

Successfully Integrating Best of Breed Development Tools

Dave Tomkins
Jumar Solutions
Session Number 33
Tuesday, 13 June

Session Abstract

- The presentation will discuss the reasons for integrating different model-based development tools from the same or different vendors into the same development lifecycle and some of the practicalities of automatically exchanging model objects between tool repositories in a managed and seamless way.
- Adopting other proprietary UML analysis tools need not be a challenge when customers also wish to use them together with AllFusion Gen's enterprise code generation and delivery capabilities. The presentation will focus on data modeling and UML modeling in an AllFusion Gen-centered development process.
- The need for effective lifecycle management, and demands for better quality and productivity, make automated integration of 'best of breed' tools a necessity.

Agenda

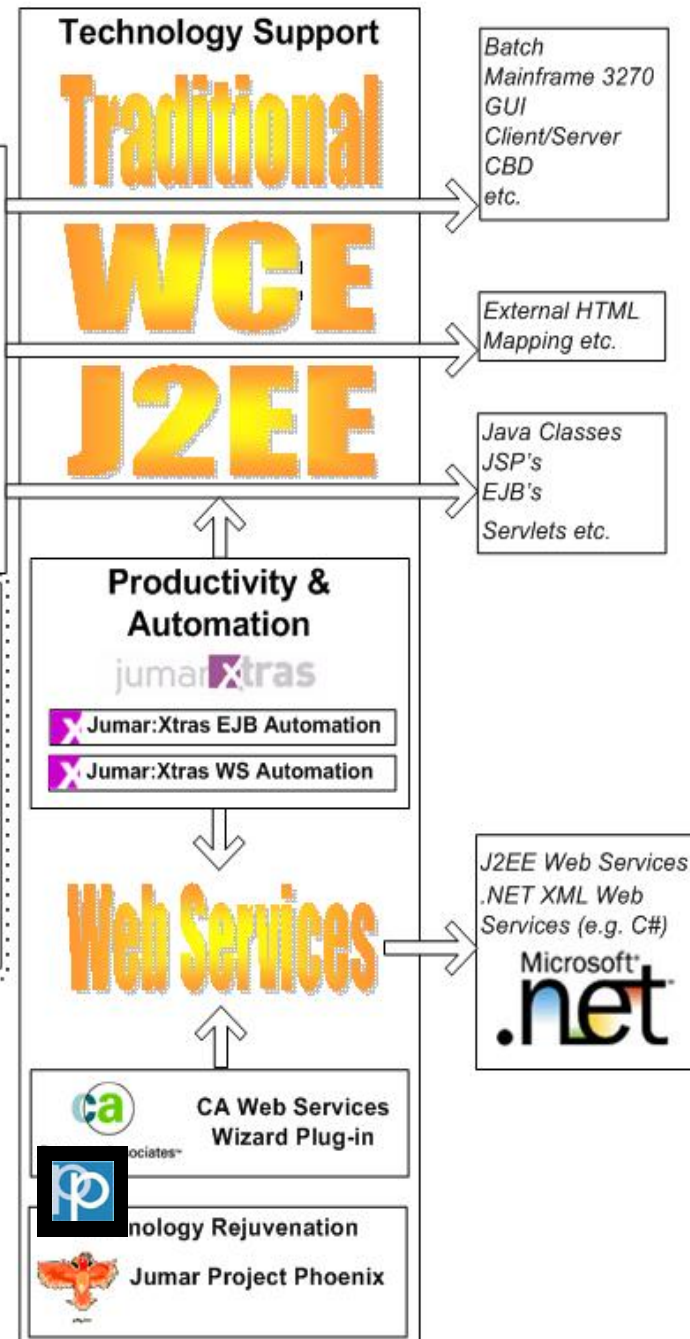
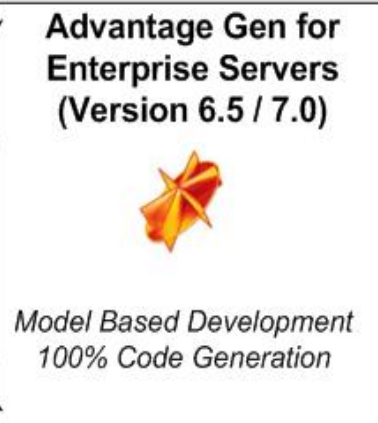
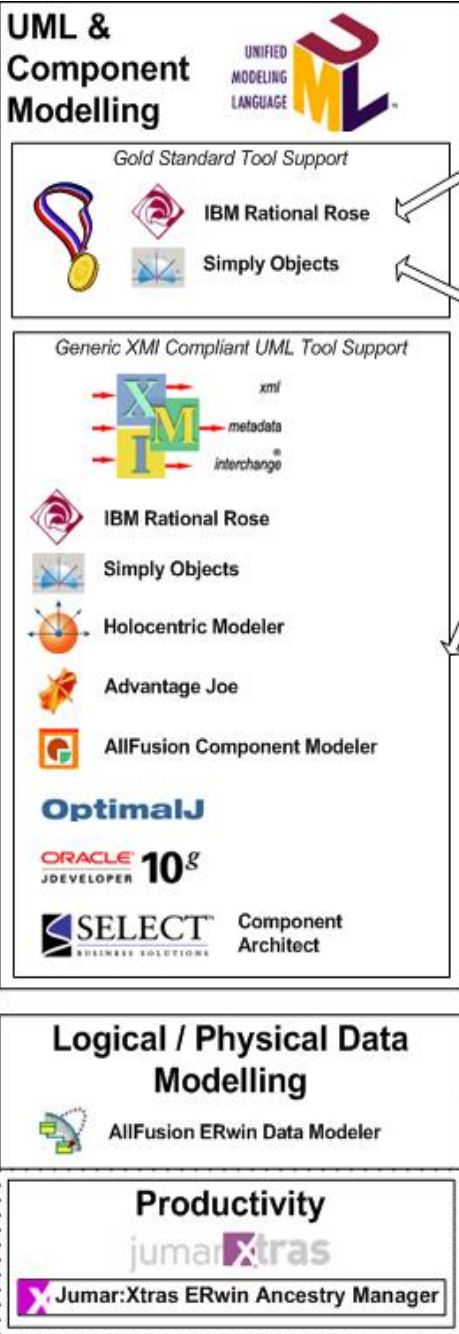
- Why Integrate Tools?
- The Big Picture
- Object Mapping and Model Transformation
- Data Modeling
- UML Modeling
- Demonstration of XMI Interchange with AllFusion Gen
- The Need for Standards
- Managing Change, and Practical Considerations

