



29 Oct. – 31 Oct. 2007

Plano, TX

ModelCVS – CA Gen goes Eclipse



Mustafa ARIKAN, Xiaoxia LIN

ARIKAN Productivity Group **APG**

Biography

- **Mustafa ARIKAN**
ARIKAN Productivity Group
mustafa.arikan@arikan.at
- Senior Software Engineer
- Prior to APG he served as software consultant and industrial engineer (operations research) for various industrial organizations
- More than 25 years of industrial experience and leadership in many innovative technical and commercial IT projects and many awards throughout his IT career so far.
- Mustafa Arikan has a BSc degree in industrial engineering from the Bosphorus University in Istanbul and some postgraduate education in IT and Mathematics in Vienna.

Biography

- Xiaoxia LIN
ARIKAN Productivity Group
xiaoxia.lin@arikan.at
- Senior Software Engineer
- Xiaoxia LIN is currently working at her doctoral work in Model Driven Development at the Vienna University of Technology
- She is author of many mission critical software systems in Europe
- She has more than 10 years of IT experience in various industrial organizations
- Xiaoxia Lin has a MSc degree in informatics from the Vienna University of Technology and a BSc degree in Medicine from China

Content

- Tool Integration
- Model & Metamodel, Model Transformation
- Eclipse Modeling Framework
- Metamodelling & Metamodelling Toolkit
- Import/Export Facilities: Tool Adapter
- Model Synchronization

- Further Work

Tool Integration

Tools are called integrated if they function coherently and effectively in an environment as a whole.

As modeling tools supporting different modeling languages and domains are not interoperable per se, they do not provide flexibility in terms of an open tool chain, and therefore impair the use of multiple tools to cooperatively model a system.

Models & Metamodels

- A model is an abstraction of a phenomena in the real world, and a metamodel is yet another abstraction, highlighting properties of the model itself. A model is said to conform to its metamodel like a program conforms to the grammar of the programming language in which it is written.
- Metamodeling or meta-modeling is the analysis, construction and development of the frames, rules, constraints, models and theories applicable and useful for the modeling in a predefined class of problems. This concept definition is composed with the notions of the terms meta- and modeling.

Model Transformation

- Can be defined as an operation taking one or more models as input and creating another model(s) as output (batch)
- May support change propagation (incremental)
- May support inverse transformation (bi/multi-directional)
- May cover only parts of the input/output models (partial)
- May operate on overlapping input/output (in-place)

Model based Tool Integration

- Model defines entities and relationships of a system
- Metamodel defines entities and relationship of a model
- Transformation Rules are derived from Tools Metamodels
- Transformation occurs between Metamodel Instances
- Metamodels must belong to the same Metametamodel

Tool Integration

- Tools Metamodels
- Programmatic Access to the model objects
- Higher Abstraction through Model Access Services
- Transformation among Tools Model Instances
- Code Generation using various Models
- Runtime Glue of Generated Code

Eclipse Modeling Framework

- EMF is a modeling framework and code generation facility for building tools and other applications based on a structured data model.
- Described in XMI, EMF provides tools and runtime support to produce a set of Java classes for a model, a set of adapter classes that enable viewing and command-based editing of this model, and a basic editor.
- Models can be specified using annotated Java, XML documents, or modeling tools like Rational Rose, then imported into EMF.
- Most important of all, EMF provides the foundation for interoperability with other EMF-based tools and applications.

<http://www.eclipse.org/modeling/emf/?project=emf>

ModelCVS's CA Gen Specific Work

- Metamodeling Toolkit
 - Metamodel Generation
 - Metamodel Editor
 - Creation of Sub-Metamodel(s)
 - ...
- Gen Adapter
 - Gen Adapter Framework + Metamodel Elements Decorators
 - A kind of object-oriented API for CA Gen's Encyclopedia
 - For any kind of Encyclopedia-access by using EMF

CA Gen Adapter & Metamodeling Toolkit

- Constructs Tools ecore conform Metamodel
- Supports creation of different packages
- Preserves Object consistency
- Generates Tool Adapter

CA Gen Adapter Classes

- JAVA Classes in Eclipse
- Wraps the CA Gen API's
- The Model Objects can be read and added
- CA Gen models can be created programmatically
- CA Gen models can be reconstructed
- Important for Legacy Renewal and Model Modernization

