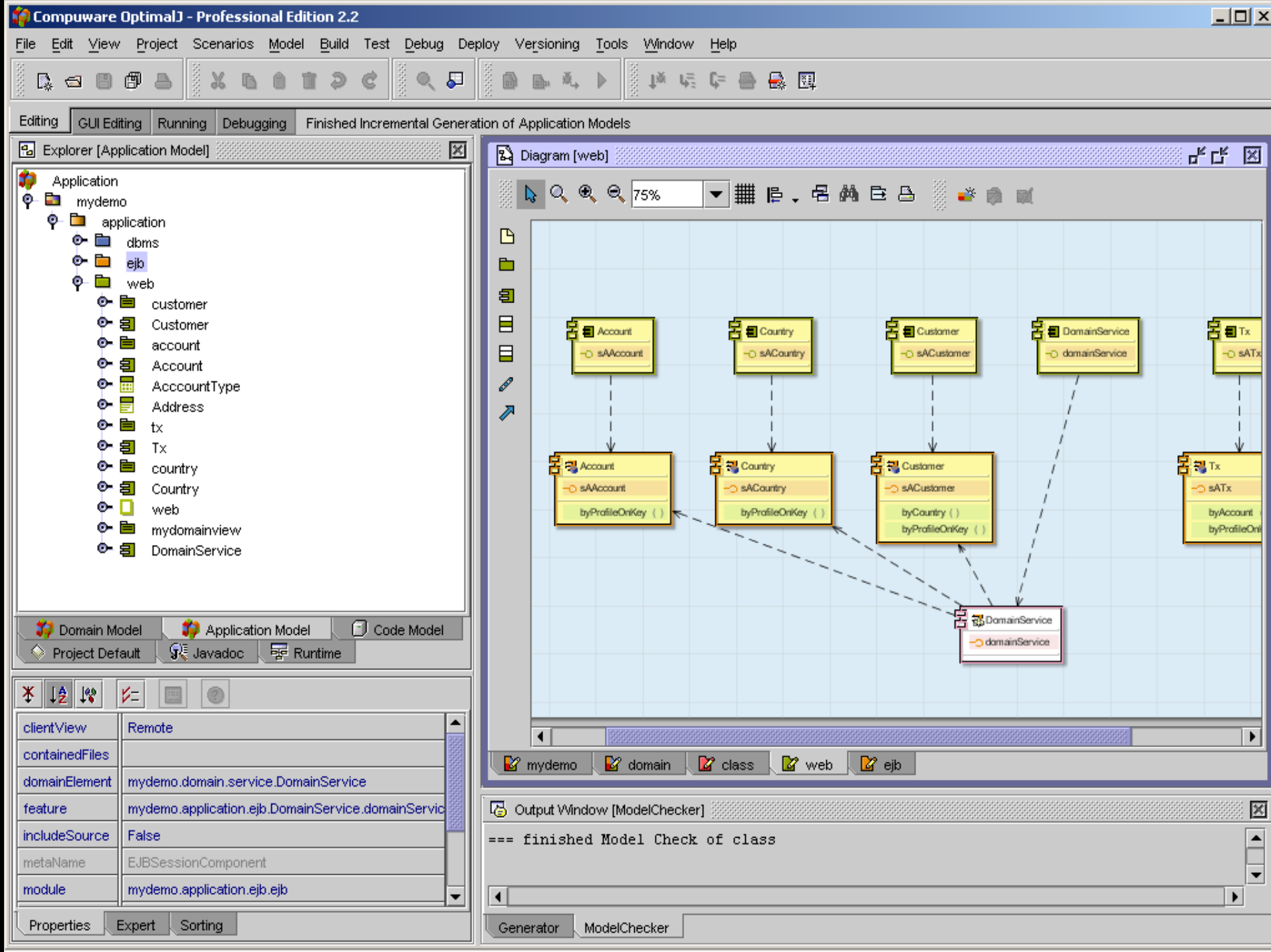
OptimalJ accelerates application delivery by simplifying Java development through the implementation of OMG's Model-Driven Architecture (MDA), enabling a team of architects, designers and developers to rapidly produce reliable J2EE business applications.

