

# Model-Driven Development: Concepts, Trends and Standardization

Stephen J. Mellor  
Freeter

# A Brief History of Modeling

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

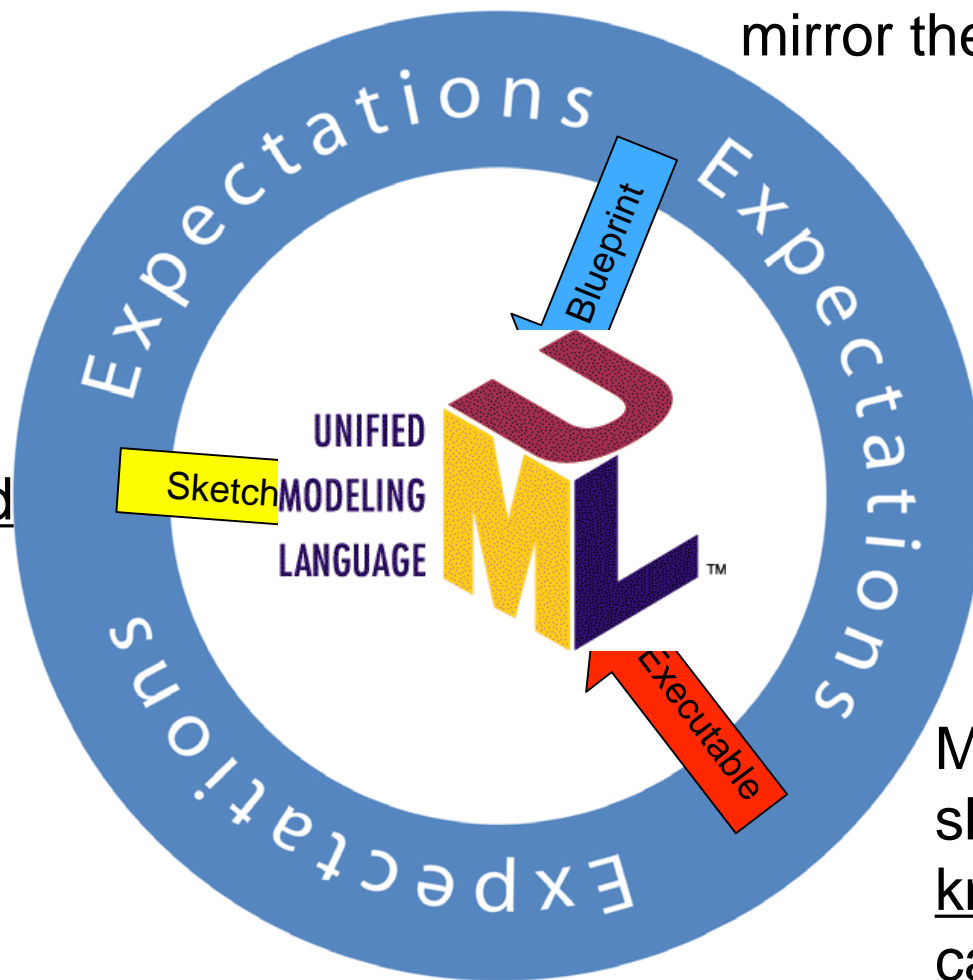
Structured Analysis: De Marco 1981

Structured Design: Yourdon and Constantine 1979

# Differing Expectations

Modeling formalism should mirror the implementation.

