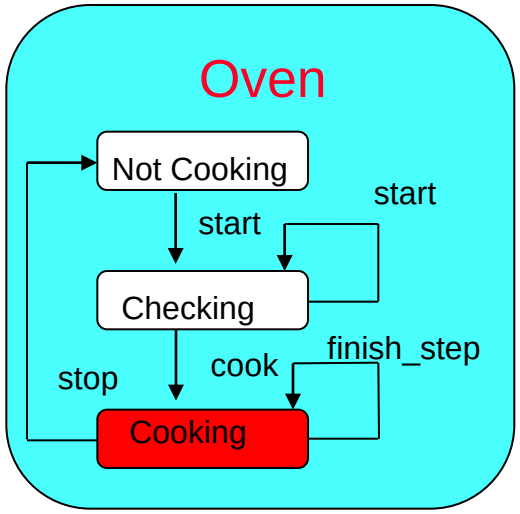
Model-Driven Development:
Concepts and Reality

Stephen Mellor

Model-Driven Development

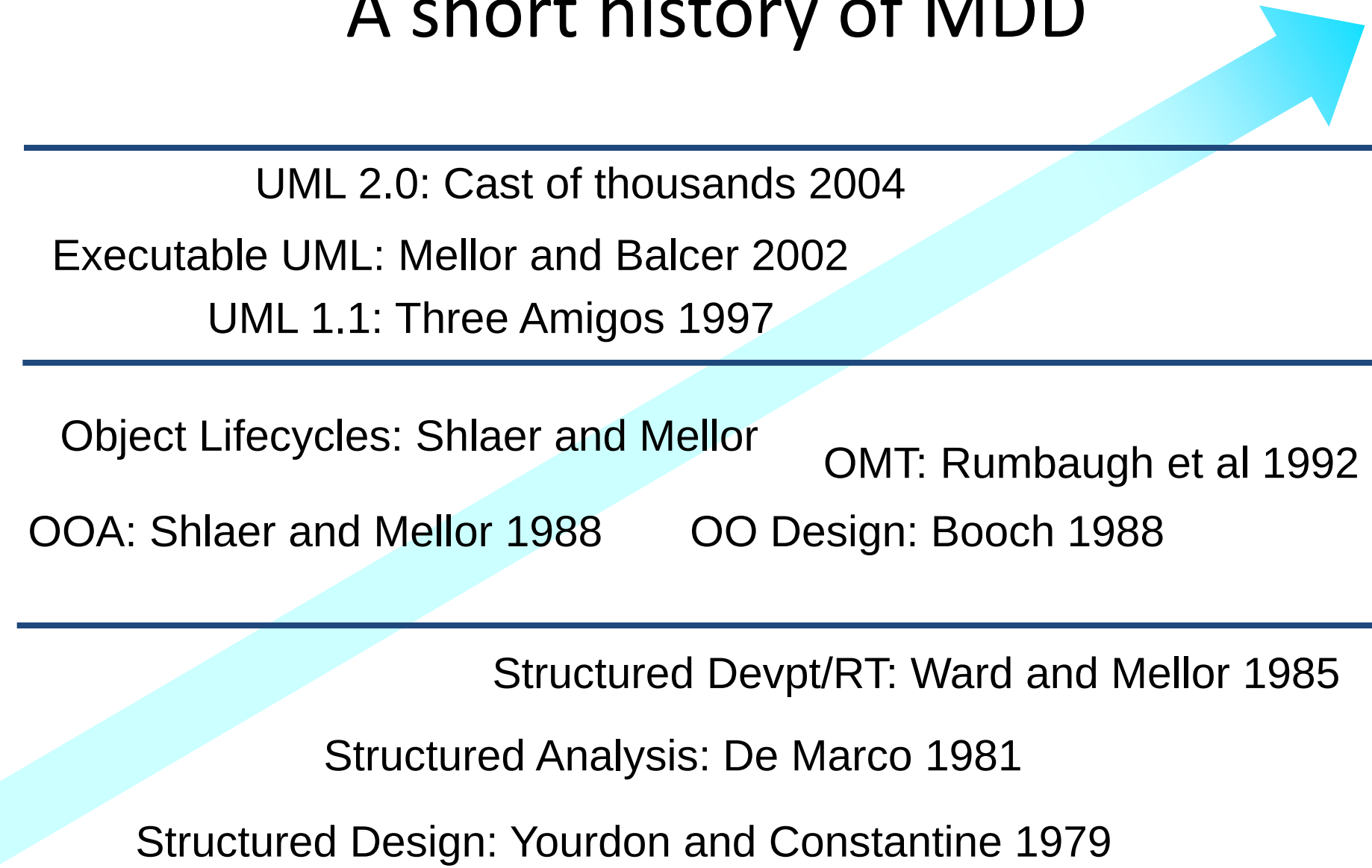
- Model-driven development is the idea that we can transform models into systems.

