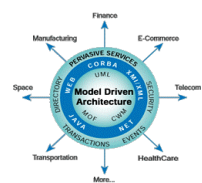


Model-Driven Metadata Integration using MOF 2.0 and Eclipse

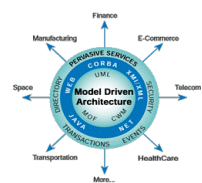
OMG MDA Implementers' Workshop
December 2003, Burlingame

Tracy Gardner, IBM



Agenda

- MOF Recap
- MOF 2.0 Overview
- Eclipse
- Eclipse Modeling Framework (EMF)

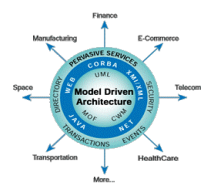


MOF

The OMGs Meta Object Facility is a *model driven framework* for

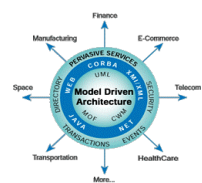
- *Specifying*
- *Constructing*
- *Managing*
- *Interchanging and*
- *Integrating*

metadata in software systems, thus *enabling the flexible integration* of systems



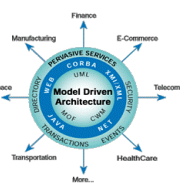
MOF: Original Motivation

- Lots of proprietary metadata sources
- Need to manage IT assets (components, documentation, schemas...)
- New metamodels are proliferating
 - OMG, W3C, etc
 - Customized development methods
 - XML Models
 - Products addressing both new and existing areas
- Need to bridge the gap
 - Common language across metamodels
 - Interoperability and federation of tools, data sources
 - Ease metamodel, tool and middleware development