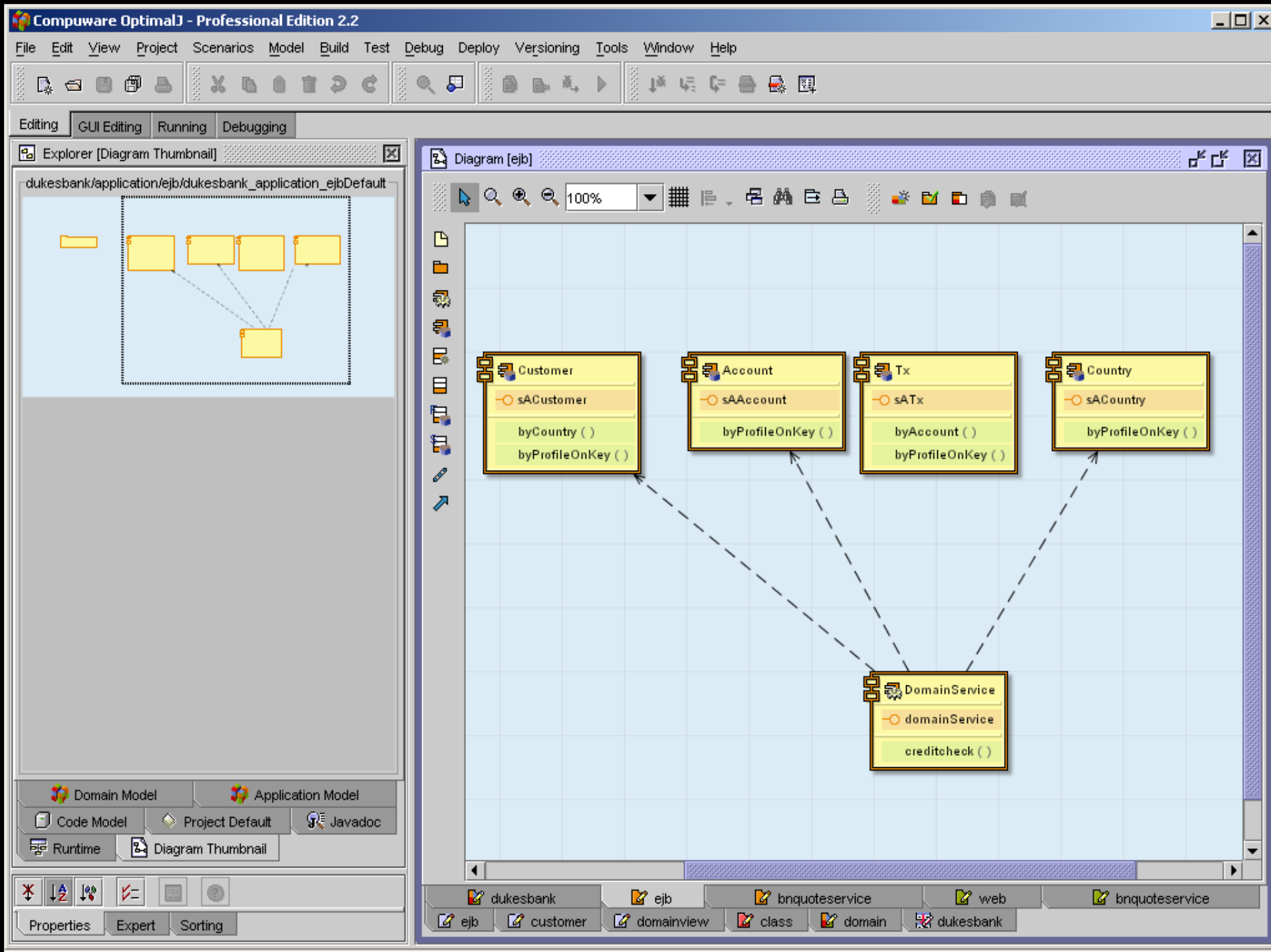
OptimalJ is a Java development environment that generates working applications directly from visual models, leveraging patterns to implement best practices for coding to J2EE the specs.