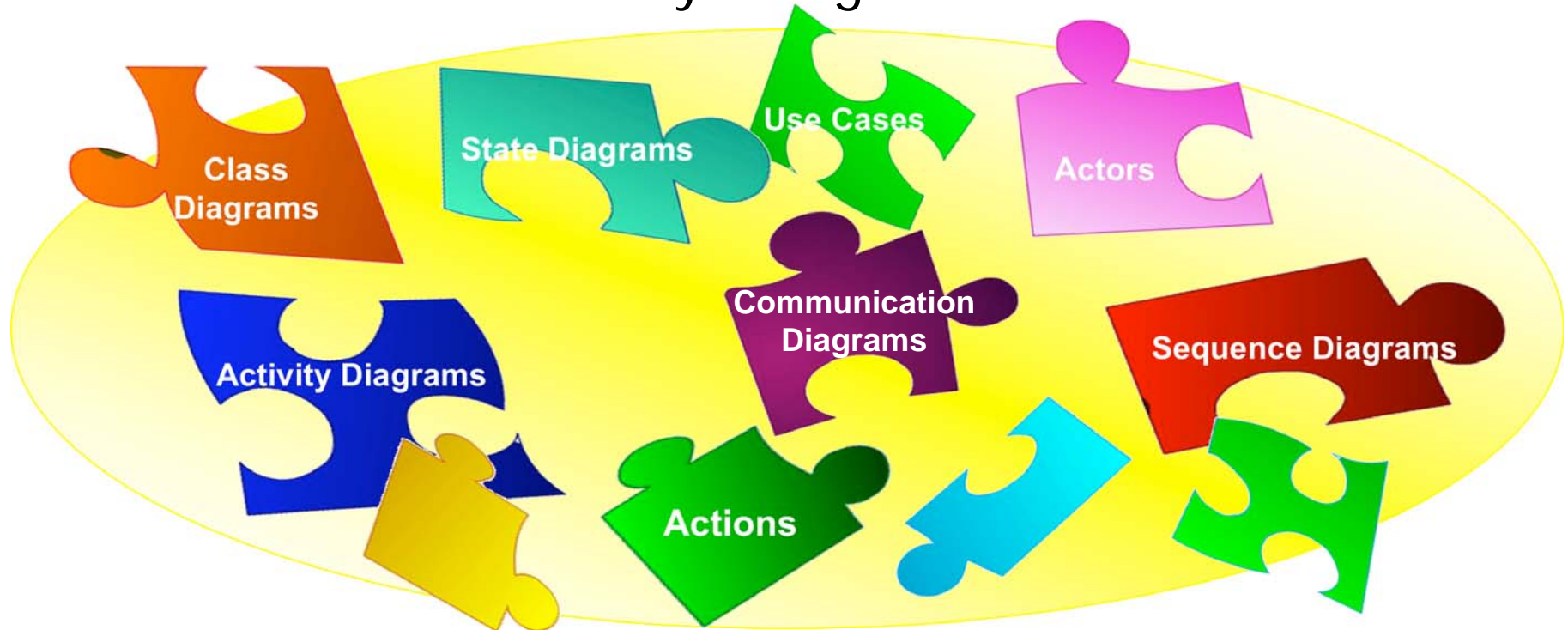
Model should mirror my mind



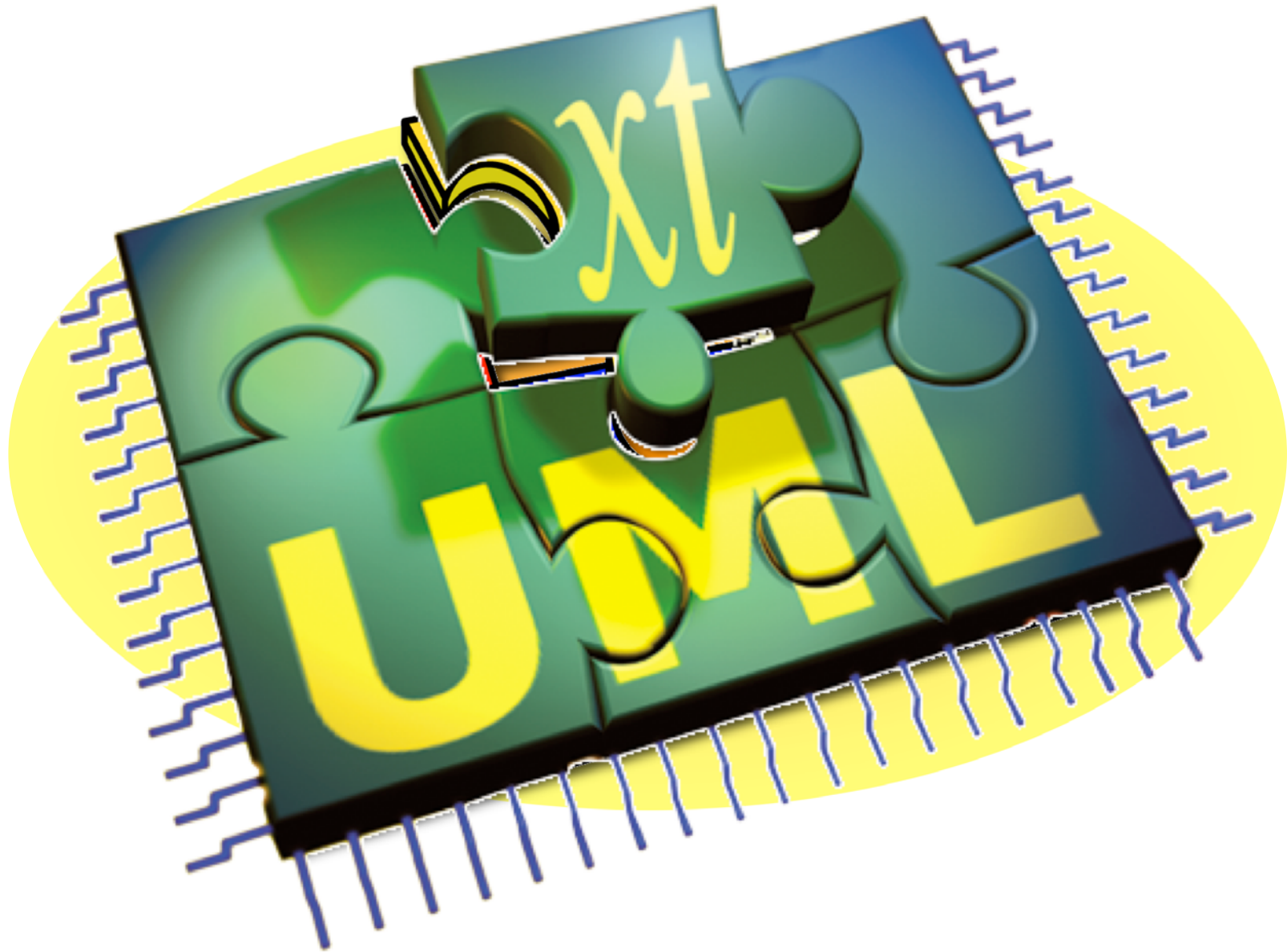
Modeling formalism should mirror the knowledge we're capturing.

# UML Design Decisions

- Can't satisfy all constituencies
- Define each diagram
- Do *not* define how they fit together



# Executable Translatable UML



# Customer X

- Customer X has been using xtUML and BridgePoint on pilot projects.
- They measured memory requirements, processor speed etc. and calculated the cost differential between generated and hand coding
- They used Logiscope to get a measure of code in terms of Maintainability, Testability, Stability, Changeability and Analyzability

# Results from Pilot Projects

*(results from projects where high performance was essential)*

		Performance of Generated Code vs. Hand-written Code
The Pilot Projects	Prop. language	<ul style="list-style-type: none"><li>• 0.8% higher data memory cost</li><li>• 5.2% lower program memory cost</li><li>• worst case 6.4% higher cpu cost</li></ul>
	C++	<ul style="list-style-type: none"><li>• no additional data memory cost (0%)</li><li>• no additional program memory cost (~0%)</li><li>• cpu cost not verified yet</li></ul>
	Firmware (C)	<ul style="list-style-type: none"><li>• no additional data memory cost (0%)</li><li>• 2.4% higher program memory cost</li><li>• 0.7% lower cpu cost</li></ul>