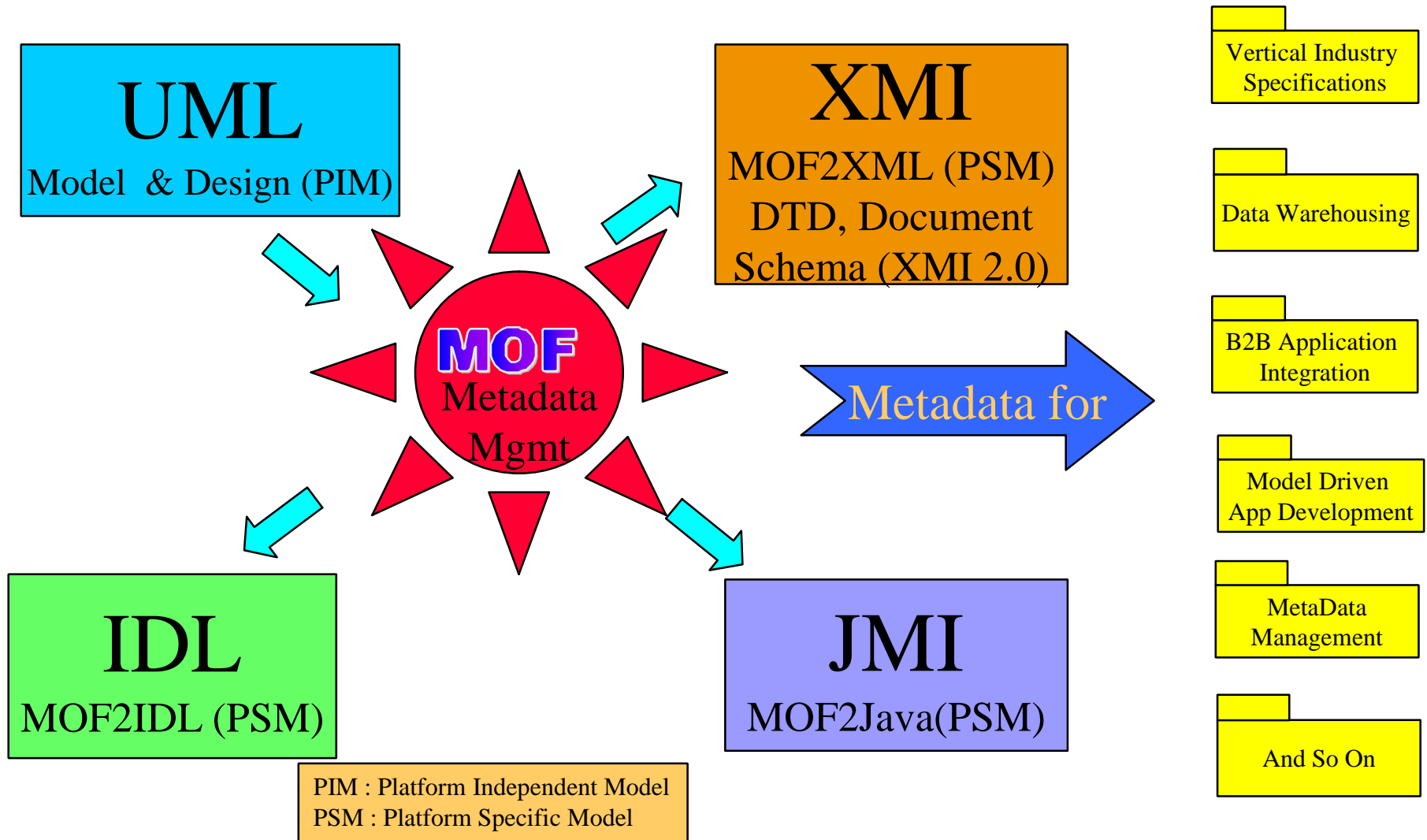


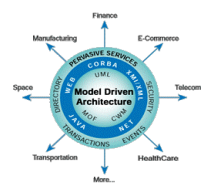
MOF 2.0

- MOF 2.0 is a major revision of MOF, addressing issues that have risen from usage of MOF 1.4
- MOF 2.0 has been designed alongside UML 2.0 and reuses a common core from UML 2.0 Infrastructure
- MOF 2.0 has been adopted by the OMG and is currently in finalization
- Separate RFPs have been issued for additional MOF 2.0 capabilities including: XMI mapping, IDL mapping, Versioning, Transformation
- Migration: A valid MOF 1.4 model can be translated to a valid MOF 2.0 model