Why Integrate Tools?

- Development lifecycles are complex. Different stages and tasks have different requirements
- Different types of user have different requirements
 - Front end analysts demand more visualisation
 - Coders tend not to use a lot of diagrams
- Objects created during one development stage are used in another
- No single tool can be the best at everything

The Big Picture

- A development architecture diagram which shows how various development tools and technologies relate to one another, with AllFusion Gen at the core
- Shows where Jumar's technologies help by enabling integration and easing transition from one phase or technology to another through automation



UML & Component Modelling

UNIFIED MODELING LANGUAGE

Gold Standard Tool Support

IBM Rational Rose

Simply Objects

Generic XML Compliant UML Tool Support

xml

metadata

interchange

IBM Rational Rose

Simply Objects

Holocentric Modeler

Advantage Joe

AllFusion Component Modeler

OptimalJ

ORACLE 10g JDEVELOPER

Microsoft SQL Server Business Solutions Architect

Logical / Physical Data Modelling

AllFusion ERwin Data Modeler

Productivity

jumarXtras

Jumar:Xtras ERwin Ancestry Manager

jumar links

Jumar:Links for Rational Rose and Advantage Gen

jumar links

Jumar:Links for Simply Objects and Advantage Gen

jumar links

Jumar:Links for Advantage Gen and UML

jumar links

Jumar:Links for AllFusion ERwin Data Modeler and Advantage Gen

Advantage Gen for Enterprise Servers (Version 6.5 / 7.0)

Model Based Development
100% Code Generation

Productivity & Automation

Jumar:Xtras Print Enhancer

Jumar:Xtras PlugIn Manager

Technology Support

Traditional

Batch
Mainframe 3270
GUI
Client/Server
CBD
etc.

WCE

External HTML Mapping etc.

J2EE

Java Classes
JSP's
EJB's
Servlets etc.

Web Services

J2EE Web Services
.NET XML Web Services (e.g. C#)
Microsoft .net

Productivity & Automation

CA Web Services Wizard Plug-in

Technology Rejuvenation

Jumar Project Phoenix

UML & Component Modelling



Gold Standard Tool Support



IBM Rational Rose



Simply Objects

Generic XMI Compliant UML Tool Support



IBM Rational Rose



Simply Objects



Holocentric Modeler



Advantage Joe



AllFusion Component Modeler

OptimalJ

ORACLE 10^g
JDEVELOPER



Component Architect

jumar
links

Jumar:Links for Rational Rose and Advantage Gen

jumar
links

Jumar:Links for Simply Objects and Advantage Gen

jumar
links

Gen and UML

Advantage Gen for Enterprise Servers (Version 6.5 / 7.0)



