Model-Driven Development: Concepts and Reality
Stephen Mellor
Model-Driven Development

• Model-driven development is the idea that we can transform models into systems.

• Models can be of many kinds:
  – Parametrics for controllers
  – Control diagrams
  – Programs
  – UML

• We all use model-driven development today.
## A short history of MDD

<table>
<thead>
<tr>
<th>Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1979</td>
<td>Structured Design: Yourdon and Constantine</td>
</tr>
<tr>
<td>1981</td>
<td>Structured Analysis: De Marco</td>
</tr>
<tr>
<td>1985</td>
<td>Structured Devpt/RT: Ward and Mellor</td>
</tr>
<tr>
<td>1988</td>
<td>OOA: Shlaer and Mellor</td>
</tr>
<tr>
<td>1988</td>
<td>OO Design: Booch</td>
</tr>
<tr>
<td>1988</td>
<td>OMT: Rumbaugh et al</td>
</tr>
<tr>
<td>1992</td>
<td>Object Lifecycles: Shlaer and Mellor</td>
</tr>
<tr>
<td>1997</td>
<td>UML 1.1: Three Amigos</td>
</tr>
<tr>
<td>2002</td>
<td>Executable UML: Mellor and Balcer</td>
</tr>
<tr>
<td>2004</td>
<td>UML 2.0: Cast of thousands</td>
</tr>
</tbody>
</table>
Why MDD now?

Knowledge
Individuals
Projects
Companies

Market Usage
Sketchers
Blueprinters
Executable
Modelers

System Complexity
Programs
Systems
Systems of Systems

Standards
Methods
UML
Interchange
Embedded Developers

- 14% currently use UML
- 25% plan to use UML by 2007

• By far the majority of UML users are sketchers.
  
  (See Fowler, for example.)
Differing Expectations

Modeling formalism should mirror the implementation.

Model should mirror my mind

Modeling formalism should mirror the knowledge we’re capturing.
UML Tools Use Different Subsets of UML

• *Meaningful* interchange between tools is difficult.

What's the problem? We have XMI already!
Semantics Backplane

- Select a *system specification semantic* subset.
- Every tool uses the *same* executable subset....
- ...which implies a standard
Executable UML Foundation

• The Executable UML Foundation defines:
  – An executable subset
  – A definition of the execution semantics of that subset
  – A base semantics
Next Steps

• There has to be a complete separation between:
  – Subject matters (that capture aspects)
  – Metamodels
  – Notation

• With model transformation technology, each ontology—preferably expressed using the same metamodel—can be combined into a system.

• Separating notation enables domain-specific languages.
DSLs displace UML but not MDA

- A domain-specific language is a graphical language specific to a particular domain:
  - VCR controls
  - Fax machines
  - Chemical plant
  - Train control

- UML would be used for domains with no pre-existing standard language, or for software