**No time spent on optimizing the generated code**

# Logiscope Code Quality Results

		Functions possible to grade (% of total)			
		Legacy (51%)		Generated (86%)	
Logiscope Quality Report	Maintainability	Excellent 58%	Good 37%	Excellent 63%	Good 28%
	Testability	Excellent 86%	Good 10%	Excellent 74%	Good 14%
	Stability	Excellent 65%	Good 18%	Excellent 82%	Good 18%
	Changeability	Excellent 57%	Good 36%	Excellent 82%	Good 13%
	Analyzability	Excellent 56%	Good 18%	Excellent 88%	Good 7%

- Legacy application has 394 functions
- 193 of these not possible to grade
- Application has 215 functions
- 29 of these not possible to grade



# Ricoh

- Printers, office equipment
- C
- Unknown RTOS
- Renesas and Mitsubishi
- 128k



# Tellabs

- Communications switching (internet/telephony)
- C++
- pSOS, Solaris
- PowerPC (over 1000 controllers)
- >1 MB per controller card



# Saab

- Military systems
- Ada
- Classified
- Classified
- Classified



# Bioanalytical Systems

- Laboratory equipment
- C
- No RTOS
- 8051
- 32k ROM + 6kb RAM



# Ericsson

- Cellular telephony, base stations, wireless
- C, C++
- Various
- ARM
- Various



# St Jude Medical

- Defibrillator, pacemaker
- Assembly, C, C++
- No RTOS
- Zilog Z8 series
- Unknown



# Delphi

- Heating Ventilation Air Conditioning
- C
- No RTOS
- Unknown
- <64k





# Visa

- Credit Card Authorization
- C++
- Solaris
- Sparc
- 4GB



Well over 10,000 Trx/Sec  
(actual number proprietary)