Model Based Development
100% Code Generation

Productivity & Automation

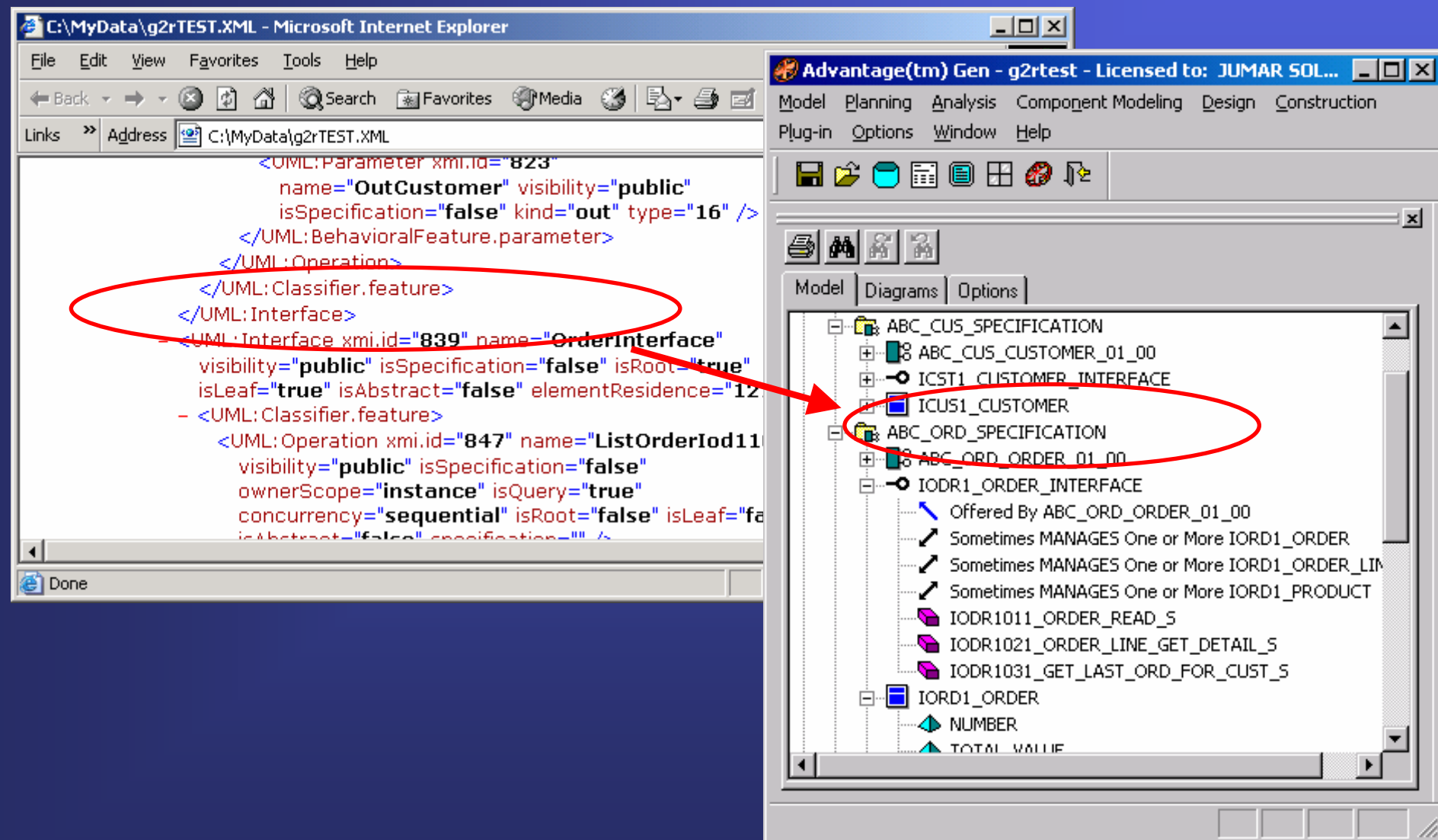


Xtras Model Reporter

- X Jumar:Xtras Bulk Update
- X Jumar:Xtras Difference Analysis
- X Jumar:Xtras Print Enhancer
- X Jumar:Xtras PlugIn Manager

UML modelling and AllFusion Gen

Connecting Tools - Object Mapping



The image displays two windows side-by-side, illustrating the process of object mapping between a UML XML file and a software model.

Left Window: Microsoft Internet Explorer
 The address bar shows the file path: `C:\MyData\g2rTEST.XML`. The main content area displays XML code for UML elements. A red circle highlights the `<UML:Interface xmi.id="839" name="OrderInterface"` element. A red arrow points from this element to the corresponding entry in the right window.

```

<UML:Parameter xmi.id="823"
  name="OutCustomer" visibility="public"
  isSpecification="false" kind="out" type="16" />
</UML:BehavioralFeature.parameter>
</UML:Operation>
</UML:Classifier.feature>
</UML:Interface>
<UML:Interface xmi.id="839" name="OrderInterface"
  visibility="public" isSpecification="false" isRoot="true"
  isLeaf="true" isAbstract="false" elementResidence="12" />
</UML:Classifier.feature>
<UML:Operation xmi.id="847" name="ListOrderIod11"
  visibility="public" isSpecification="false"
  ownerScope="instance" isQuery="true"
  concurrency="sequential" isRoot="false" isLeaf="fa"
  isAbstract="false" specification="..." />
  
```

Right Window: Advantage(tm) Gen - g2rtest - Licensed to: JUMAR SOL...
 The window shows a hierarchical tree view of the model. A red circle highlights the `ABC_ORD_SPECIFICATION` entry, which is the model's representation of the `OrderInterface` from the XML file. Below it, several other elements are listed, including `ABC_CUS_SPECIFICATION`, `ICST1_CUSTOMER_INTERFACE`, `ICUS1_CUSTOMER`, `ABO_ORD_ORDER_01_00`, `IODR1_ORDER_INTERFACE`, and `IODR1_ORDER`.