- 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



UML 2.0: Cast of thousands 2004

Executable UML: Mellor and Balcer 2002

UML 1.1: Three Amigos 1997

Object Lifecycles: Shlaer and Mellor

OMT: Rumbaugh et al 1992

OOA: Shlaer and Mellor 1988

OO Design: Booch 1988

Structured Devpt/RT: Ward and Mellor 1985

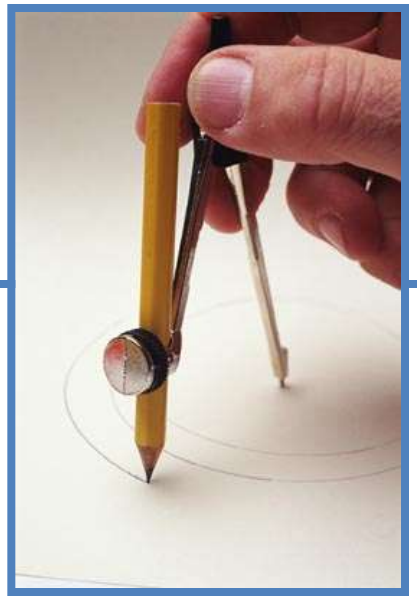
Structured Analysis: De Marco 1981

Structured Design: Yourdon and Constantine 1979

Why MDD now?

