JavaOne℠

Sun's 2004 Worldwide Java Developer Conference™

The Eclipse Modeling Framework

Introducing Modeling to the Java™ Technology Mainstream

java.sun.com/javaone/sf

Ed Merks, Ph.D.
Software Architect
IBM Canada Ltd.

www.eclipse.org/emf

Sun microsystems

Goal

Eclipse Modeling Framework (EMF)

Learn about modeling and how the Eclipse Modeling Framework can help you write your application in significantly less time, simply by leveraging the data model you've probably already defined, although you might not know it

Agenda

What is Modeling?

EMF Model Definition

Code Generation

EMF Architecture

Demo

Summary

What is Modeling?

Model Driven Architecture (MDA)™

- A software development architecture proposed by the OMG (Object Management Group)
- Application is specified in a high-level platform independent model (PIM)
  - abstracts away underlying platform technology
- Transformation technologies are used to convert PIM to platform specific model (PSM), implementation code, etc.
- Includes several open modeling standards
  - UML, MOF, XMI, CWM

What is Modeling?

Why don't we care about MDA?

1. It's mostly vaporware
2. "Real programmers" know that implementing complex systems by simply transforming a picture, is a "pipe dream"
   - not to mention the fact that it could put us "real programmers" out of business!
3. Smart people know that all the expressive power of Java ™ software can't be available in the model
   - if it was, it wouldn't be any simpler (higher level)
   - it would just be another "programming language"

What is Modeling?

Why should we care about MDA?

1. It's not totally vaporware
2. "Real programmers" know that generating some of the code that we write over and over, must be possible
   - It will simply pave the way for even more complex systems on top of it … programmers like us will never be out of business!
3. Smart people (that have been around long enough) recognize many of the same arguments that were used to oppose high-level languages vs. assembly language

What is Modeling?

Model Driven Development with EMF

- Contrary to most programmers' belief, modeling can be useful for more than just documentation
- Just about every program we write manipulates some data model
  - It might be defined using Java ™, UML, XML Schema, or some other definition language
- EMF aims to extract this intrinsic "model" and generate some of the implementation code
  - Can be a tremendous productivity gain

Agenda

What is Modeling?

EMF Model Definition

Code Generation

EMF Architecture

Demo

Summary

EMF Model Definition

What is an EMF “model”?

- Specification of an application’s data
  - Object attributes
  - Relationships (associations) between objects
  - Operations available on each object
  - Simple constraints (e.g., multiplicity) on objects and relationships
- Essentially the Class Diagram subset of UML

EMF Model Definition

Model Sources

- EMF models can be defined in (at least) three ways:
  1. Java ™ Interfaces
  2. UML Class Diagram
  3. XML Schema
- Choose the one matching your perspective or skills, and EMF can generate the others as well as the implementation code

EMF Model Definition

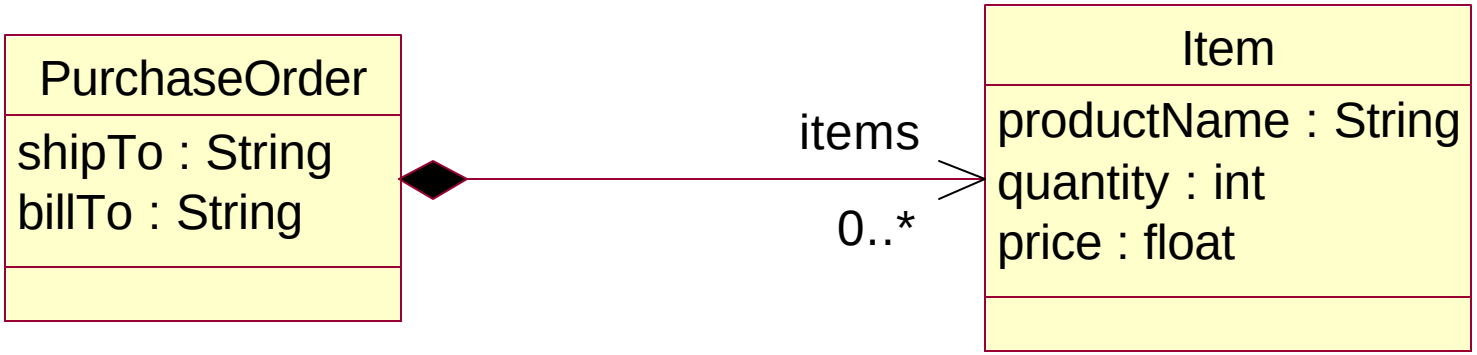
1. Java Interfaces

```
public interface PurchaseOrder {
  String getShipTo();
  void setShipTo(String value);
  String getBillTo();
  void setBillTo(String value);
  List getItems(); // List of Item
}

public interface Item {
  String getProductName();
  void setProductName(String value);
  int getQuantity();
  void setQuantity(int value);
  float getPrice();
  void setPrice(float value);
}
```