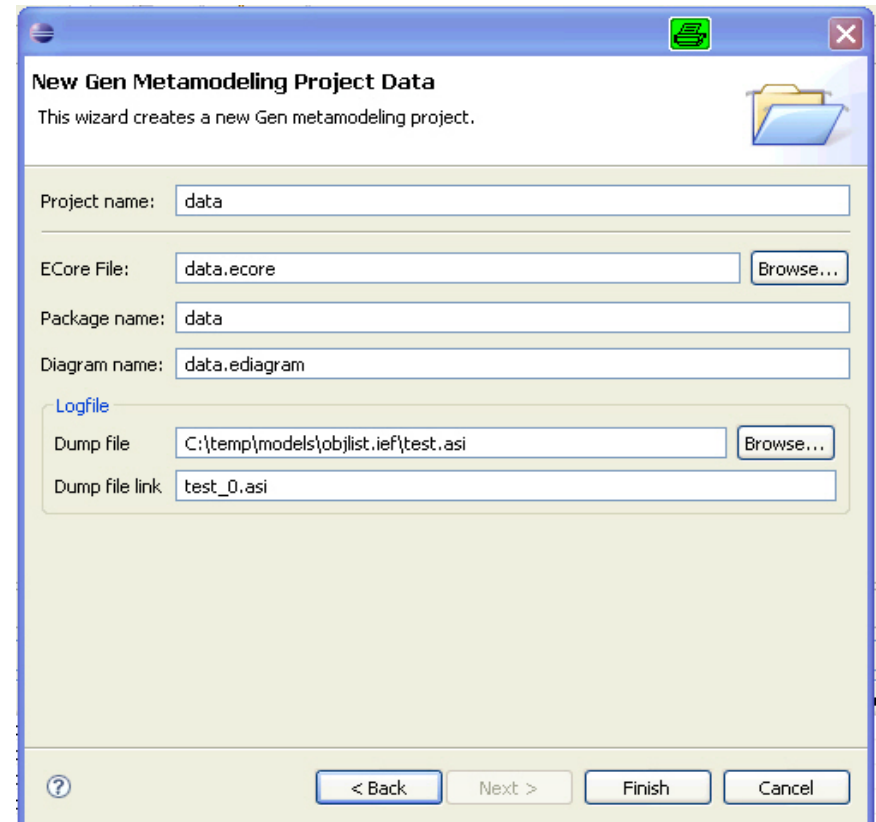
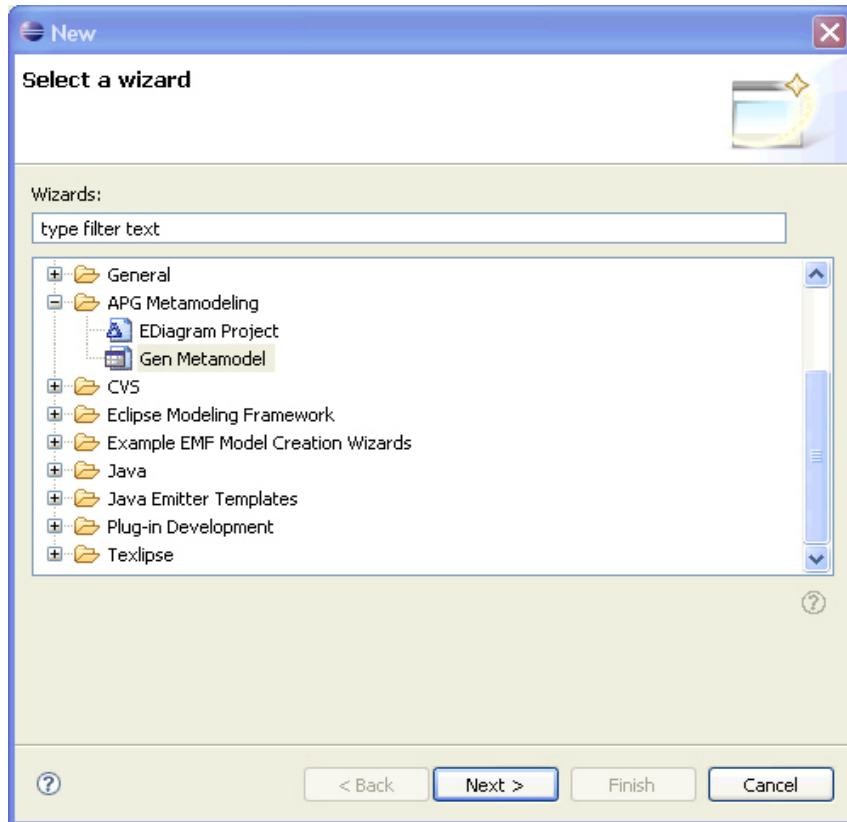
Add Gen Object by using Adapter

```
public void addEntityTypeByUsingAdppter(){  
    //create Entity  
    HighestLvlAnalysisEntityType entType =  
        DataFactory.eINSTANCE.createHighestLvlAnalysisEntityType();  
    entType.setName("Entity_A");  
    //create Attribute  
    UserDefinedAttribute attr1 =  
        DataFactory.eINSTANCE.createUserDefinedAttribute();  
    attr1.setName("Attr_1");  
    attr1.setDomain(DomainType.TEXT_LITERAL);  
    //make relationship  
    entType.getDescribedByATTR().add(attr1);  
    //add to model  
    adapter.addObject(entType);  
}
```