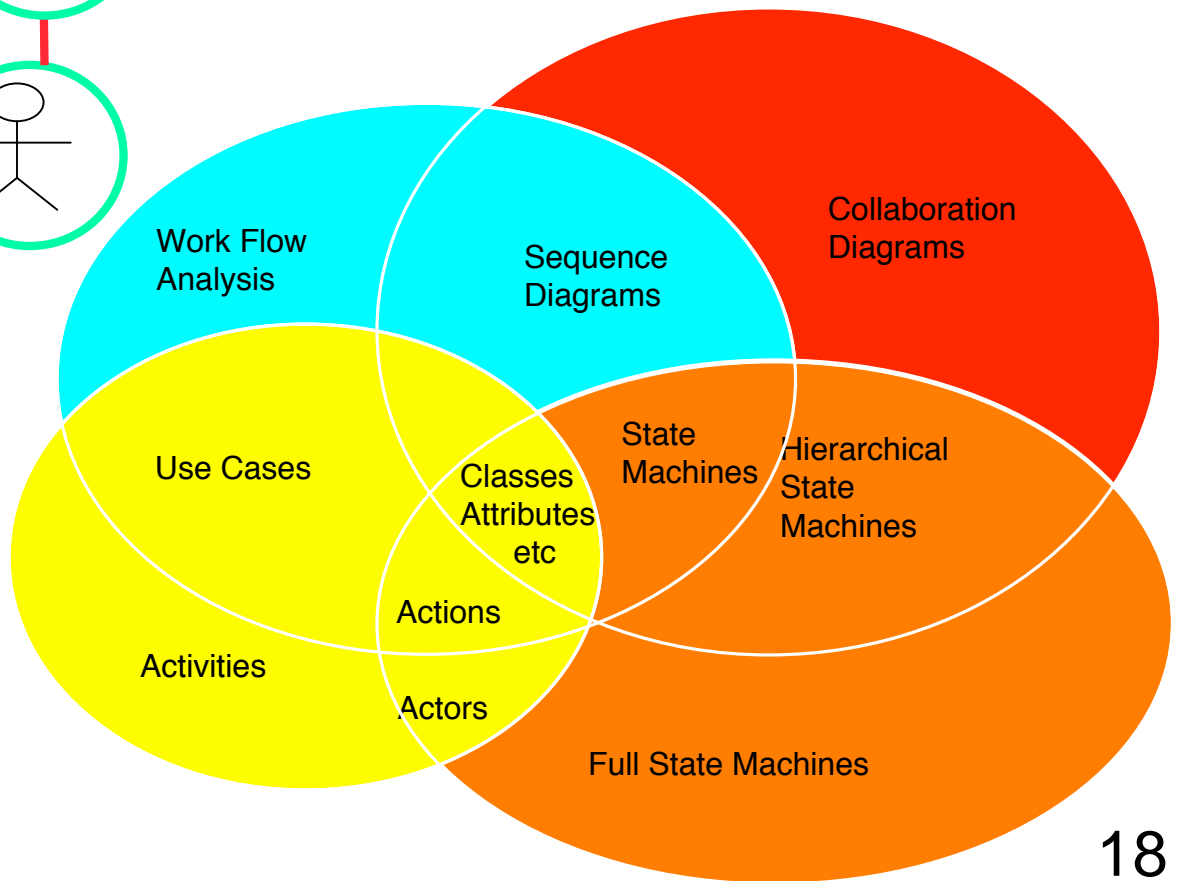
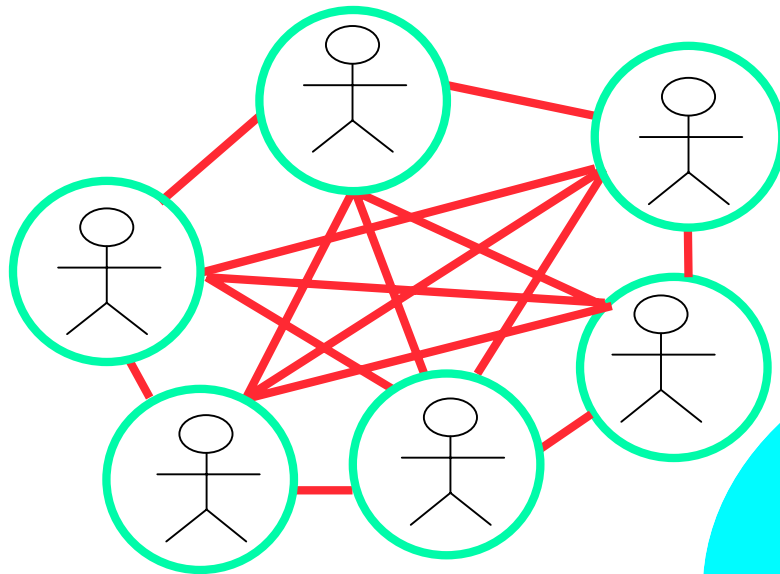
# Orange (nee France Telecom)