Knowledge
Individuals
Projects
Companies

Market Usage
Sketchers
Blueprinters
Executable Modelers



<u>System Complexity</u>
Programs
Systems
Systems of Systems

<u>Standards</u>
Methods
UML
Interchange

Embedded Developers

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

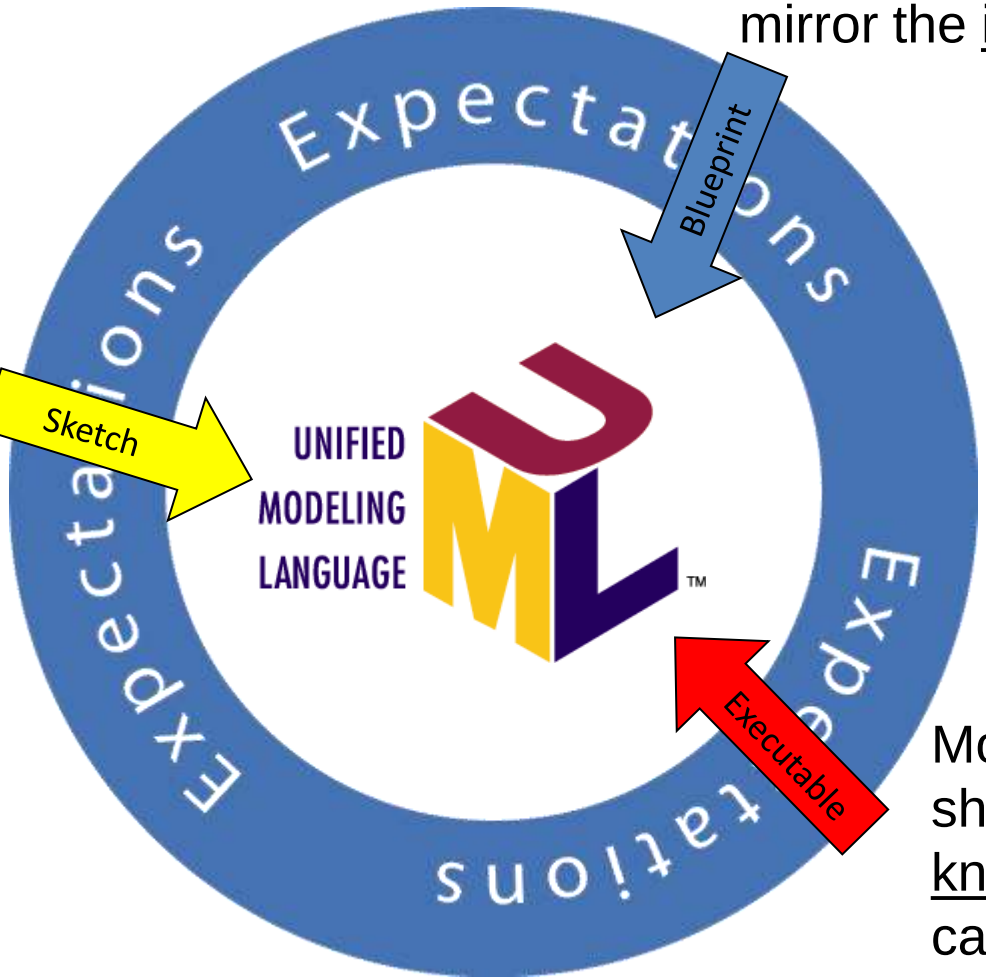
- VDC Embedded Software Developers' Requirements and Preferences report 2005

- By far the majority of UML users are sketchers.

  (See Fowler, for example.)

Differing Expectations

Modeling formalism should mirror the implementation.

