A Brief History of MDD

UML 2.0: Cast of thousands 2004
UML 1.1: Three Amigos 1997
Object Lifecycles: Shlaer and Mellor 1988
OO Design: Booch 1988
Structured Design: Yourdon and Constantine 1979
Structured Analysis: De Marco 1981
Structured Devpt /RT: Ward and Mellor 1985
Executable UML: Mellor and Balcer 2002

Copyright © 2013 Stephen J. Mellor. All rights reserved.
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
- SysML

We all use model-driven development today.
Model-Driven Architecture

- An OMG initiative to develop standards based on the idea that modeling is a better foundation for developing and maintaining systems
- A brand for standards and products that adhere to those standards
- A set of technologies and techniques associated with those standards
Model-Driven Development/Engineering

- An OMG initiative to develop standards based on the idea that modeling is a better foundation for developing and maintaining systems
- A brand for standards and products that adhere to those standards
- A set of technologies and techniques for transforming models
The Agile Critique

Models are bad (they say), because models:

- Don’t run.
- Can’t be tested automatically.

Models are bad (they say) because they are “documentation” which:

- Has no correspondence to the code
- Is extra work to build…
- …and maintain (or throw away)
Agile Manifesto

“We are uncovering better ways of developing software by doing it and helping others do it.

We value:

- *Individuals and interactions* over processes and tools
- *Working software* over comprehensive documentation.
- *Customer collaboration* over contract negotiation.
- *Responding to change* over following a plan.”
What problem is agility meant to solve?

- Delayed feedback on requirements
- Right solution to the wrong problem
- Unsustainable pace
- Poor customer relations
- Long time-to-market
What solution does agility propose?

- Immediate feedback by executing software
- Frequent releases with consequent feedback
- Timeboxed deliveries
- Customer on Site (aka Whole Team on Site)
- Incremental delivery of working code
Signatories to the Agile Manifesto

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Stephen Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

www.aanpo.org
But models are executable too!

Code is so very important because it runs, right?

An executable model runs, so it can be verified, right?

So, if a model is executed, it is as good as code, right?

Argh!!!!!!

Yes!

Yes...

No. Code is the most important thing.
Models, Models, Models

Models

Sketch

Blueprint

Executable
The Executable UML Foundation defines:

- An executable subset
- A definition of the execution semantics of that subset
- A base semantics
Model Capture

An application model is captured in a *metamodel*.

- **Class**
  - Class ID: 100
  - Name: Recipe

- **State**
  - Class ID: 101
    - State #: 1
      - Name: Filling
    - State #: 2
      - Name: Cooking
    - State #: 3
      - Name: Emptying
  - Class ID: 102
    - State #: 1
      - Name: ....
    - State #: 2
      - Name: ....
    - State #: 3
      - Name: ....

- **Rules**
  - Create Batch( Amount of Batch, Recipe Name)
  - Filled( Batch ID )
  - Temperature Ramp
  - Complete( Batch ID )
  - Emptied( Batch ID )
Software Architecture

An application-independent software architecture proclaims and enforces the organization of:

- data
- control
- structures
- time

An architecture is realized as a model compiler.
Execution Engine

A limited set of reusable components sufficient to execute Executable UML.

- Data access
- State machines
- Activation of threads
- Order of execution
Transformations

A generator executes transformation rules against the populated metamodel.

- Read the repository
- Make substitutions

```
[template classToStruct( s: Class)]
Struct [s.name/] { ... };
[/template]
```

- Generate text

<table>
<thead>
<tr>
<th>Class ID</th>
<th>Name</th>
<th>Descr'n</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Recipe</td>
<td>.....</td>
</tr>
<tr>
<td>101</td>
<td>Batch</td>
<td>.....</td>
</tr>
<tr>
<td>102</td>
<td>TempRamp</td>
<td>.....</td>
</tr>
</tbody>
</table>

Struct Recipe { ... };
Struct Batch { ... };
Struct TempRamp { ... };
Model Compilers

- Build a complete system from models \textit{consistently}
- Translate into the selected architecture for the system
- Leverages expertise of best architects,
- Captures that expertise.
- Reuses expertise across many applications
- Maintains application and architecture separately
How can modeling be agile?

- Immediate feedback by executing models*
- Frequent releases with consequent feedback
- Timeboxed deliveries
- Customer on Site (aka Whole Team on Site)
- Incremental delivery of working models

* The Agile Manifesto uses “software,” not “code.”
Want to learn more?

**Executable UML: A Foundation for Model-Driven Architecture**, Stephen J. Mellor, Marc Balcer

**Comprehensive language introduction and reference**

An Open-Source Reference Implementation for a Foundational Subset for Executable Models

[EXECUTABLE UML](http://fuml.modeldriven.org)

Model Driven Solutions under contract to Lockheed Martin Corporation

Semantics Of A Foundational Subset For Executable UML Models (fUML)


Concrete Syntax for a UML Action Language: Action Language For Foundational UML (ALF)

[omg.org/spec/ALF/1.0.1/Beta3/PDF/](http://omg.org/spec/ALF/1.0.1/Beta3/PDF/)
Thank you