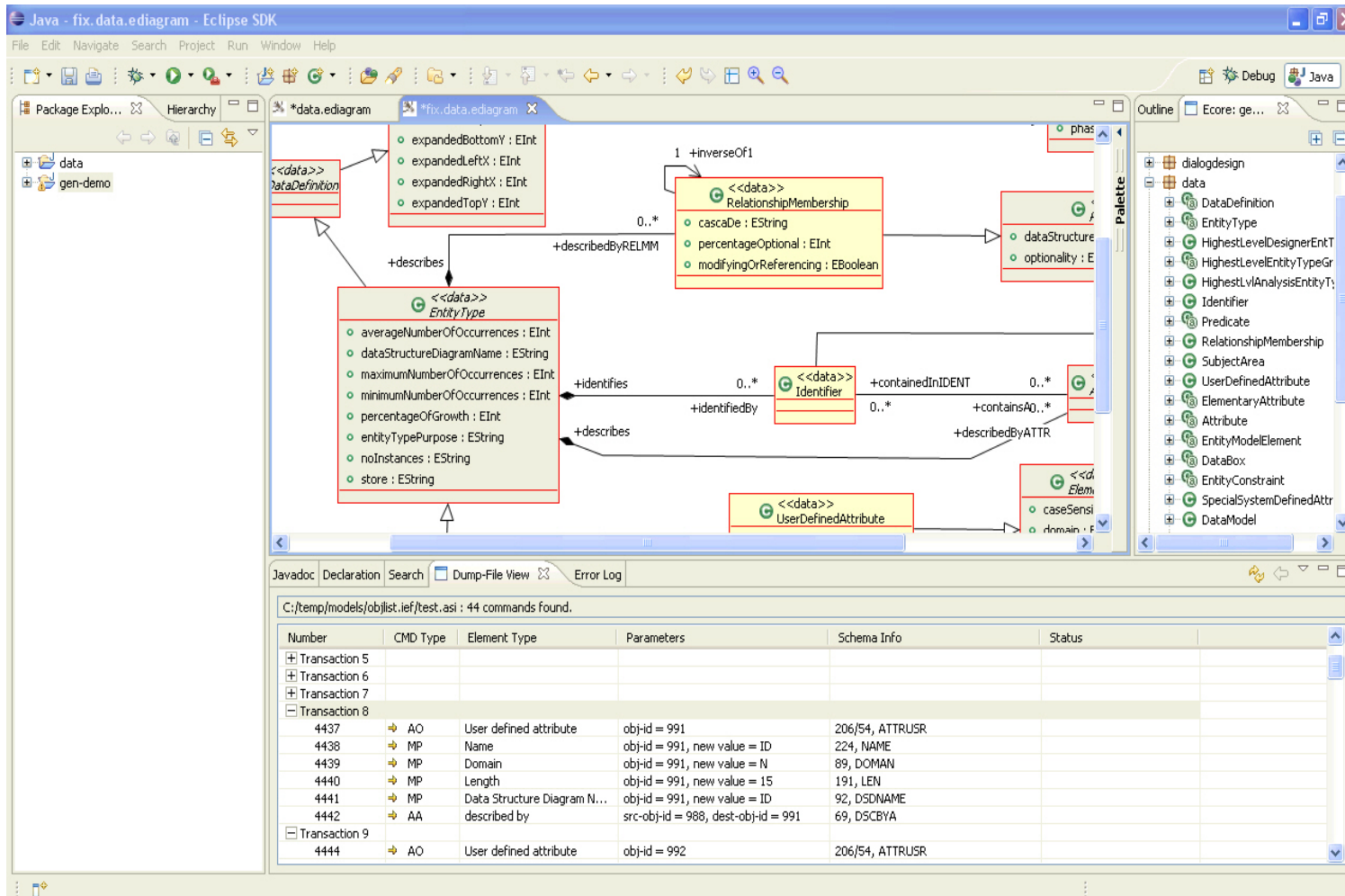
Add Gen Object by „Brute force API“

```
public void addEntityTypeByUsingAPI()throws Exception{  
    //create entity type  
    long entId = (Long)api.addObject(0, 113).getReturnObj();  
    api.updateStringPrp(entId, 224, "Entity_A");  
  
    //create attribute  
    long attrId = (Long)api.addObject(0, 54).getReturnObj();  
    api.updateStringPrp(attrId, 224, "Attr_1");  
    api.updateStringPrp(attrId, 89, "T");  
  
    //make relationship  
    api.addAssociation(entId, 69, attrId);  
}
```