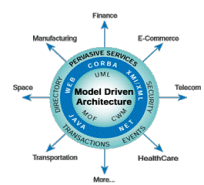
OMG MDA for Metadata Integration



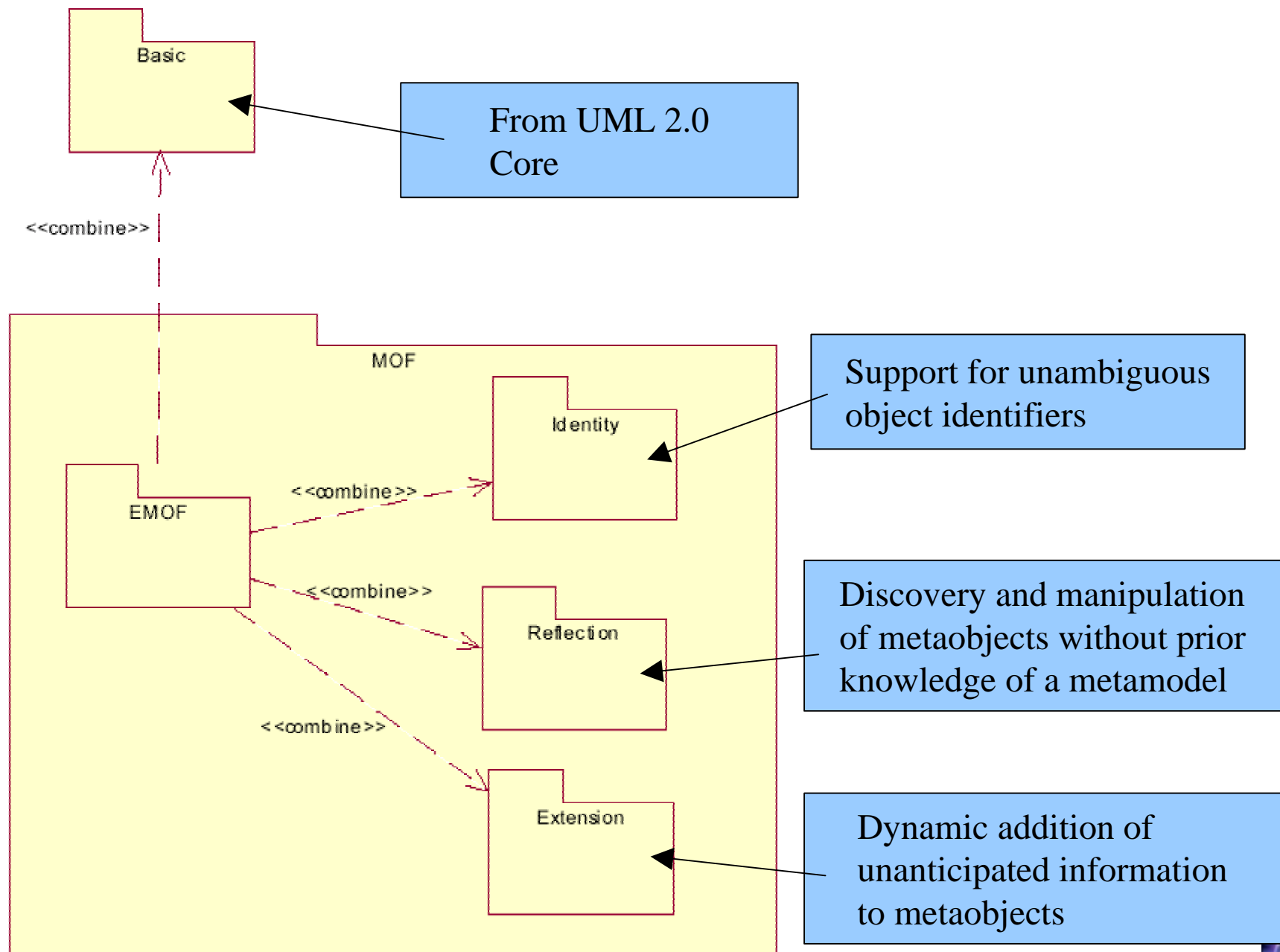


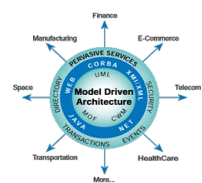
Essential MOF 2

- EMOF is the part of the MOF 2 specification that is used for defining simple metamodels using simple concepts
- EMOF uses object oriented concepts
- MOF 2.0 shares UML 2.0 class diagrams so MOF metamodels can be created with a UML tool
- MOF 2.0 also defines Complete MOF (CMOF) with further advanced capabilities
- This presentation will cover EMOF only