EMF Model Definition

2. UML Class Diagram



EMF Model Definition

3. XML Schema

```
<xsd:complexType name="PurchaseOrder">
 <xsd:sequence>
  <xsd:element name="shipTo" type="xsd:string"/>
  <xsd:element name="billTo" type="xsd:string"/>
  <xsd:element name="items"  type="PO:Item"
               minOccurs="0" maxOccurs="unbounded"/>
 </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Item">
 <xsd:sequence>
  <xsd:element name="productName" type="xsd:string"/>
  <xsd:element name="quantity" type="xsd:int"/>
  <xsd:element name="price" type="xsd:float"/>
 </xsd:sequence>
</xsd:complexType>
```

EMF Model Definition

Unifying Java™, XML, and UML technologies

- All three forms provide the same information
  - Different visualization/representation
  - The application's "model" of the structure
- From a model definition, EMF can generate:
  - Java™ implementation code, including UI
  - XML Schemas
  - Eclipse projects and plug-ins

Agenda

What is Modeling?

EMF Model Definition

Code Generation

EMF Architecture

Demo

Summary

Code Generation

Feature Change Notification

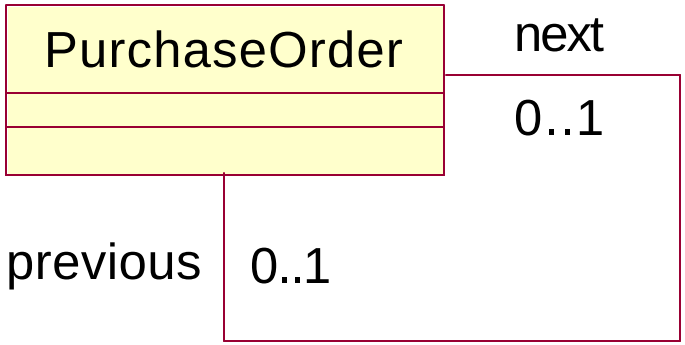
- Efficient notification from “set” methods
  - Observer Design Pattern

```
public String getShipTo() {
  return shipTo;
}

public void setShipTo(String newShipTo) {
  String oldShipTo = shipTo;
  shipTo = newShipTo;
  if (eNotificationRequired())
    eNotify(new ENotificationImpl(this, ... );
}
```