Transformer Requirements

- Must deal with all major objects on both sides
- Must deal with the issues of change and configuration management
- Must be flexible to support in-house standards
- Must show quality and productivity gain in the development cycle
- Must be intelligent - more than a simple metadata bridge
- Must be simple to use on a daily basis

FRAMEWORK

Model Transformer

FRAMEWORK

API or
interface

Modelling
Tool

ActiveX
layer

Gen API

Advantage
Gen

**BUSINESS OR
MAPPING RULES**

Jumar:Links Features

- Simple GUI interface, easy to use
- Bi-directional
- Direct AllFusion Gen model manipulation
- Common transformation engine
- Slot-in business/mapping rules module
- Flexible options
- Supports change management

Data Modelling

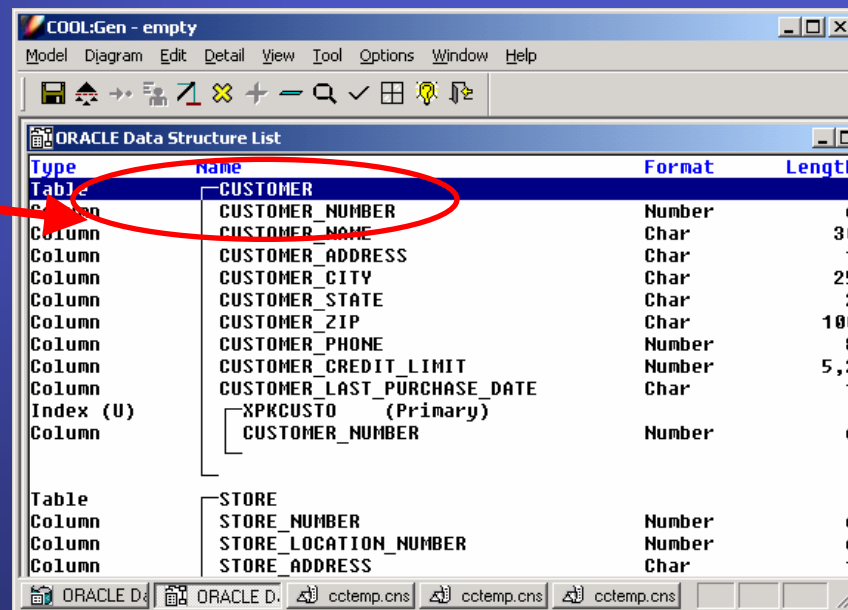
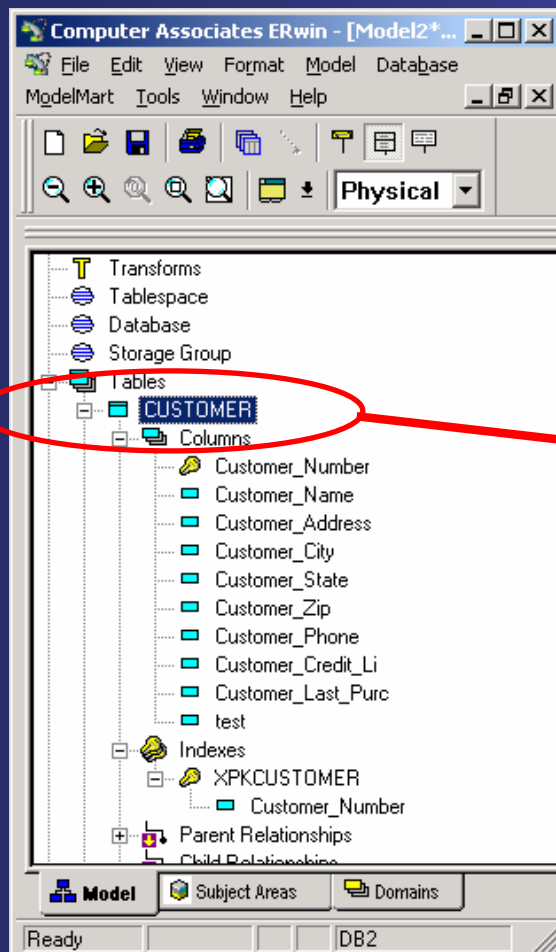
- Data Modelling is supported in a minimalistic way by Gen
- Dedicated data modelling tools such as CA's AllFusion ERwin Data Modeler are full-featured data and database modeling tools with many more features
- Combination of Gen and ERwin allows rapid development of new web/J2EE applications based upon existing data stores
- Improves user and analyst communication
- Allows the DBA function and the Gen development world to be properly integrated

ERwin Features

- Design layer architecture
- Full physical property support
- Datatype mapping facility
- Naming Standards and Glossary
- Complete Compare bi-directional synchronization across models
- Database design generation
- Forward and reverse engineering
- Large model management via Subject Areas and Stored Displays
- Drawing objects
- Data Warehouse design

Object Mapping Rules

How to map ERwin
model objects to Gen?



Mapping Rules - Examples

An ERwin Entity becomes a Gen Entity. Its attributes and relationships are also transformed

The ERwin attribute properties are mapped to the Gen attribute. Datatype mapping is user-customisable

Validation Rules in ERwin become Permitted Values or Ranges in Gen

ERwin Subtype relationships are converted to Gen Entity/Subtype hierarchies

Classifying attributes, values and partitionings are automatically created in Gen where needed for subtypes (optional)

UML Modelling

- Promotes the robust and scalable architectures critical for enterprise applications
- Component design for reuse
- Use Case technique popular and powerful
- Visualisation
- Maintenance and documentation
- Openness
- Future proofing - part of OMG's MDA* strategy
- Resource availability

Different Paradigms?

UML

- Class
- Attribute
- Association
- Component
- Interface
- Operation
- Parameter

AllFusion Gen

- Entity Type / Spec Type
- Attribute
- Relationship
- Component Spec Type
- Interface Type
- Action Diagram
- Information View

Some things map relatively easily...

Different Paradigms?