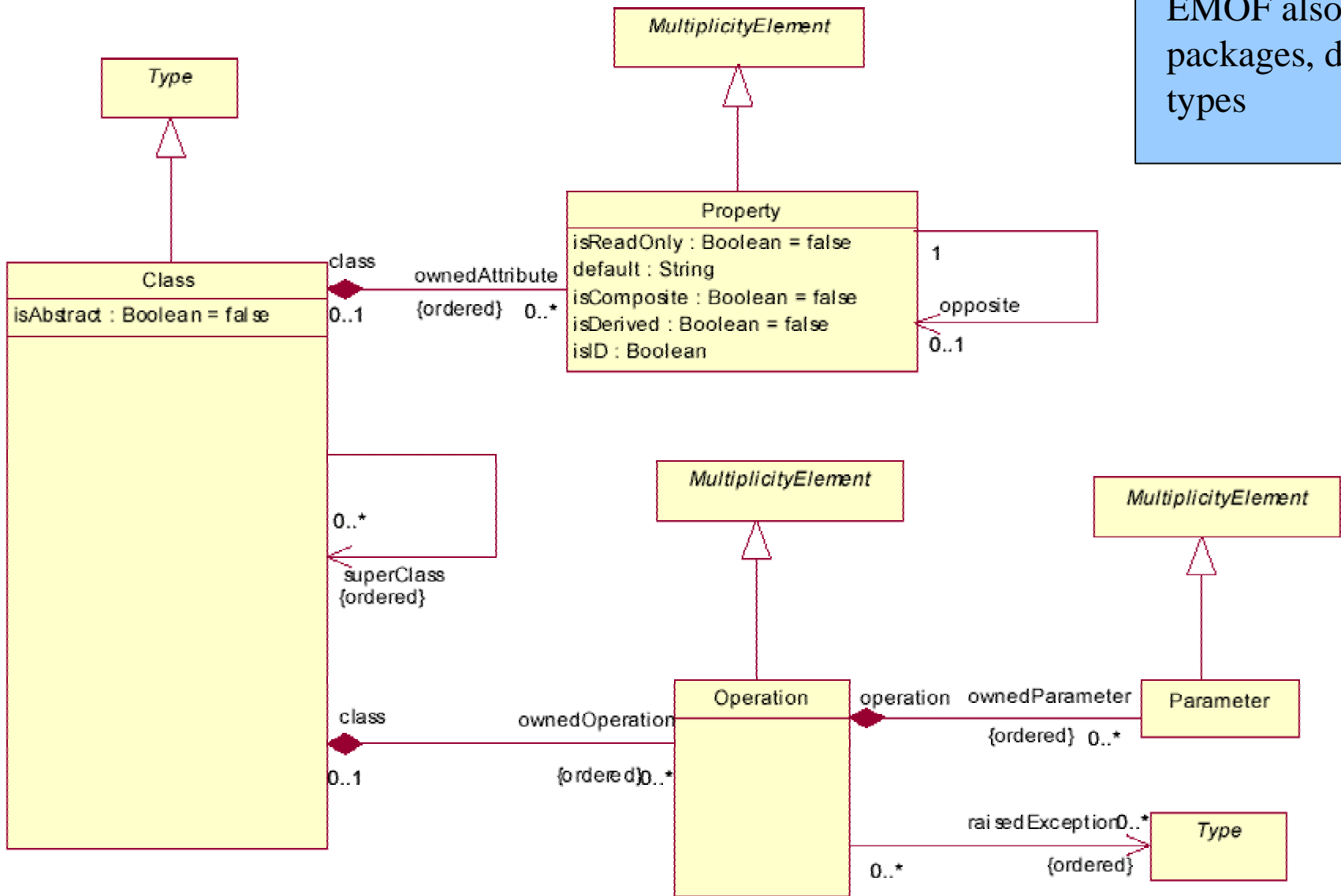
EMOF Overview

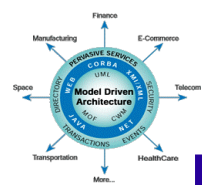




EMOF Classes

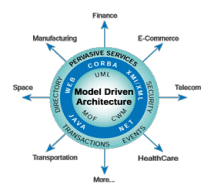
EMOF also includes packages, data types, types





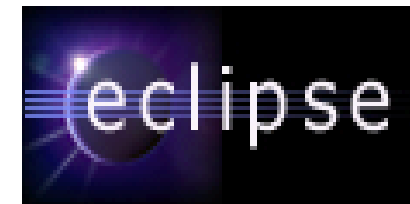
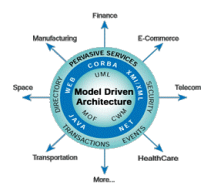
Model Serialization and Manipulation

- EMOF models can be serialized using XMI. The MOF 2 XMI standard provides the mapping rules.
- A separate specification will define mappings from EMOF to Java for typesafe metamodel manipulation. (The JMI JSR defines the MOF 1.4 mappings.) MOF 2 IDL mapping will define IDL interfaces.
- EMOF also specifies reflective operations that can be used to manipulate metamodel elements in a non type-safe manner.