Code Generation

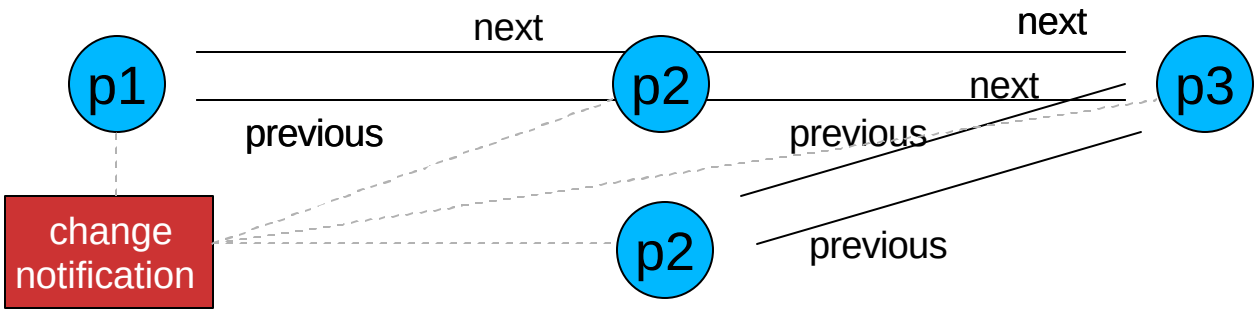
Bidirectional Reference Handshaking




```
public interface PurchaseOrder {
  ...
  PurchaseOrder getNext();
  void setNext(PurchaseOrder value);
  PurchaseOrder getPrevious();
  void setPrevious(PurchaseOrder value);
}
```

Code Generation

Bidirectional Reference Handshaking (cont)




```
p1.setNext(p3);
```

Code Generation

Reflective EObject API

- All EMF classes implement interface EObject
- Provides an efficient API for manipulating objects reflectively
  - Used by the framework (e.g., generic serializer, copy utility, generic editing commands, etc.)
  - Also key to integrating tools and applications built using EMF

```
public interface EObject {
  Object eGet(EStructuralFeature f);
  void eSet(EStructuralFeature f, Object v);
  ...
}
```

Code Generation

Reflective EObject API (cont)

- Efficient generated switch implementation of reflective methods

```
public Object eGet(EStructuralFeature eFeature) {
  switch (eDerivedStructuralFeatureID(eFeature))
  {
    case POPackage.PURCHASE_ORDER__SHIP_TO:
      return getShipTo();
    case POPackage.PURCHASE_ORDER__BILL_TO:
      return getBillTo();
    ...
}
```

Code Generation

Model Persistence

- Save model objects using EMF Resources
  - Generic XML Resource implementation
  - Other Resource implementations possible

```
poResource = ...createResource(..."p1.xml"...);
poResource.getContents.add(p1);
poResource.save(...);
```

p1.xml:

```
<PurchaseOrder>
  <shipTo>John Doe</shipTo>
  <next>p2.xml#p2</next>
</PurchaseOrder>
```