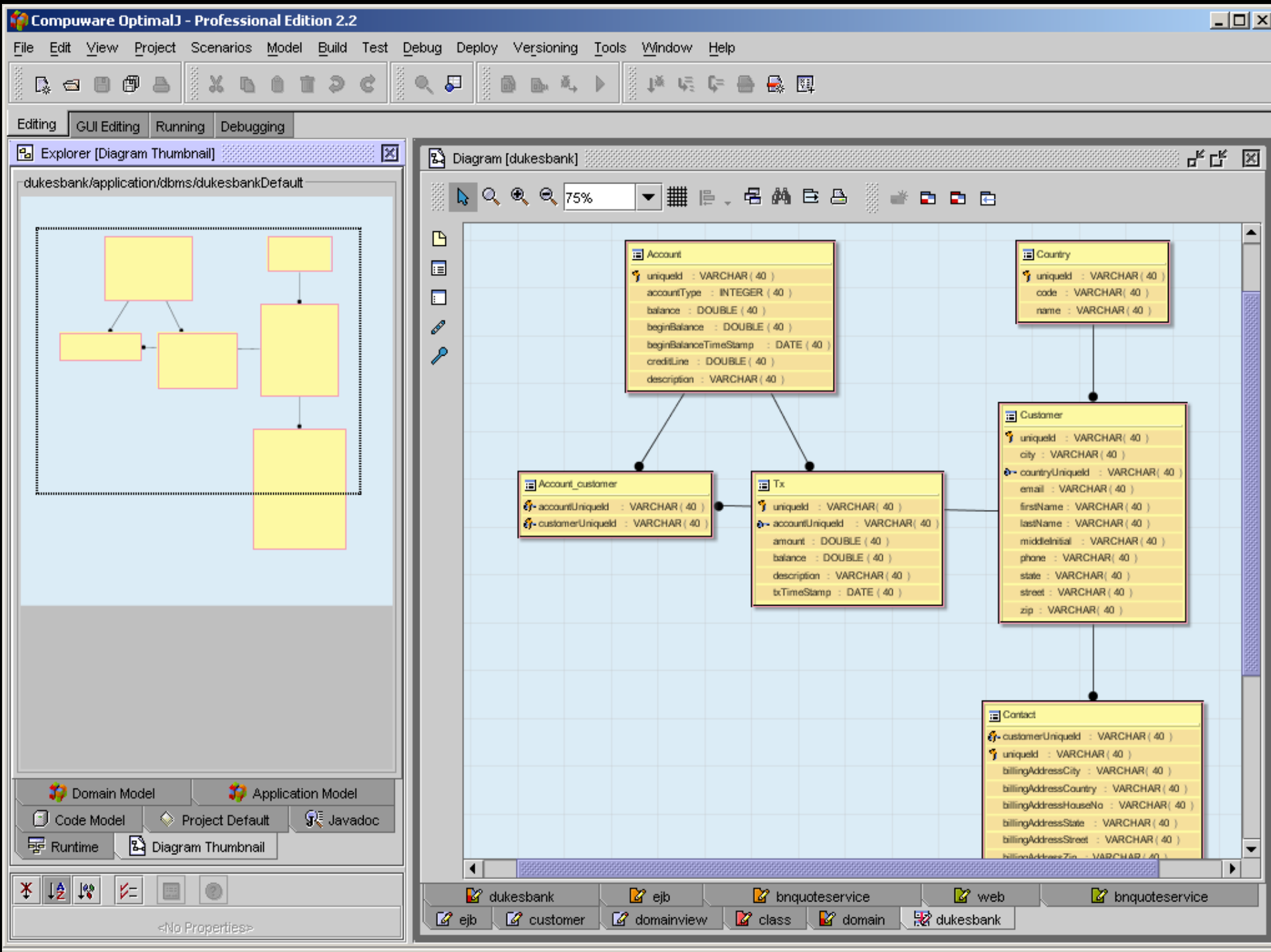
COMPUWARE®

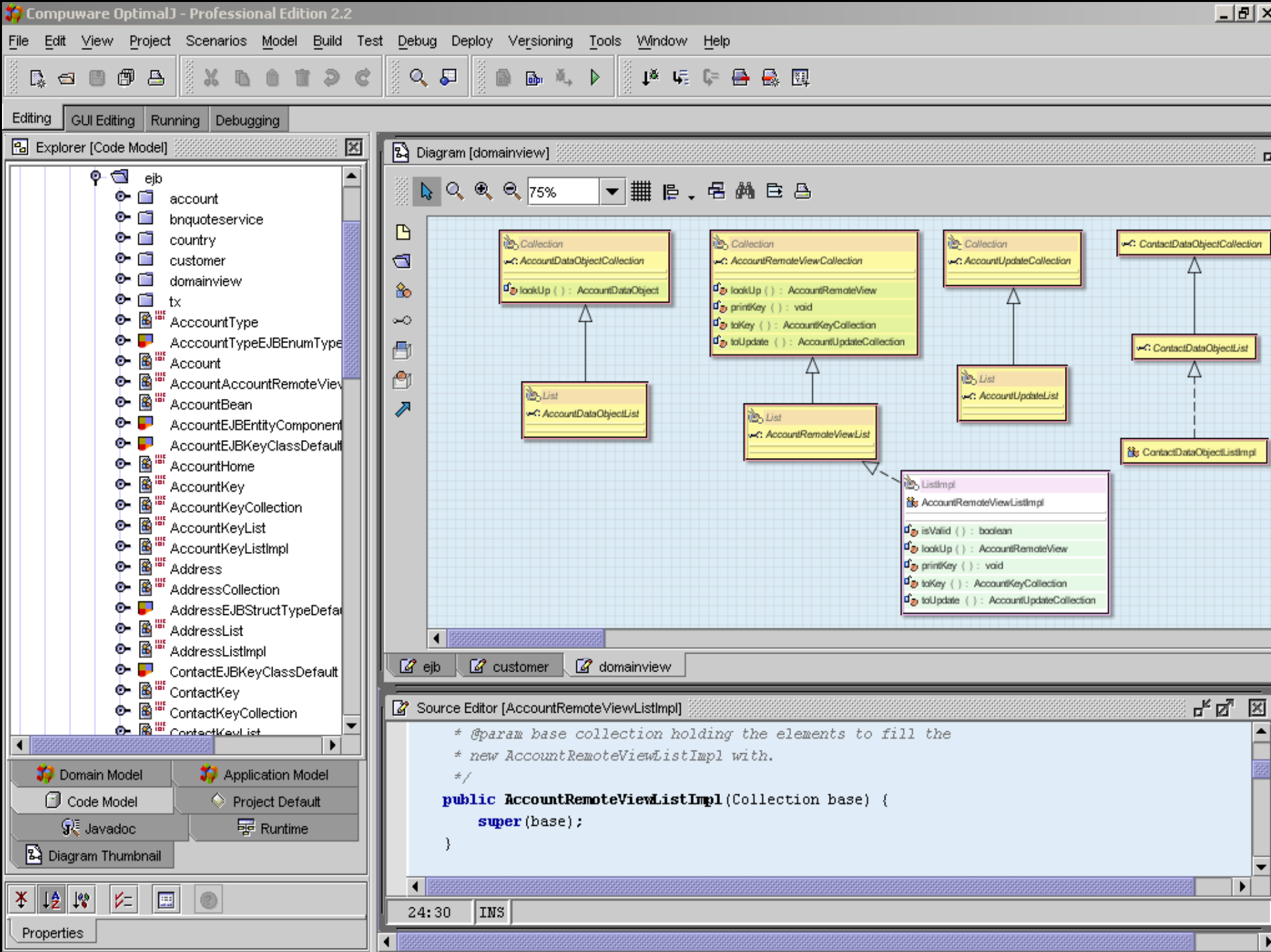
People and software for business applications℠


Compuware OptimalJ - Professional Edition 2.2
File Edit View Project Scenarios Model Build Test Debug Deploy Versioning Tools Window Help
Editing GUI Editing Running Debugging
Explorer [Domain Model]
Domain
mydemo
domain
class
Customer
Account
AcccountType
Address
Contact
Tx
Country
Contact_customer
Account_customer
Customer_country
Tx_account
service
DomainService
mydomainview
Domain Model
Application Model
Code Model
Project Default
Javadoc
Runtime
applicationElement
domainView
mydemo.domain.service.DomainService.mydomai
feature
metaName
DomainService
name
DomainService
regenerate
Only Creation Of Children
role
Properties
Expert
Sorting
Diagram [class]
75%
Account
uniqueId : String
accountType : mydemo.domain.class.AccountType
balance : double
beginBalance : double
beginBalanceTimeStamp : Date
creditLine : double
description : String
primary
Customer
uniqueId : String
city : String
email : String
firstName : String
lastName : String
middleInitial : String
phone : String
state : String
street : String
zip : String
primary
Contact
uniqueId : String
billingAddress : mydemo.domain.class.
mailingAddress : mydemo.domain.clas
name : String
primary
Country
uniqueId : String
code : String
name : String
primary
Tx
uniqueId : String
amount : double
balance : double
description : String
txTimeStamp : Date
primary
account
1..*
customer
contact
customer
customer
1
account
tx
1
country
mydemo
domain
class
Output Window [Generator]