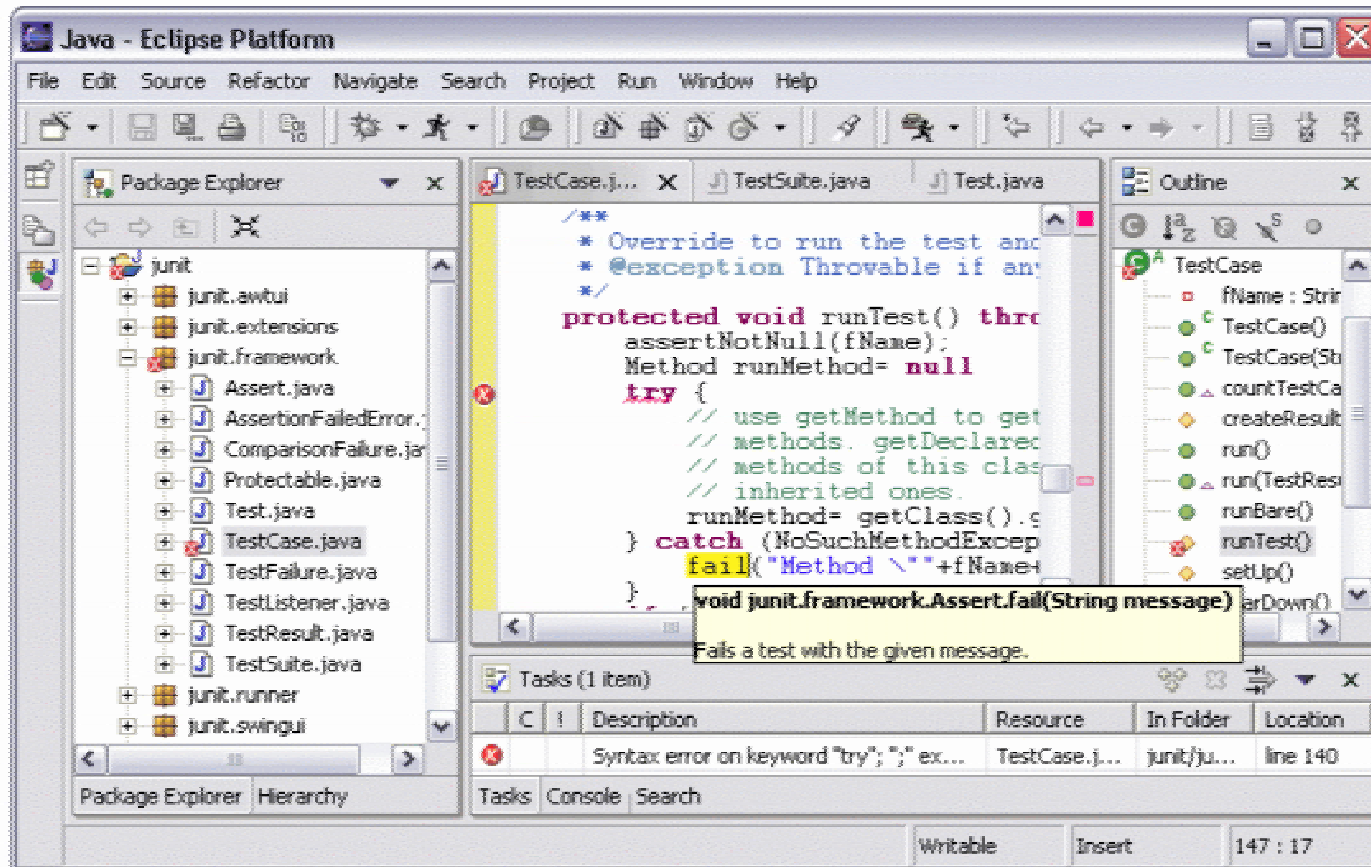
MOF in practice

- The OMG standards specify a platform independent approach to metadata-driven development
- The Eclipse Modeling Framework (EMF) is a platform specific realization of the same concepts (including Java and XMI mappings) which is integrated with the Eclipse tools integration platform



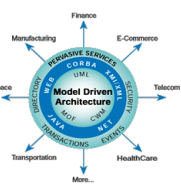
Eclipse

- Eclipse is an 'IDE for anything and nothing in particular'
- Eclipse is open source