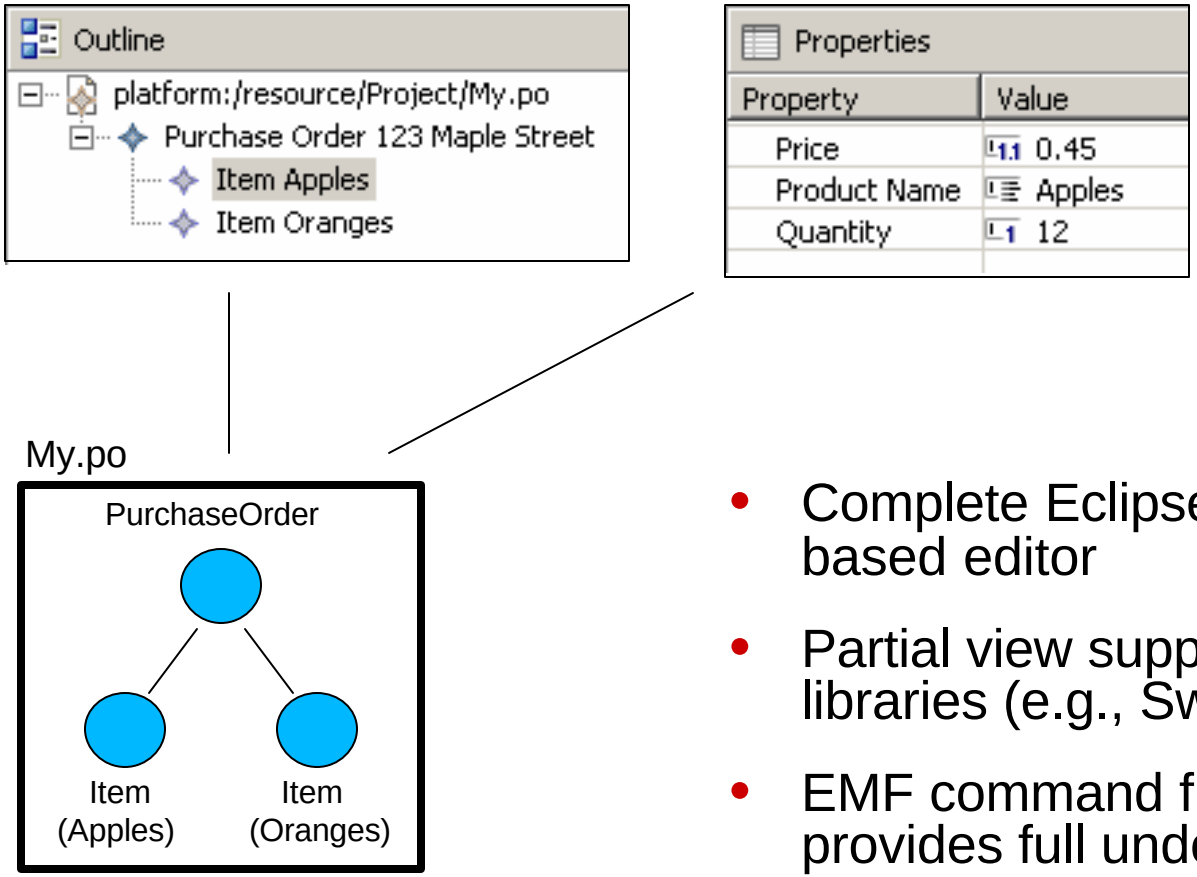
Code Generation

Proxy Resolution and Demand Load


```
PurchaseOrder p2 = p1.getNext();
```

Code Generation

EMF.Edit Model Viewers and Editors



- Complete Eclipse SWT/JFace based editor
- Partial view support for other UI libraries (e.g., Swing)
- EMF command framework provides full undo/redo support

Agenda

What is Modeling?