Compuware OptimalJ - Professional Edition 2.2
File Edit View Project Scenarios Model Build Test Debug Deploy Versioning Tools Window Help
Editing GUI Editing Running Debugging Finished Incremental Generation of Application Models
Explorer [Application Model]
Application
mydemo
application
dbms
ejb
web
customer
Customer
account
Account
AcccountType
Address
tx
Tx
country
Country
web
mydomainview
DomainService
Domain Model
Application Model
Code Model
Project Default
Javadoc
Runtime
clientView Remote
containedFiles
domainElement mydemo.domain.service.DomainService
feature mydemo.application.ejb.DomainService.domainServic
includeSource False
metaName EJBSessionComponent
module mydemo.application.ejb.ejb
Properties
Expert
Sorting
Diagram [web]
75%
Account
sAAccount
Country
sACountry
Customer
sACustomer
DomainService
domainService
Tx
sATx
byProfileOnKey ( )
byCountry ( )
byAccount
mydemo
domain
class
web
ejb
Output Window [ModelChecker]
=== finished Model Check of class
Generator
ModelChecker


Compuware OptimalJ - Professional Edition 2.2
File Edit View Project Scenarios Model Build Test Debug Deploy Versioning Tools Window Help
Editing GUI Editing Running Debugging
Explorer [Diagram Thumbnail]
dukesbank/application/ejb/dukesbank_application_ejbDefault
Domain Model Application Model
Code Model Project Default Javadoc
Runtime Diagram Thumbnail
Properties Expert Sorting
Diagram [ejb]
100%
Customer
sACustomer
byCountry ( )
byProfileOnKey ( )
Account
sAAccount
byProfileOnKey ( )
Tx
sATx
byAccount ( )
byProfileOnKey ( )
Country
sACountry
byProfileOnKey ( )
DomainService