Creating Metamodel for CA Gen's Data Model(1/2)



Creting Metamodel for CA Gen's Data Model(2/2)



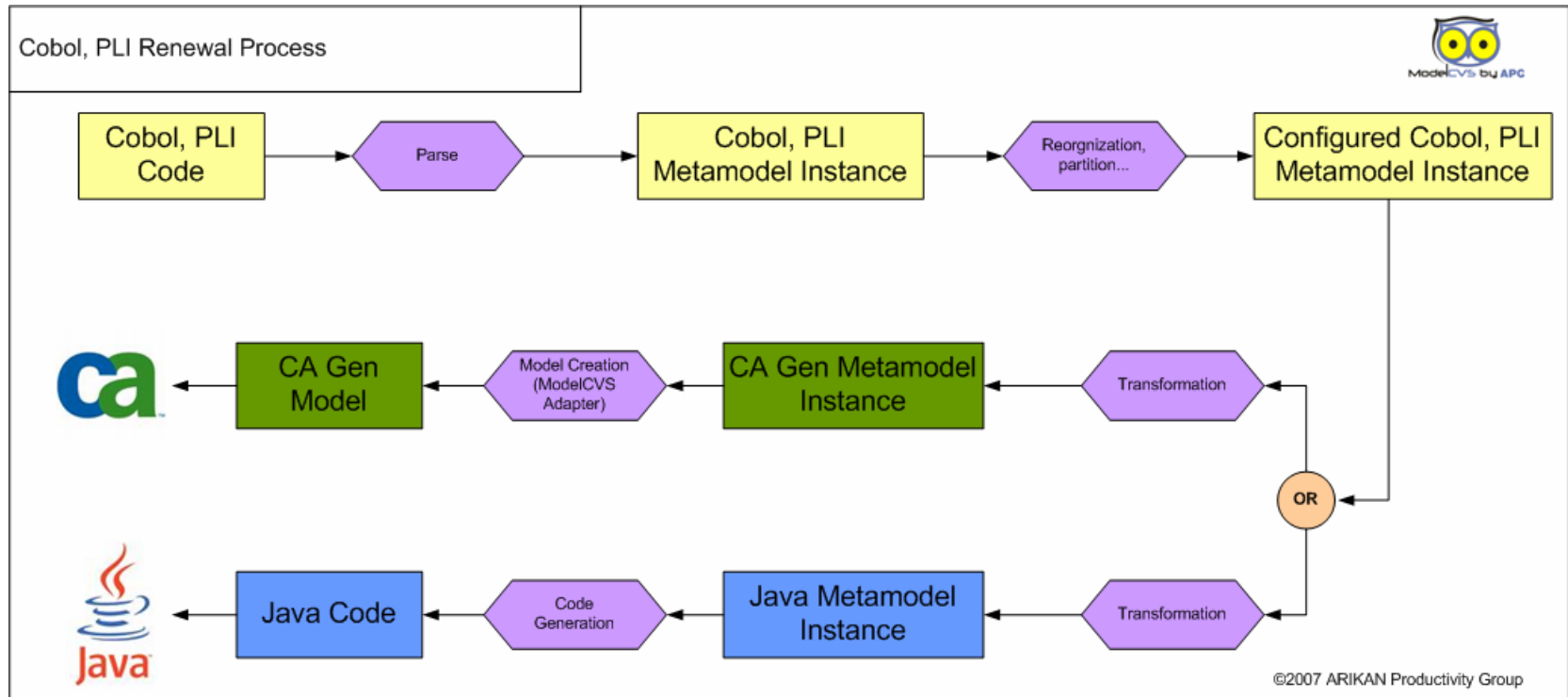
Test Support

- Definition & Recording of Testcases
 - Definition of Data Snapshots
- Code Change Analysis, Impact Analysis
- GenUnit Tests, Testdriver Generation
- Automated Integration Tests

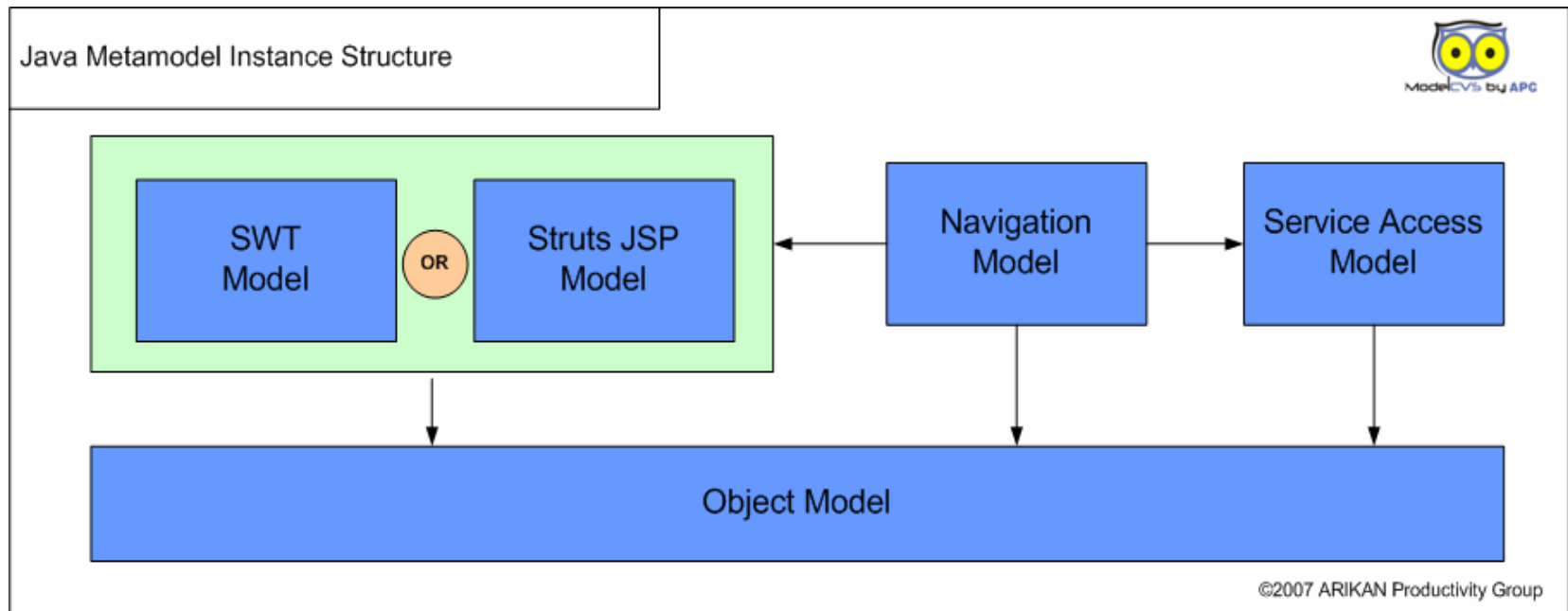
Legacy Renewal

- COBOL, PLI, -> CA Gen
- Code Analysis & Visualization
- Code Reconstruction
- Code Generation

Legacy Renewal



Legacy Renewal



EAB Adapter Generation

- Interface Analyzer
- Complete Generation of EAB Code
- WSDL Service Consumption
- Adapter Generation for Eclipse BIRT and any other callable third party products

EDGE Needs Your Feedback!

- Please complete the conference evaluations
- Your feedback/suggestions are necessary to continue to bring you top-notch events
- Thank You for Coming – See you in 2008!