Combining the Power of Meta-Programming and Meta-Modeling within the OMG MDA Framework.

Jean Bézivin & Nicolas Ploquin

Université de Nantes - CRGNA
Faculté des Sciences et Techniques
2, rue de la Houssinière BP 92208
44322 Nantes cedex 3, France
Jean.Bezivin@sciences.univ-nantes.fr

2nd Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise
Outline

- **Why the MDA?**
  - Rapid paradigm shift from objects to models

- **Basic concepts of the MDA**
  - Revisiting the 4-layer architecture

- **Tooling the MDA**
  - A tour of models and tools

- **Model extraction**
  - Static MX
  - JIT/MP

- **Conclusion**
Why the MDA?

- Objects failed to bring conceptual simplicity
- Platform migration is too frequent and too costly
- New models are emerging
- Models for humans and models for computers
- The unique (object) model is replaced by the multiple model
- Consequence: The middle-war is over
The middleware war is over

- There is no clear winner nor loser
- The next battlefield will be model transformation
- The OMG's Model Driven Architecture (MDA) initiative is aimed at using modelling and meta-modelling to drive the design and implementation of distributed systems.

+ the Next Wonderful Middleware Platform (~2005)
New models are emerging

- From object to components ...
- ... and then to:
  - Rules,
  - Workflow,
  - Services,
  - ...

2nd Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise
Consequence: having to deal simultaneously with several models of different semantics.

Diagram:

UML model →\rightarrow Java model
From contemplative to productive

From human-readable to computer-understandable (XMI)

2nd Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise
Basic concepts of the MDA

Revisiting the 4-layer organization

Abstract Syntax Systems Compared

<table>
<thead>
<tr>
<th>Technology #1</th>
<th>Technology #2</th>
<th>Technology #3</th>
<th>Technology #4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(formal grammars)</td>
<td>(MOF + OCL)</td>
<td>(XML Meta-Language)</td>
<td>(Ontology engineering)</td>
</tr>
</tbody>
</table>

MOF

The UML meta-Model

A Specific UML Model

A Specific phenomenon corresponding to a UML Model

A XML DTD or Schema

KIF Theories

XPath, XSLT
+RDF, OIL, DAML +etc.

A XML DTD or Schema

Description Logics
+Conceptual Graphs +etc.

Upper Level Ontologies

Model serialisation

Technologies evaluation grid
A model $M$ is a simplified representation of the world, as a matter of fact of only a part $S$ of the world called the system.
Limited Substituability Principle

The purpose of a model is always to be able to answer some questions in place of the system, exactly in the same way the system itself would have answered similar questions.

System

\[\text{ask()}\]

represents

Model

\[\text{ask()}\]
The global MDA model space

- The development software cycle is populated with models
  - Models are of unequal importance
  - The model space is structured
  - Models are linked in a complex organization network
  - The content of each model is defined (constrained) by a corresponding meta-model (ontology)
  - The model space is constantly broadening starting from the essential models (Domain, Service, Resource)

- Many different kinds of models
  - Business models and computer models
  - Models of product & models of processes
  - Object, component, rule, workflow, service models among others
  - Legacy (Cobol, RDB) and NT (Web, SOAP, etc) models
  - PSMs, PIMs, PDMs,
  - etc.
Various kinds of models

- Products and processes
- Legacy and components
- Static and dynamic
- etc.

```
^ System
  ^ System
    ^ Dynamic System
    ^ Static System
      ^ Model
        ^ Dynamic Model
          ^ Static Model
```

System represents Model
What is a Meta-Model?