- Network Applications
- C++
- Solaris
- Sparc
- 4GB and up



# UML Tools Use Different Subsets of UML

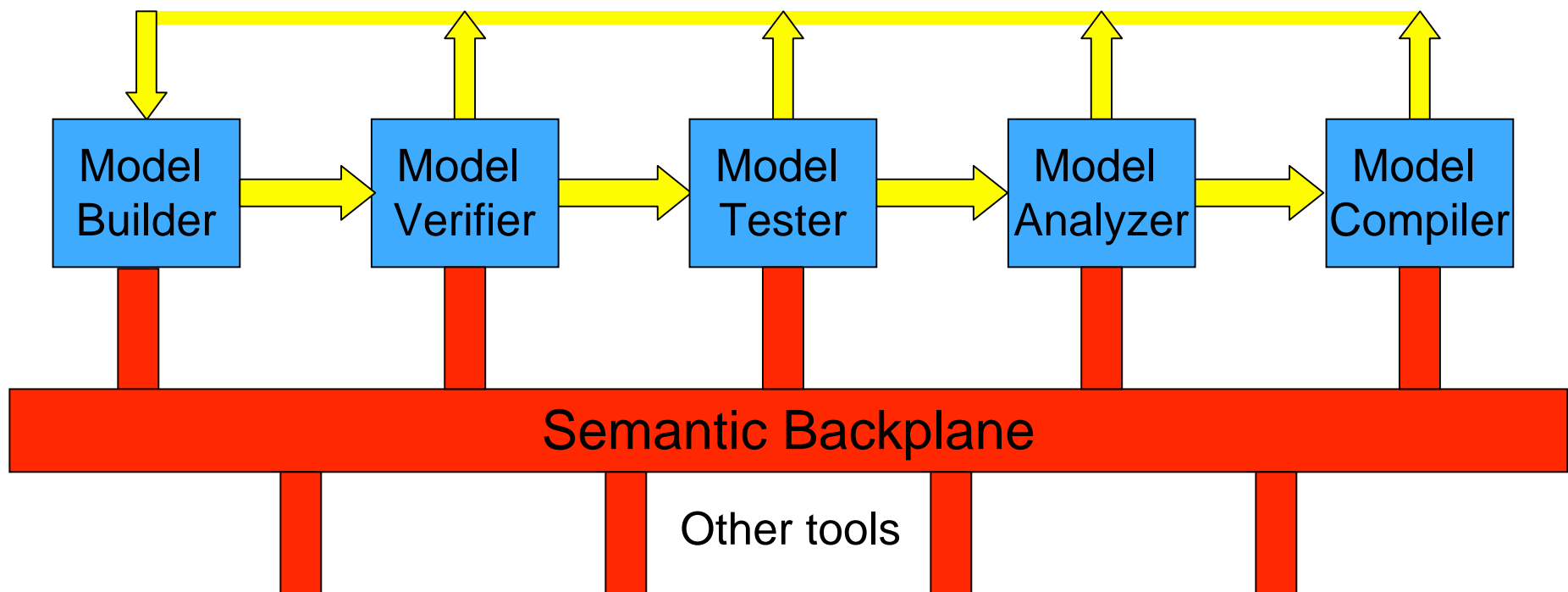
*Meaningful interchange between tools is difficult.*



# Executable UML Foundation

The Executable UML Foundation defines:

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



# When will Execution be Commonplace?

1985: "In three years time..."  
1987: "In three years time..."  
1989: "In three years time..."  
1991: "In three years time..."  
1993: "In three years time..."  
1995: "In three years time..."  
1997: "In three years time..."  
1999: "In three years time..."  
2001: "In three years time..."  
2003: "In three years time..."  
2005: "In three years time..."



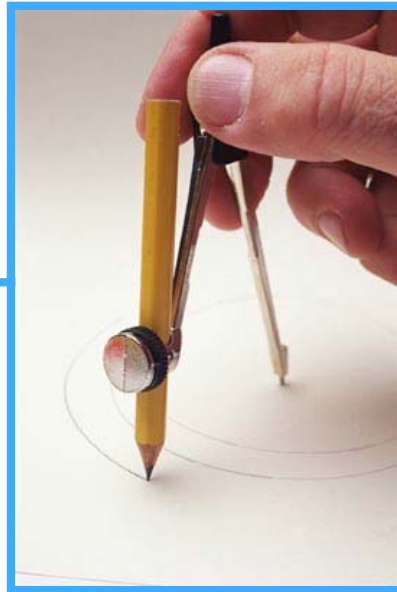
# Why now?