domainService
creditcheck ( )
dukesbank ejb bnquoteservice web bnquoteservice
ejb customer domainview class domain dukesbank


Compuware OptimalJ - Professional Edition 2.2
File Edit View Project Scenarios Model Build Test Debug Deploy Versioning Tools Window Help
Editing GUI Editing Running Debugging
Explorer [Diagram Thumbnail]
dukesbank/application/dbms/dukesbankDefault
Domain Model Application Model
Code Model Project Default Javadoc
Runtime Diagram Thumbnail
<No Properties>
Diagram [dukesbank]
75%
Account
uniqueId : VARCHAR (40)
accountType : INTEGER (40)
balance : DOUBLE (40)
beginBalance : DOUBLE (40)
beginBalanceTimeStamp : DATE (40)
creditLine : DOUBLE (40)
description : VARCHAR (40)
Country
uniqueId : VARCHAR (40)
code : VARCHAR (40)
name : VARCHAR (40)
Customer
uniqueId : VARCHAR (40)
city : VARCHAR (40)
countryUniqueId : VARCHAR (40)
email : VARCHAR (40)
firstName : VARCHAR (40)
lastName : VARCHAR (40)
middleInitial : VARCHAR (40)
phone : VARCHAR (40)
state : VARCHAR (40)
street : VARCHAR (40)
zip : VARCHAR (40)
Account_customer
accountUniqueId : VARCHAR (40)
customerUniqueId : VARCHAR (40)
Tx
uniqueId : VARCHAR (40)
accountUniqueId : VARCHAR (40)
amount : DOUBLE (40)
balance : DOUBLE (40)
description : VARCHAR (40)
txTimeStamp : DATE (40)
Contact
customerUniqueId : VARCHAR (40)
uniqueId : VARCHAR (40)
billingAddressCity : VARCHAR (40)
billingAddressCountry : VARCHAR (40)
billingAddressHouseNo : VARCHAR (40)
billingAddressState : VARCHAR (40)
billingAddressStreet : VARCHAR (40)
dukesbank ejb bnquoteservice web bnquoteservice
ejb customer domainview class domain dukesbank