The correspondence between a model and a system is defined by a meta-model.
The 3+1 Layers

Level $M^3$

- the MOF

Level $M^2$

- MMM
- the UPM
- the UML
- the CWM

Level $M^1$

- a UML model $m$
- another UML model $m'$

Level $M^0$

- a particular use of $m$
- another use of $m$

(This is not a modeling layer, it is the real world.)
Local and global definitions

\[ M_1 \rightarrow \text{aCat} \rightarrow \text{instanceOf} \rightarrow \text{Cat} \]

\[ M_2 \rightarrow \text{StkInstance} \rightarrow \text{instanceOf} \rightarrow \text{StkClass} \]

\[ M_3 \rightarrow \text{Node} \rightarrow \text{meta} \rightarrow \text{Link} \rightarrow \text{meta} \rightarrow \text{StkClass} \rightarrow \text{meta} \]

2nd Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise
Global organization

MOF

Smalltalk Meta-model

Smalltalk model

Real world

2nd Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise
Meta-programming vs. Meta-modeling

\[ M_3 \]

\[ M_2 \]

\[ M_1 \]

Meta-class

STK::Iof

STK::Iof

STK::Iof

STK::Iof

STK::Iof

MOF::Class

STK::MetaClass

STK::Class

STK::Instance

Cat class

Cat

Felix

Université des Arts et Métiers
Tooling the MDA: Sample

- University of Berne MOOSE [http://www.iam.unibe.ch/](http://www.iam.unibe.ch/)

and much more...
Example: The Y cycle

- PIMs (Platform Independent Models)
- PSMs (Platform Specific Models)
- Merging/binding phase
- PDMs (Platform Description Models)
What is a pure "MDA tool"?

- It implements some operations on models or meta-models
- It is compliant with the main MDA recommendations (UML, MOF, XMI, etc.)
- It is interoperable with other MDA tools
- It is compatible with the MDA vision (meta-model driven)

Operations on M. & MM.
- M. checking
- M. transformation
- M. merging
- M. presentation
- M. & MM. browsing
- Code generation
- Reverse engineering
- MM. Alignment
Source code model extraction

This is a static system.

- a Cobol program
- a MOF meta-model
- Syntax analyser (lex, yacc, etc.)
- a Cobol model

This is a static model of a static system

Meta-model driven SC (source code) model extraction
Example: Semantor Explorer

Full Syntax Source Code Model Extraction

(once extracted, models can be worked on)
Round-Trip Engineering

UML model

Forward engineering

Java Program

Reverse engineering

2nd Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise
**MM-driven SC model extraction**

- What kind of MM for source model extraction?
  - Full syntax
    - 100% of the info extracted
    - Operation is reversible
  - Partial syntax
    - Non reversible operation
  - Semantic
    - Need some heuristics
    - Examples
      - Process classes
      - Exception classes
      - Interface classes
      - Business classes

- Important (and obvious) remark:
  - Class instances only exist at run-time (usual instances, processes, exceptions, etc.)
Part of the C# MetaModel (grammar)
JIT/MP: a new concept

- Just In Time Model Production
- An execution of a program dynamically produces an (XMI) model of its current situation
  - on demand,
  - Periodically,
  - on given events (internal or externals), ...
- The produced model corresponds to an (XMI) MM that has been directly read by the program
- Produced models may be concurrently used and combined
- The nature of the produced models may change in time (MM-driven)
Combining the power of meta-modeling and meta-programming

This is a dynamic model.

Introspection at work

- a C# program execution
- a C# meta-model
- a C# model

- read XMI
- write XMI

[This may be the basic strategy for modern software maintenance]
Challenge

How to associate instructions to different elements of the meta-model that will specify how the model may be extracted

Hint: these instructions are C# code that uses the introspection API of C#

This allows defining complex heuristics (e.g. finding patterns, etc.)
Interesting question

- A C# source program
- MM-driven model extraction
- A model

- A C# program execution
- JIT model production
- A model

?
Several migration paths

Meta-Model Driven software maintenance and evolution system (UML, MOF, XMI, etc.)

MS/JUMP

Cobol
ADA
Smalltalk
Java
C#
Eiffel
etc.

etc.
Java
C#
Eiffel

MS/JUMP

2nd Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise
Conclusions

- *Meta-modeling and meta-programming are two orthogonal solutions*
- *They may be combined*
- *JIT/MP is a very powerful technique with tremendous potential*