UML

- Generalisation and Specialisation
- Package
- Dependency
- Datatype
- Stereotype
- TaggedValue
- ...

AllFusion Gen

- Subtype
- Subject Area
- Attribute Properties
- Identifier
- Info View Properties
- Work Set
- ...

Some not so easily...

UML Tools

Dozens to choose from



Rational Rose



Component Architect



AllFusion
Component
Modeler



Popkin
System Architect



Borland Together



Oracle JDeveloper



Simply Objects

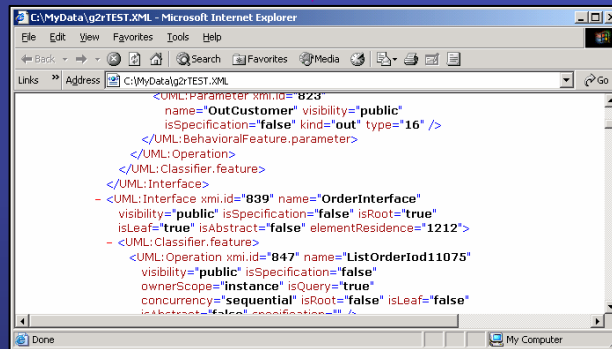
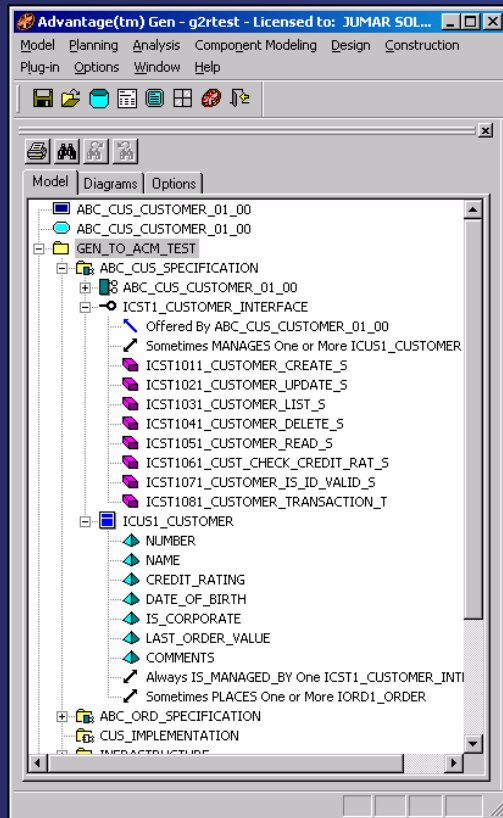


Compuware
OptimalJ

Model Transformation

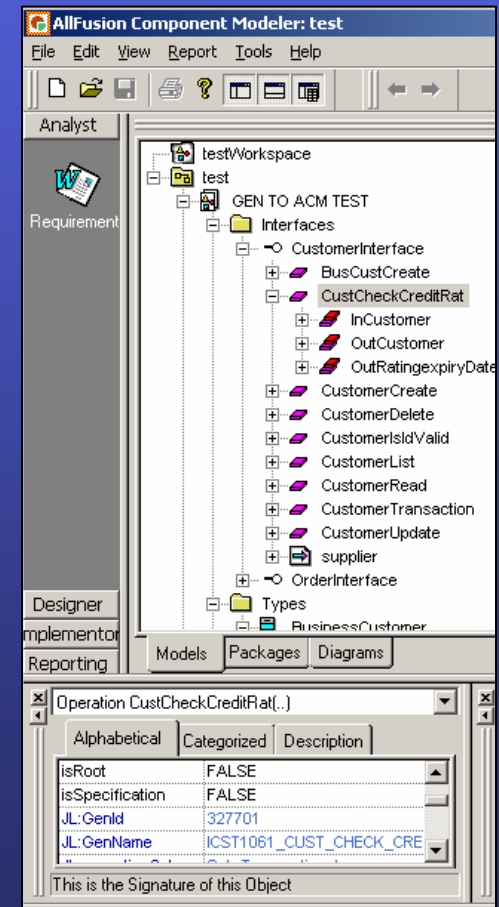
Creating a starter UML Model from AllFusion Gen