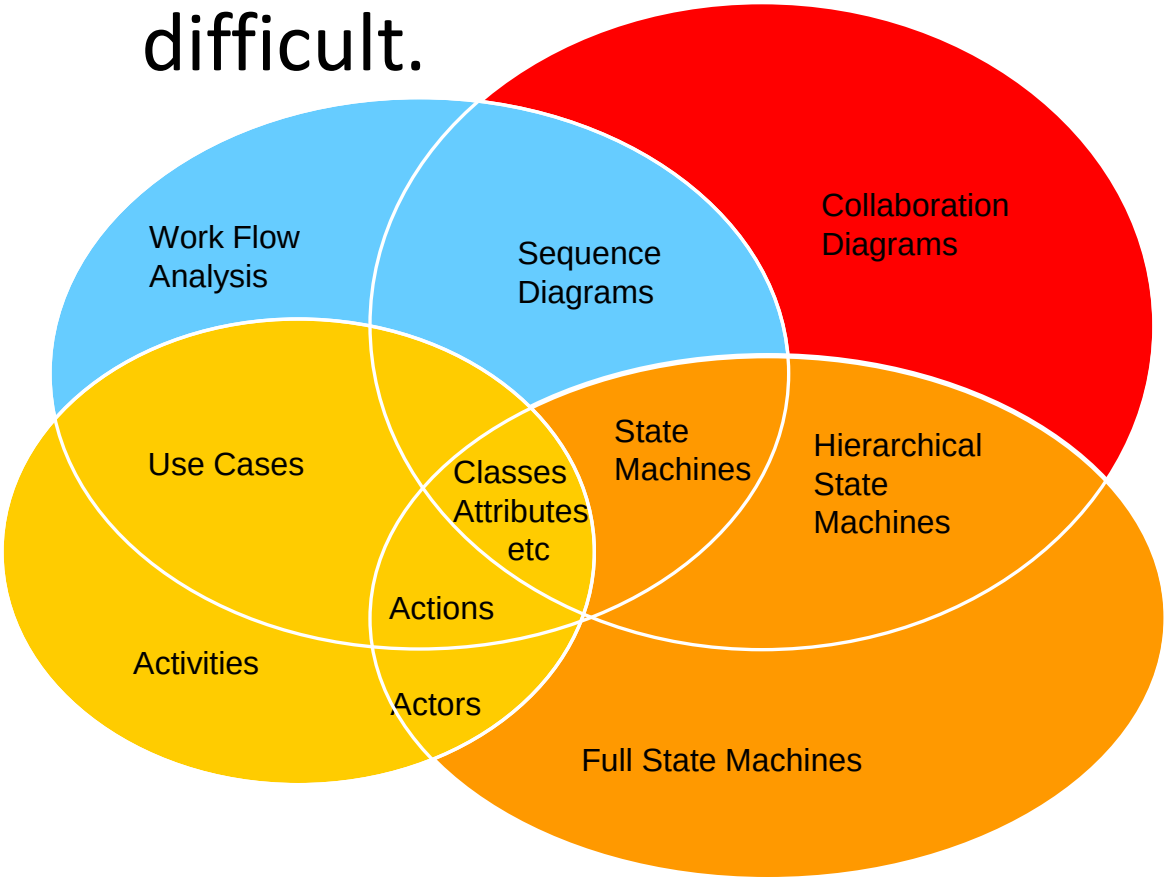


Model should mirror my mind

Modeling formalism should mirror the knowledge we're capturing.

