The Eclipse Modeling Framework and MDA®
Status and Opportunities

David Frankel Consulting
df@DavidFrankelConsulting.com
www.DavidFrankelConsulting.com

Portions adapted from the book Model Driven Architecture: Applying MDA™ to Enterprise Computing, by David Frankel, published by John Wiley & Sons OMG Press

© 2004 David Frankel Consulting, All Rights Reserved
Agenda

• Model-Driven Metadata Management
• EMF and MDA Standards
  – EMF and the MOF™ Core
  – EMF and XMI® (XML Metadata Interchange)
  – EMF and JMI (Java Metadata Interface)
  – EMF and CMI (CORBA® Metadata Interface)
• Looking Forward: EMF, MDA, and other related technologies
  – Aspect-Oriented Modeling
  – Product Line Practices
  – Intentional Programming
  – Generative Programming
  – Microsoft modeling directions
Model-Driven Metadata Management

MOF-Compliant Metamodel

- MOF-XML Mapping (XMI)
- MOF-Java Mapping (JMI)
- MOF-CORBA Mapping (CMI)

- XMI
- DTD or Schema
- MOF-Compliant Java APIs
- MOF-Compliant CORBA APIs
Metadata Management Scenario
1—Integrated MOF Repository

- MOF CORBA Interfaces
- MOF Java Interfaces (JMI)
- MOF XML (XMI) Documents

= Import/Export

MOF Repository

- UML™ Models
- Data Models
- Workflow Models
- CCM CORBA Interfaces
- B2Bi Collaboration Descriptions
Metadata Management Scenario
2—Federated MOF Repositories

- MOF CORBA Interfaces
- MOF Java Interfaces (JMI)
- MOF XML (XMI) Documents
- Import/Export

MOF Repository
UML Models
B2Bi Collaboration Descriptions

MOF Repository
UML Models
B2Bi Collaboration Descriptions
Metadata Management Scenario
3—Eclipse IDE

- EMF Java Interfaces (Not JMI)
- MOF XML (XMI) Documents
- Import/Export

In-Memory

- UML Models
- Data Models
- Workflow Models
- CCM CORBA Interfaces
- B2Bi Collaboration Descriptions
Eclipse and the MOF Core

- Ecore is closer to MOF 2.0 EMOF (Essential MOF) than MOF 1.x
- IBM has indicated that Ecore will be aligned with EMOF
EMF XMI essentially implements XMI 2.0
—But XMI 2.0 is based on the MOF 1.x Core
EMF XMI in effect maps Ecore to MOF 1.x Core
EMF XMI does not implement XMI 2.0 parameterizations
EMF and JMI
Java Mapping

• EMF Java mapping does not align with JMI
• IBM indicates alignment will happen with JMI for MOF 2.0
  – No new JMI JSR process has been started
EMF and CMI
CORBA Mapping

- CMI for MOF 1.x is impractical for distributed systems
- CMI for MOF 2.0 is excellent
  - Defined by real time CORBA vendors
- EMF has no implementation of either (understandable)
Looking Forward
EMF, MDA, and Other Related Technologies

- Aspect-Oriented Modeling
- Product Line Practices
- Intentional Programming
- Generative Programming
  - Key book: *Generative Programming*, Krzysztof Czarnecki and Ulrich W. Eisenecker
- Microsoft modeling directions
Aspect-Oriented Programming

- Separating different aspects of a system at design time
  - Related to Multidimensional Separation of Concern
- An approach to separation of concern
- Addresses “code tangling” problem
Product Line Practices

• **Product Line**
  – “…a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way. “—Carnegie Mellon Software Engineering Institute

• **Core Asset Development**
  – Capture domain knowledge in the form of reusable assets
    • Define the scope of the domain
    • Model the domain
    • Develop components
    • Define an architecture

• **Production Plan**
  – How to produce systems using the core assets

• **Product Development**
  – Uses core assets according to the production plan
  – Creates individual products
Intentional Programming

- Objective: “Make the source look like the design”
- Programming via intentions
  - High-level abstractions
- Active Source
  - Knows how to compile itself, support editing, rendering, and debugging
    - Behaviors called at programming time
- Source graph
  - Each abstract syntax tree (AST) node has a link to its metadata
  - The primary representation that plug-in modules deal with
    - Not text or graphics vectors
- Transformations from one level of abstraction to another

Charles Simonyi’s new company: Intentional Software
Active Source Graphs

- **Metamodel**: CWM™ Relational Abstract Syntax
- **Model**: CWM Relational Data Model Abstract Syntax Tree
- **Active Source behavior**: invoked at programming time
- **Generic Host Environment**
Generative Programming (GP)

- Synthesis of
  - Aspect-Oriented Programming
  - Product Line Practices
  - Intentional Programming

- Product Line Practices extended to include specifying Domain-Specific Languages (DSLs) as core assets for a product line
- Different DSLs for different aspects of the system
- Generators encapsulate product line knowledge
  - Transformations from one level of abstraction to another

- Extensible development environment based on common technology for representing source graphs in memory
  - Capable of hosting active source for multiple DSLs

- Model Integrated Computing
  - Vanderbilt Institute for Software Integrated Systems
Product Line Practices
Extended to Include Domain-Specific Languages

Individual Product 1  Individual Product 2  • • •  Individual Product n

Individual systems produced via *product development*

Reusable assets for the product line
Created via *core asset development*

Domain Model  Components  Architecture  Specialized Specification Language(s) i.e. DSL(s)

The Sims “Water Line”
Generative Programming (GP)
Design Time Composability

- Component description in some DSL pulled in at design-time
  - Application-specific configuration added
- Generator produces tailor-made component with minimal footprint
- Similar to the latest manufacturing processes
MDA as a Standards Base for GP

• Domain Specific Languages
  – Languages defined via MOF (or UML® profiling)
  – MOF-HUTN specification for textual DSLs
  – MOF lacks the ability to define graphical syntaxes

• Active Source
  – MOF-defined language packaged in a *modeling framework* with components, editor, generator, debugger, rendering support

• Source graphs in extensible development environment for hosting active source
  – JMI, driven by MOF metamodels of each DSL
    • JMI provides link from an AST node to its metadata via MOF reflection

• Definitions of generators
  – MOF Query View Transformations (QVT)

• Interchange of programs among tools when not “in-memory”
  – XMI, driven by MOF metamodels of each DSL
EMF as a GP Environment

- Already in place:
  - Ecore for defining abstract syntax
  - Java mapping for source graph (uses its own reflection, not JMI-MOF reflection).
  - XMI for interchange

- Still needed:
  - Ability to define textual DSLs on top of abstract syntax, using MOF-HUTN specification
  - Ability to define graphical DSLs on top of abstract syntax
    - Implement over GEF
    - DSTC project
    - Extensions to MOF standards to follow
  - Ability to define debugger plug-ins tied to abstract and concrete syntax
Industry Status

• New MOF-based initiatives
  – Business Process Definition Metamodel (OMG)
    • BPMI.org involved
  – Business Rules Metamodel (OMG)
    • Key people from business rules community involved
  – Ontology Definition Metamodel (OMG)
    • Key people from Semantic Web community involved
  – Distributed Management Task Force (DMTF)
    • Moving toward MOF-based metadata
    – Model-Driven data transformations a huge opportunity (CWM)—a killer app for MDA

• Microsoft committed to GP approach
  – But not to MOF

• MOF-Eclipse alignment is important
Trademarks

• “CORBA,” and “MDA,”, “Model Driven Architecture,” “UML,” and “XMI” are registered trademarks of the Object Management Group

• “CWM” and “MOF” are trademarks of the Object Management Group