EMF Model Definition

Code Generation

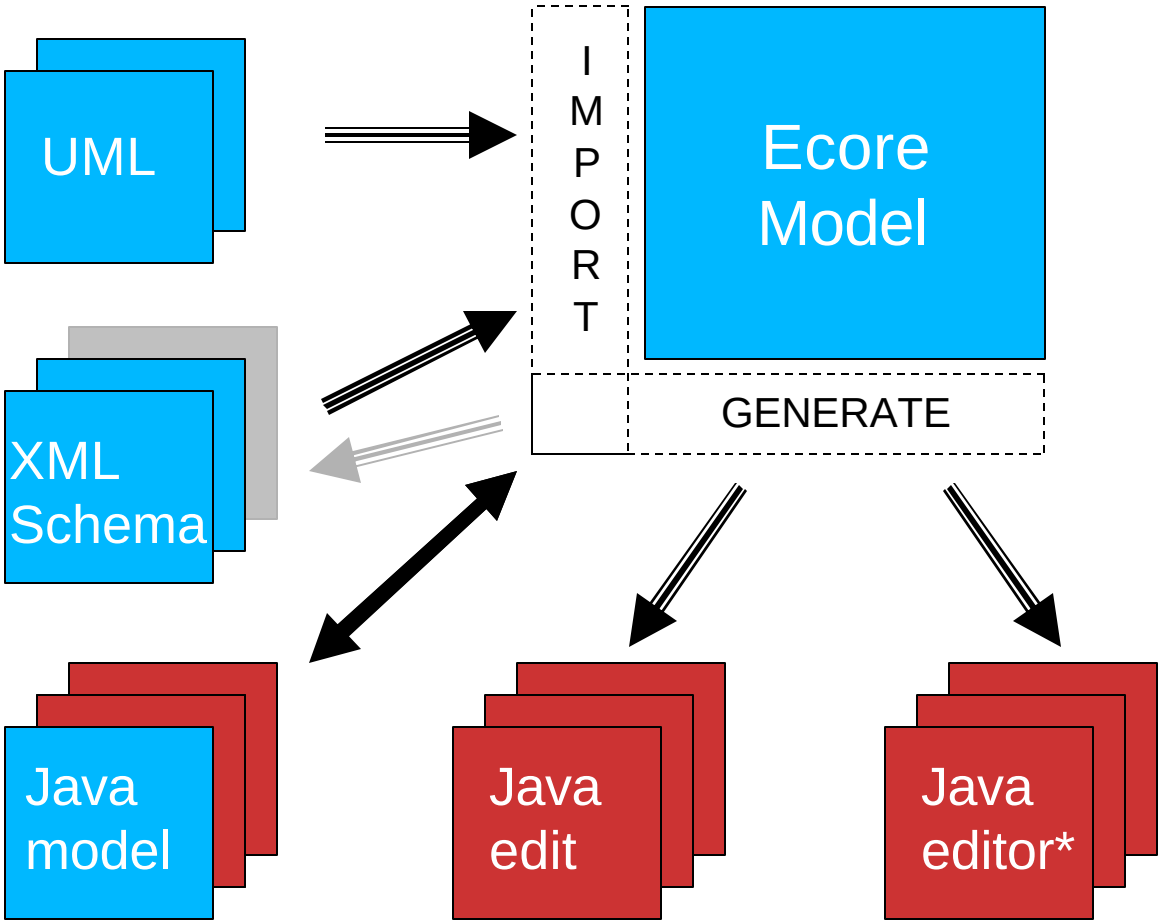
EMF Architecture

Demo

Summary

EMF Architecture

Model Import and Generation



Generator features:

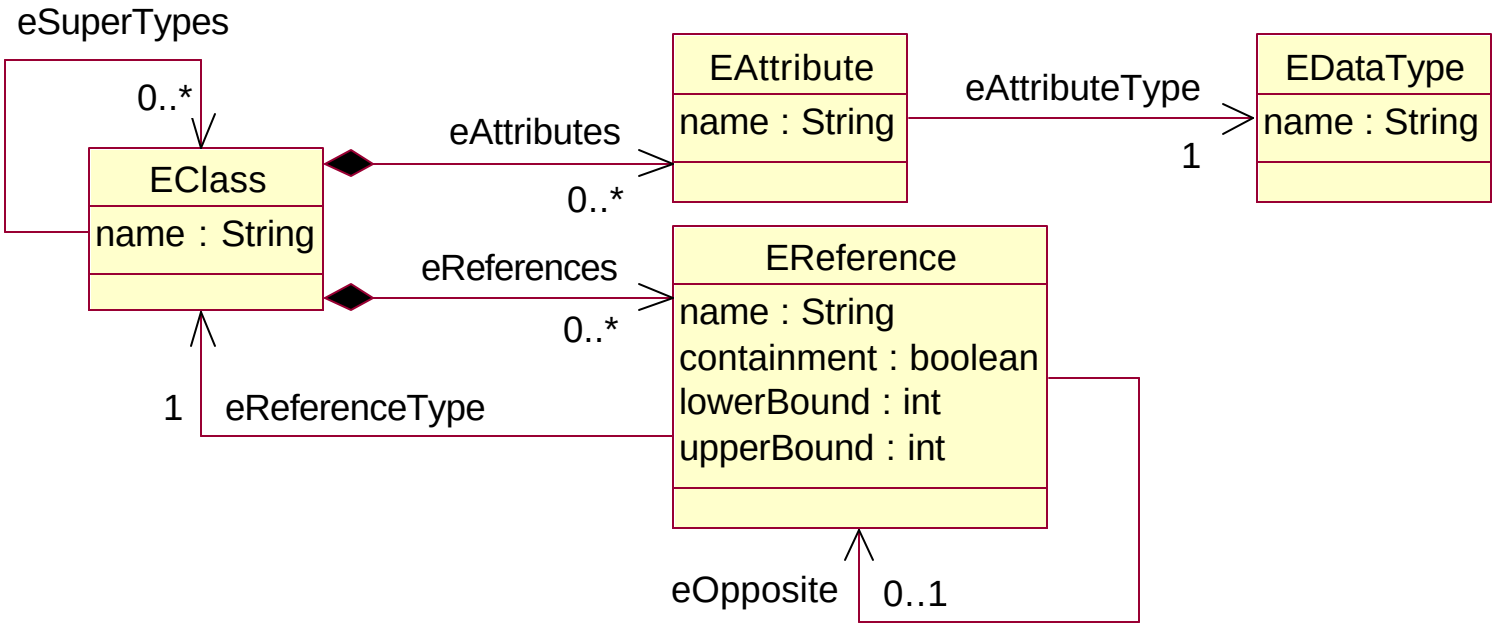
- Customizable JSP-like templates (JET)
- Command-line or integrated with Eclipse JDT
- Fully supports regeneration and merge