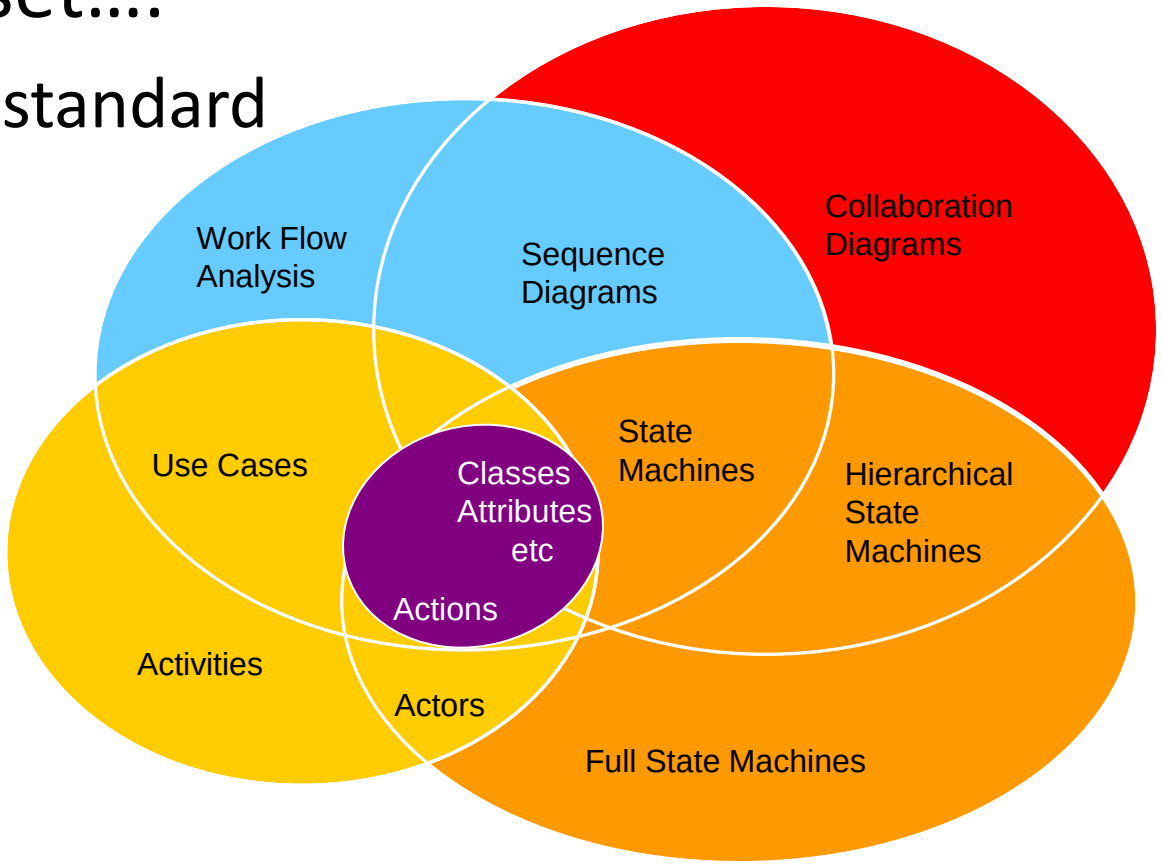
UML Tools Use Different Subsets of UML

- *Meaningful* interchange between tools is difficult.



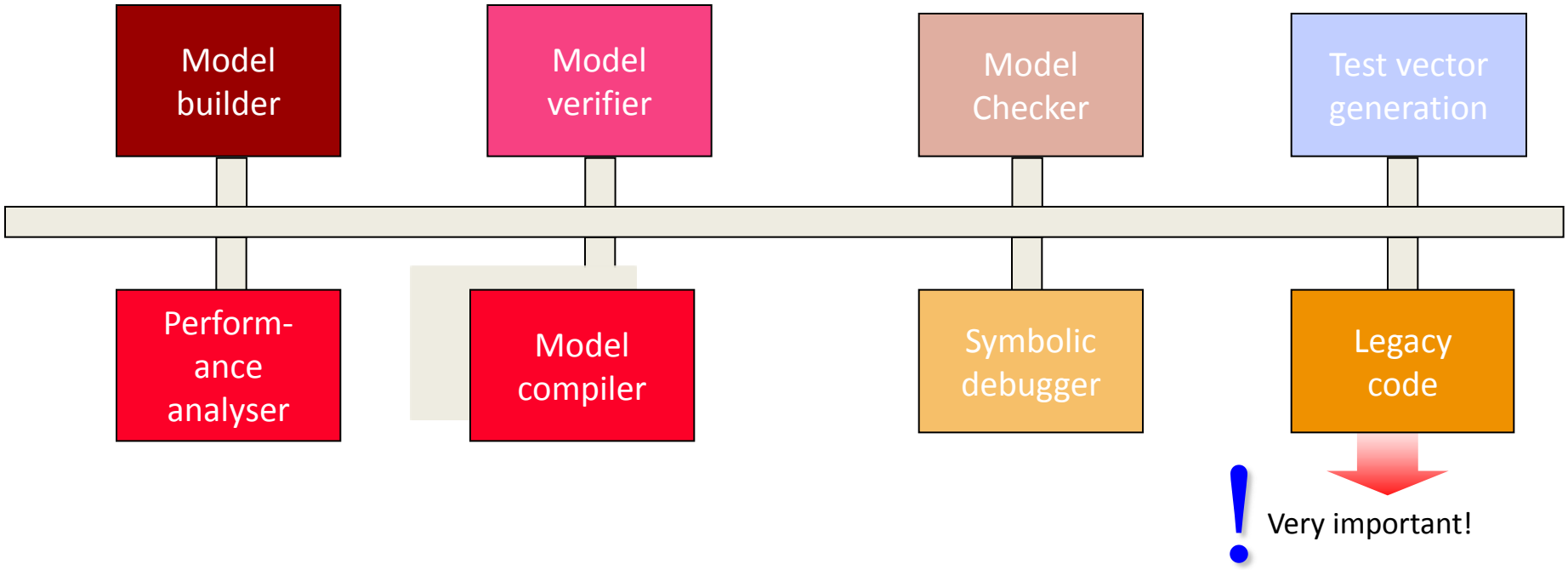
Semantics Backplane

- 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

A prediction for modeling