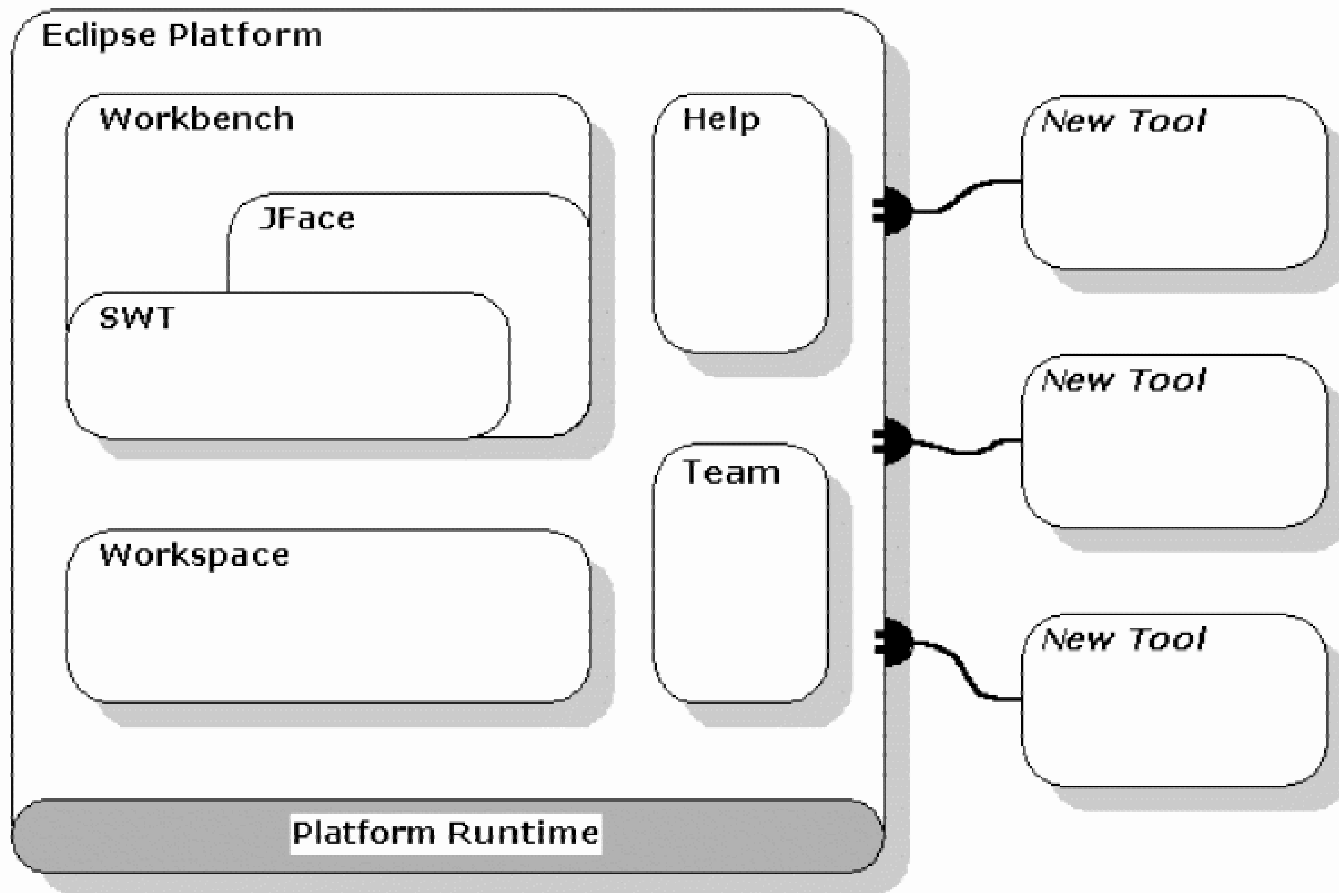
Eclipse editors exist for:

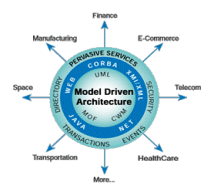
- Java
- UML
- XML
- HTML
- etc, etc



Eclipse Plugin Architecture

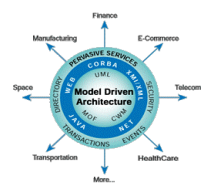
- Eclipse has an open architecture for plugging in new tools





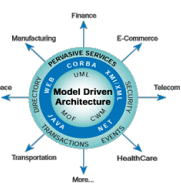
Eclipse Modeling Framework

- **EMF** - The core EMF framework includes a meta model (Ecore) for describing models and runtime support for the models including change notification, persistence support with default XMI serialization, and a very efficient reflective API for manipulating EMF objects generically.
- **EMF.Edit** - generic reusable classes for building editors for EMF models.
- **EMF.Codegen** - capable of generating everything needed to build a complete editor for an EMF model. It includes a GUI from which generation options can be specified, and generators can be invoked.



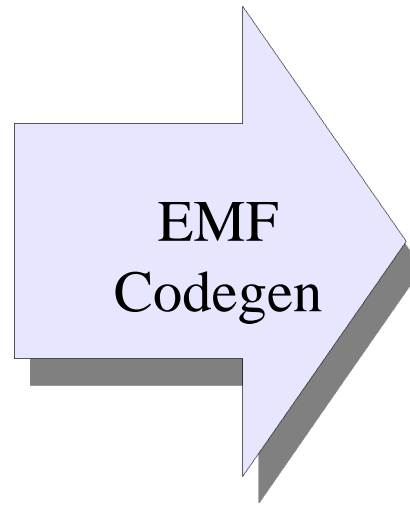
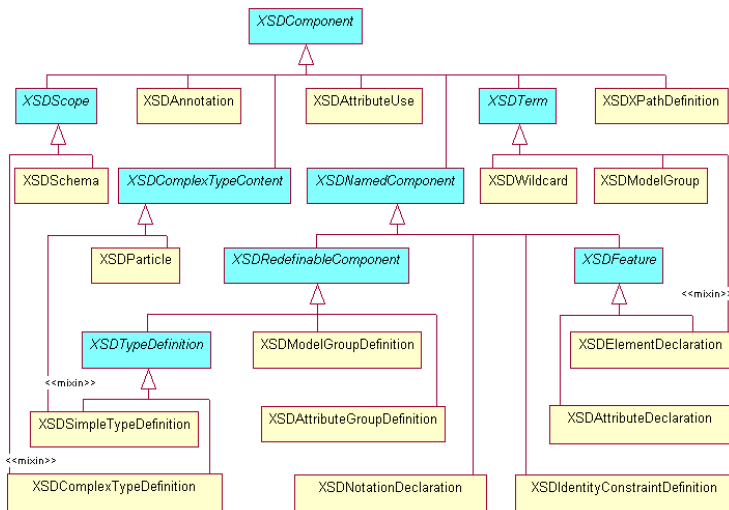
EMF inputs