* requires Eclipse to run

EMF Architecture

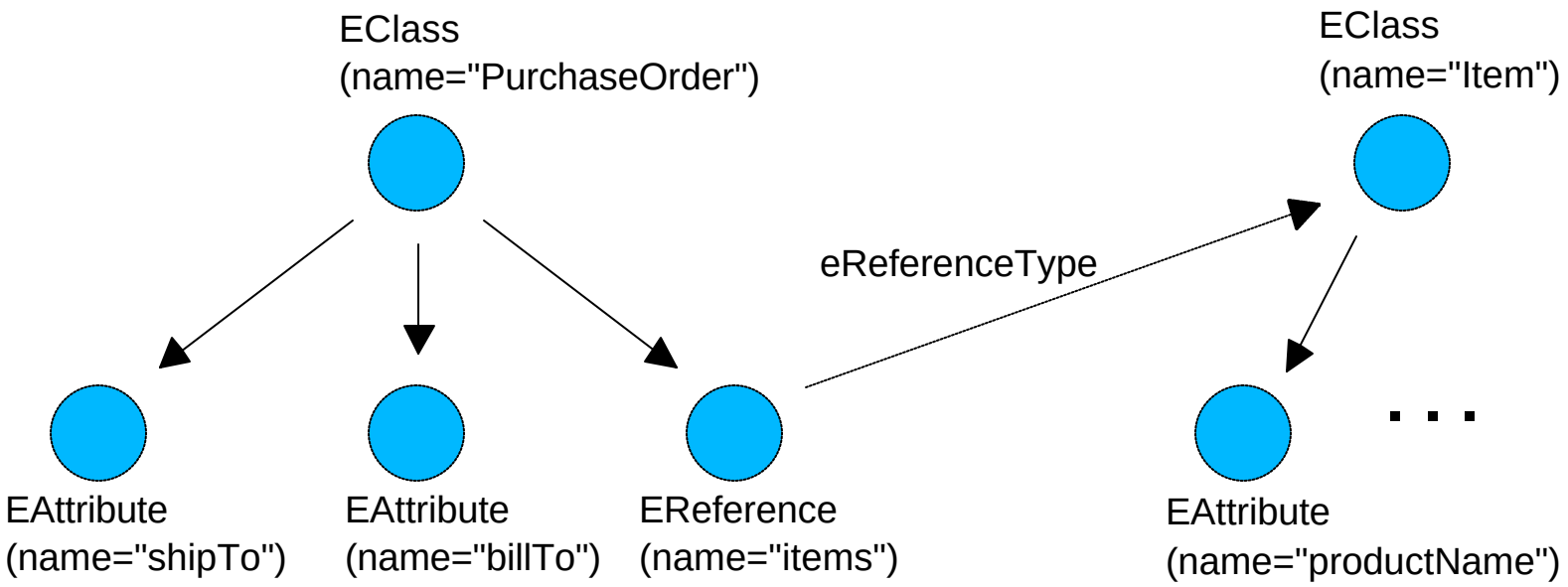
Ecore

- Ecore is EMF's model of a model (metamodel)
  - Persistent representation is XMI