AllFusion Gen to
XML file



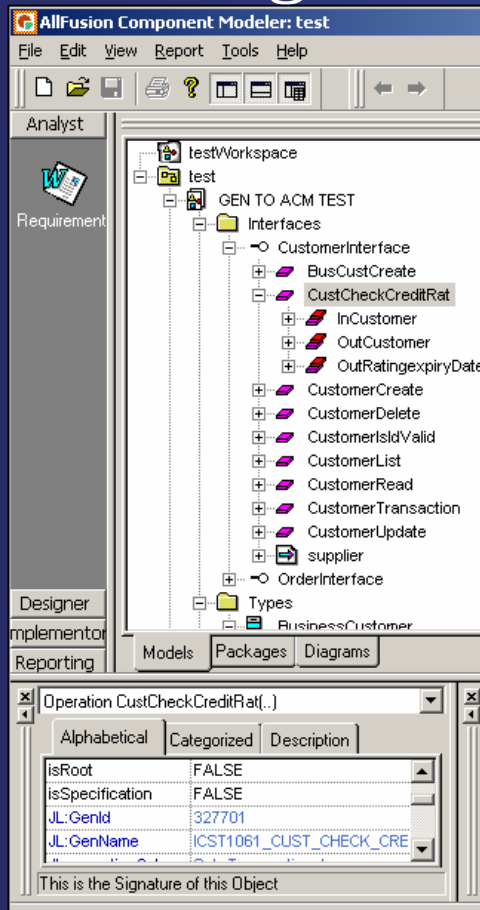
XML import

XML file
to UML Tool



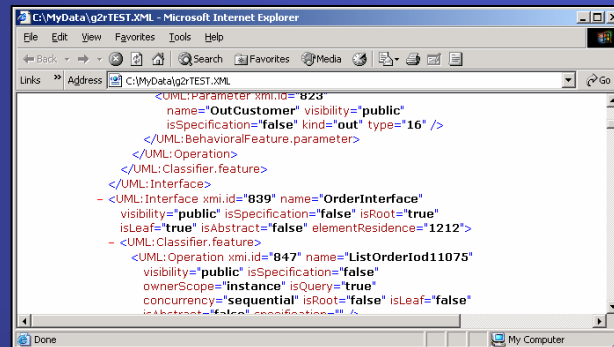
Model Transformation

Creating an AllFusion Gen model from UML Tool

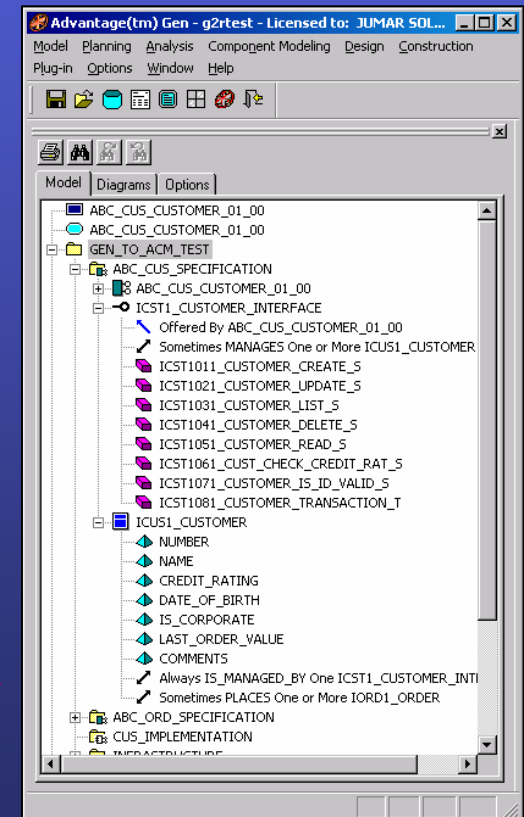


UML Tool to
XML file

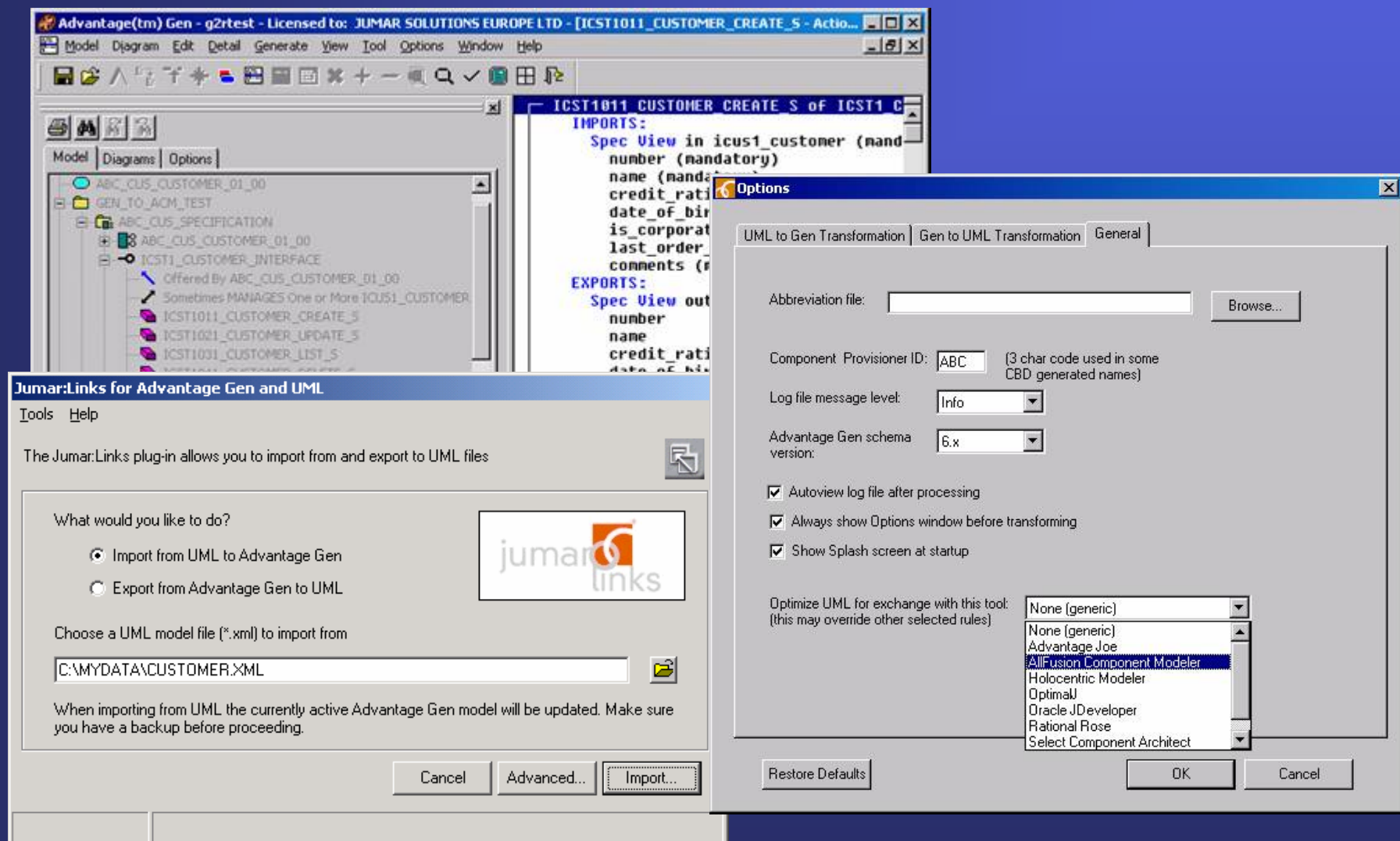
XMI export



XML file
to AllFusion Gen



Demonstration AllFusion Gen and UML