Compuware OptimalJ - Professional Edition 2.2
File Edit View Project Scenarios Model Build Test Debug Deploy Versioning Tools Window Help
Editing GUI Editing Running Debugging
Explorer [Code Model]
ejb
account
bnquoteservice
country
customer
domainview
tx
AcccountType
AccccountTypeEJBEnumType
Account
AccountAccountRemoteVie
AccountBean
AccountEJBEntityComponent
AccountEJBKeyClassDefault
AccountHome
AccountKey
AccountKeyCollection
AccountKeyList
AccountKeyListImpl
Address
AddressCollection
AddressEJBStructTypeDefa
AddressList
AddressListImpl
ContactEJBKeyClassDefault
ContactKey
ContactKeyCollection
Domain Model
Application Model
Code Model
Project Default
Javadoc
Runtime
Diagram Thumbnail
Properties
Diagram [domainview]
75%
Collection
AccountDataObjectCollection
lookUp ( ) : AccountDataObject
List
AccountDataObjectList
Collection
AccountRemoteViewCollection
lookUp ( ) : AccountRemoteView
printKey ( ) : void
toKey ( ) : AccountKeyCollection
toUpdate ( ) : AccountUpdateCollection
List
AccountRemoteViewList
Collection
AccountUpdateCollection
List
AccountUpdateList
ContactDataObjectCollection
ContactDataObjectList
ContactDataObjectListImpl
ListImpl
AccountRemoteViewListImpl
isValid ( ) : boolean
lookUp ( ) : AccountRemoteView
printKey ( ) : void
toKey ( ) : AccountKeyCollection
toUpdate ( ) : AccountUpdateCollection
ejb
customer
domainview
Source Editor [AccountRemoteViewListImpl]
* @param base collection holding the elements to fill the
* new AccountRemoteViewListImpl with.
*/
public AccountRemoteViewListImpl(Collection base) {
super(base);
}
24:30
INS