EMF Architecture

PurchaseOrder Ecore Model



EMF Architecture

PurchaseOrder Ecore XMI

```
<eClassifiers xsi:type="ecore:EClass"
  name="PurchaseOrder">
 <eReferences name="items" eType="#//Item"
   upperBound="-1" containment="true"/>
 <eAttributes name="shipTo"
   eType="ecore:EDataType http:...Ecore#//EString"/>
 <eAttributes name="billTo"
   eType="ecore:EDataType http:...Ecore#//EString"/>
</eClassifiers>
```

- Alternate serialization format is EMOF
  - Part of OMG MOF 2 Standard

EMF Architecture

Dynamic EMF

- Given an Ecore model, EMF also supports dynamic manipulation of instances
  - No generated code required
  - Dynamic implementation of reflective EObject API provides same runtime behavior as generated code
  - Also supports dynamic subclasses of generated classes
- All EMF model instances, whether generated or dynamic, are treated the same by the framework

EMF Architecture

Who is using EMF today?

- IBM WebSphere/Rational product family
- Other Eclipse projects (XSD, UML2, VE, Hyades)
- ISV's (TogetherSoft, Ensemble, Versata, Omondo, and more)
- SDO reference implementation
- Large open source community
  - O(1K) downloads/day
  - and growing …

Agenda

What is Modeling?

EMF Model Definition

Code Generation

EMF Architecture

Demo

Summary

Demo

Using EMF

Agenda

What is Modeling?

EMF Model Definition

Code Generation

EMF Architecture

Demo

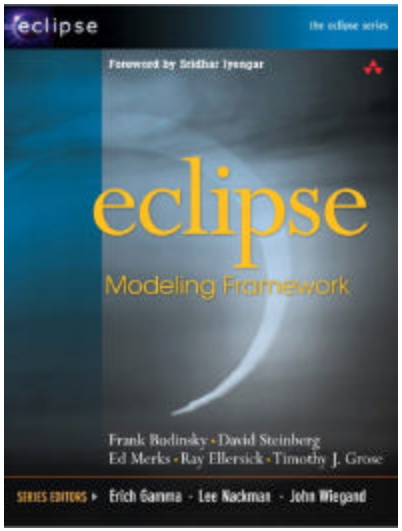
Summary

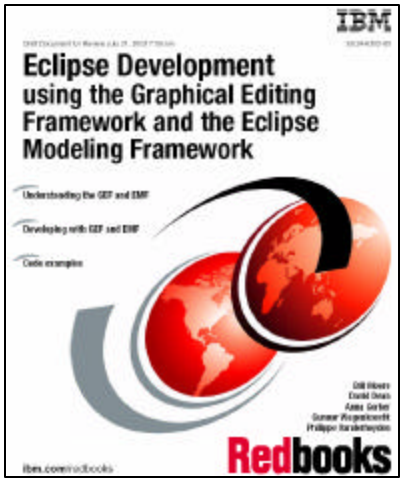
Summary

- EMF is low-cost modeling for the Java™ technology mainstream
- Leverages the intrinsic model in an application
  - No high-level modeling tool required
- Boosts productivity and facilitates integration
- Mixes modeling with programming to maximize the effectiveness of both
- A breakthrough for model-based software development? You be the judge

For More Information

- Eclipse EMF Help
  - overviews, tutorials, API reference
- EMF Project Web Site
  - http://www.eclipse.org/emf/
  - documentation, newsgroup, mailing list, Bugzilla
- Eclipse Modeling Framework by Frank Budinsky et al.
  - Addison-Wesley; 1st edition (August 13, 2003)
  - ISBN: 0131425420.
- IBM Redbook
  - publication number: SG24-6302-00





Q&A

JavaOne℠

Sun's 2004 Worldwide Java Developer Conference™

The Eclipse Modeling Framework

Introducing Modeling to the Java™ Technology Mainstream

java.sun.com/javaone/sf

Ed Merks, Ph.D.
Software Architect
IBM Canada Ltd.

www.eclipse.org/emf

Sun microsystems