## Knowledge

Individuals  
Projects  
Companies

## Market Usage

Sketchers  
Blueprinters  
Executable  
Modelers



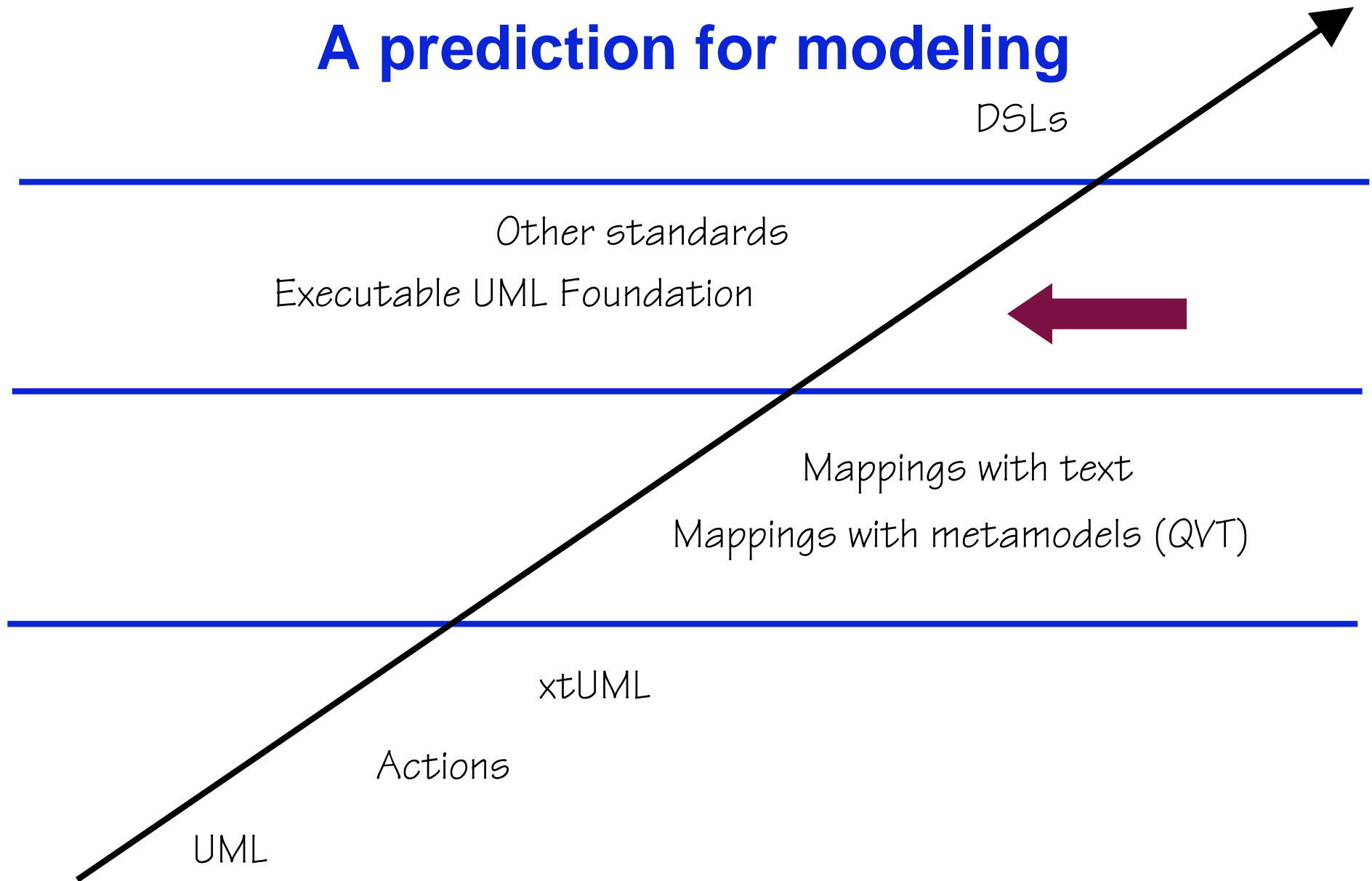
## System Complexity

Programs  
Systems  
Systems of Systems

## Standards

Methods  
UML  
Interchange

# A prediction for modeling

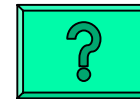
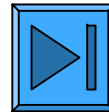
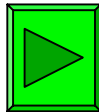
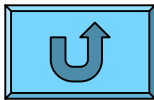


# DSLs

Separating notation enables *domain-specific languages*, graphical languages 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

Questions?

**[StephenMellor@StephenMellor.com](mailto:StephenMellor@StephenMellor.com)**