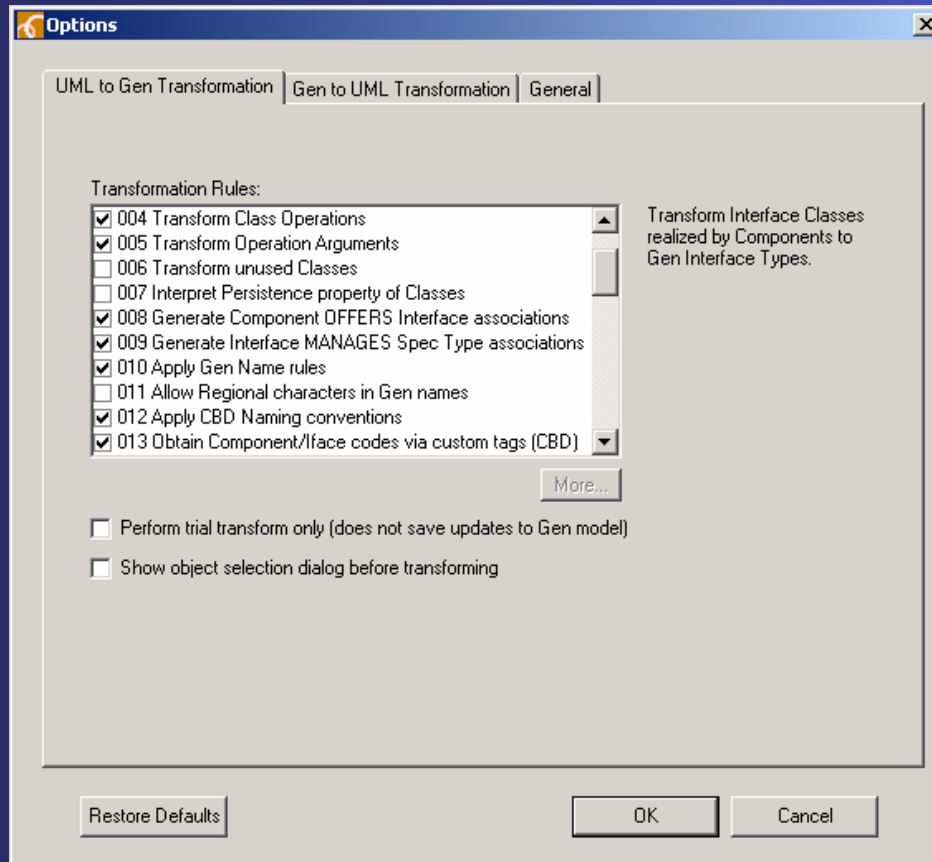


The screenshot displays the Advantage Gen software interface. The main window shows a model tree on the left with elements like 'ABC_CUS_CUSTOMER_01_00', 'GEN_TO_ACM_TEST', and 'ICST1011_CUSTOMER_CREATE_S'. The central pane shows the 'ICST1011_CUSTOMER_CREATE_S' model with its 'IMPORTS' and 'EXPORTS' sections.

Two dialog boxes are overlaid on the main window:

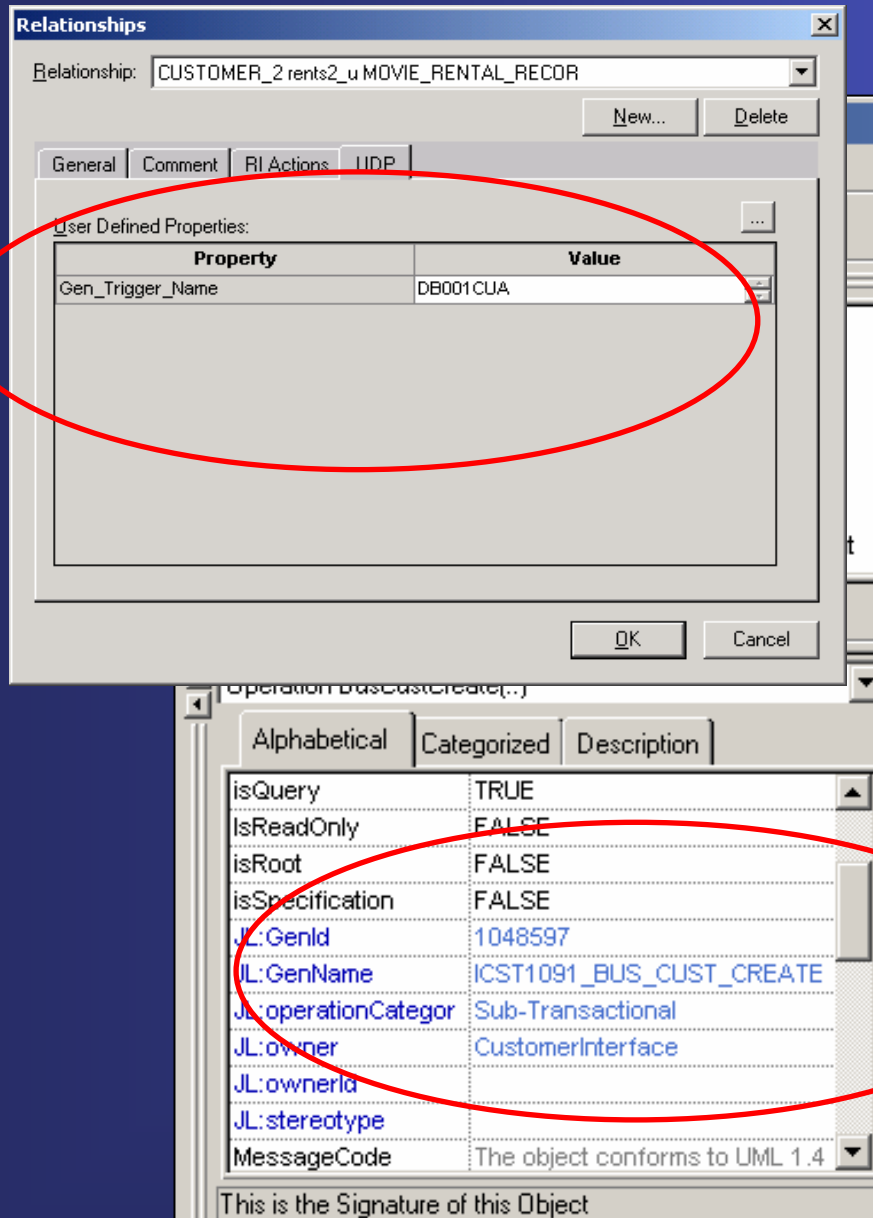
- Jumar:Links for Advantage Gen and UML**: This dialog box has a title bar with 'Tools' and 'Help'. It contains a section 'What would you like to do?' with two radio buttons: 'Import from UML to Advantage Gen' (selected) and 'Export from Advantage Gen to UML'. Below this is a text field for 'Choose a UML model file (*.xml) to import from' with the value 'C:\MYDATA\CUSTOMER.XML'. A note at the bottom states: 'When importing from UML the currently active Advantage Gen model will be updated. Make sure you have a backup before proceeding.' Buttons at the bottom include 'Cancel', 'Advanced...', and 'Import...'.
- Options**: This dialog box has tabs for 'UML to Gen Transformation', 'Gen to UML Transformation', and 'General'. The 'General' tab is active. It contains fields for 'Abbreviation file:' (with a 'Browse...' button), 'Component Provisioner ID:' (set to 'ABC'), 'Log file message level:' (set to 'Info'), and 'Advantage Gen schema version:' (set to '6.x'). There are three checked checkboxes: 'Autoview log file after processing', 'Always show Options window before transforming', and 'Show Splash screen at startup'. A dropdown menu for 'Optimize UML for exchange with this tool:' is open, showing options like 'None (generic)', 'Advantage Joe', 'AllFusion Component Modeler' (highlighted), 'Holocentric Modeler', 'OptimalJ', 'Oracle JDeveloper', 'Rational Rose', and 'Select Component Architect'. Buttons at the bottom include 'Restore Defaults', 'OK', and 'Cancel'.

Flexibility



Many click-to-select rules allow fine tuning for the result you desire.

Some rules, such as ***Apply CBD naming standards***, can save a massive amount of manual effort.



ERwin: Custom Properties

UML: TaggedValues

used to store object ids and names

Required to support round-trip transformations

Practical Considerations

- Development life cycle and model management strategies must be extended
- Roles and responsibilities must be defined – central vs distributed
- Model transformations must be controlled
- Decide which is the master model. Try to transform in one direction only
- Standards at both ends are key to maximum efficiency of the transformation process
- Some compromises may be required

Summary

- There are **significant benefits** to the IT department from taking advantage of best of breed tools
- Well-defined **object mapping rules** are the key to successful tool integration
- Integration can be direct via **APIs** or via **common interface** exchange mechanisms such as XML/XMI
- It **is** possible and practical to model data with ERwin, model components with UML and realise applications with AllFusion Gen **within the same development life cycle**
- More than just a bridge is required. **Intelligent, configurable transformation** checks, changes, adds
- High **automation** means high quality and productivity, in-line QA and standards enforcement
- Maximum benefits come from a **managed implementation** into an amended development lifecycle

Questions