- EMF is driven by a metamodel, this can be in the form of:
 - Ecore model (native - very close to MOF 2 EMOF)
 - Rose .mdl model
 - XSD Schema (provides automatic serialization according to the schema)
 - MOF 2 EMOF (coming Nov/Dec 2003)
 - Annotated Java



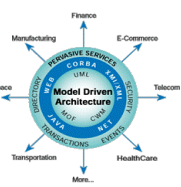
Model-Driven Tools Integration

- EMF uses a model-driven approach to generate XMI serialization code and basic Eclipse editors for MOF metamodels



Java code for manipulation and default serialization of XSD models (optional editor)

Example EMF manipulation code



```
protected XSDSchema createXMI Schema()  
{
```

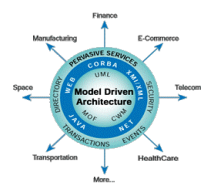
```
    XSDSchema xmiSchema =  
        XSDFactory.eINSTANCE.createXSDSchema();  
    xmiSchema.setTargetNamespace(XMI_URI);  
    xmiSchema.setSchemaForSchemaQNamePrefix("xsd");
```

```
    Map namespaces = xmiSchema.getQNamePrefixToNamespaceMap();  
    namespaces.put(XMI_PREFIX, XMI_URI);  
    namespaces.put("xsd", XSDConstants.SCHEMA_FOR_SCHEMA_URI_2001);
```

```
    // <xsd:attribute name="id" type="xsd:ID"/>
```

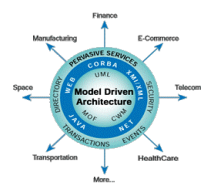
```
    XSDAttributeDeclaration xmildAttribute =  
        XSDFactory.eINSTANCE.createXSDAttributeDeclaration();  
    xmildAttribute.setName("id");  
    xmildAttribute.setTypeDefinition(xmiSchema.getSchemaForSchema()  
                                    .resolveSimpleTypeDefinition("ID"));  
    xmiSchema.getContents().add(xmildAttribute);
```

```
    ....
```



Serialization

- By default EMF supports XMI 2 serialization of models
- Where another format is required a specialized resource handler can be plugged in to handle import/export
 - E.g. XSD has a standard XML-based serialization format.



IBM usage of MOF/EMF

- IBM tools such as WSAD use a variety of EMF models

XSD

Java

BPEL4WS

J2EE

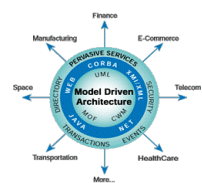
WSDL

WCCM

Ecore

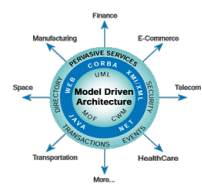
COBOL

...



Summary

- MOF 2.0 is currently under finalization at the OMG
- EMF provides a product quality, open source, model-driven tool for metadata-based tools integration in Eclipse
- EMF is widely used in IBM products for a variety of metamodels
- EMF will support MOF 2.0 EMOF models as input in Nov/Dec 2003



Contact Details and Resources

- Tracy Gardner: tgardner@uk.ibm.com
- MOF 2.0: OMG document [ptc/03-10-04](#)
- MOF 2.0 XMI: OMG document [ptc/03-10-09](#)
- Eclipse: www.eclipse.org
- EMF: www.eclipse.org/emf
- EMF XSD: www.eclipse.org/xsd
- Recommended reading: Using EMF, Catherine Griffin:
www.eclipse.org/articles/Article-Using%20EMF/using-emf.html