A Modeler's Journey: from Hostility to Possibility

Jim Logan

Director of Semantic Technologies & Interoperability

No Magic, Inc. (now Dassault Systemès)

Disclaimer

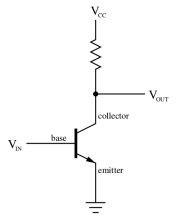
This presentation contains personal opinions that do not necessarily represent those of No Magic or Dassault Systemès.

I Love Modeling!

Photo: Enric Fradera, CC BY 3.0, via Wikimedia Commons

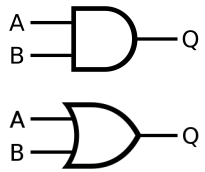
Journey: Concrete to Abstract

Discrete Electronics



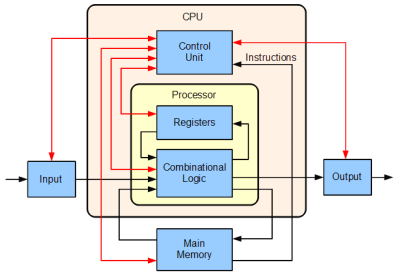
Michael9422, CC BY-SA 3.0, via Wikimedia Commons

Gates



Inductiveload, Public domain, via Wikimedia Commons

CPUs



Lambtron, CC BY-SA 4.0, via Wikimedia Commons

Machine code

MIPS32Add Immediate Instruction

001000	00001	00010	0000000101011110
OP Code	Addr 1	Addr 2	Immediate value

Equivalent mnemonic: **addi** $r1, $r2, 350

en:User:Booyabazooka, CC BY-SA 3.0, via Wikimedia Commons

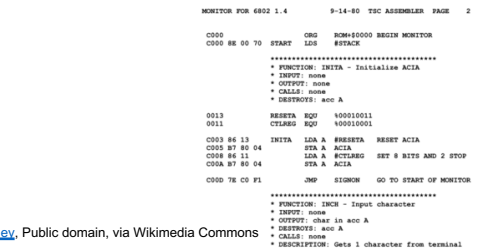
Journey: Managing Complexity

Assembly Language

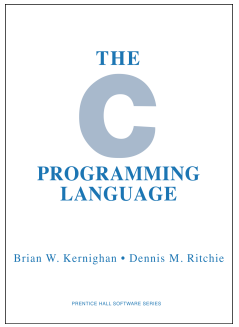
C

Structure Charts

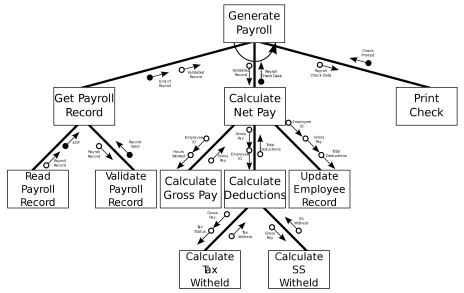
DFDs



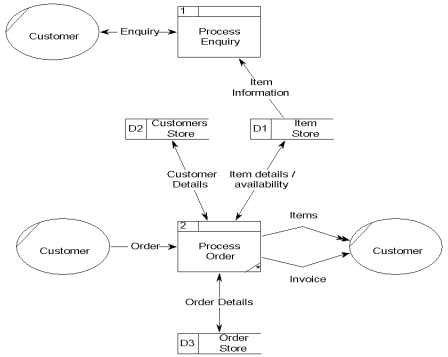
Swtpc6800 en:User:Swtpc6800 Michael Holley, Public domain, via Wikimedia Commons



Julesmazur, Public domain, via Wikimedia Commons



Structured_Chart_Example.jpg: Sandia National Laboratories
derivative work: Pluke, Public domain, via Wikimedia Commons



Turkystuffing, CC BY-SA 3.0, via Wikimedia Commons

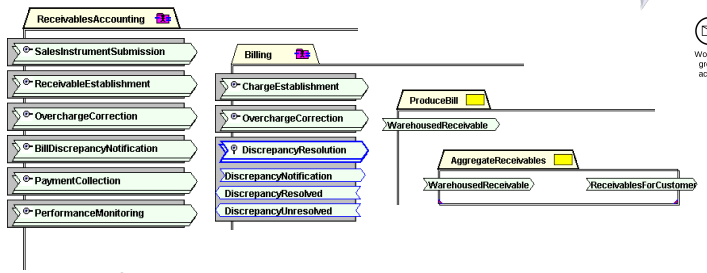
Journey: Analyzing Domains

BPMN

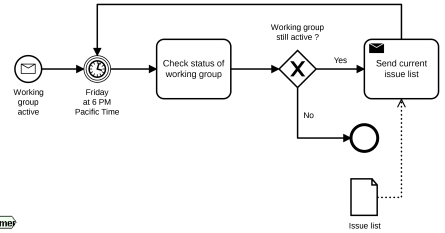
EDOC

UML

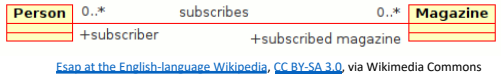
OMT



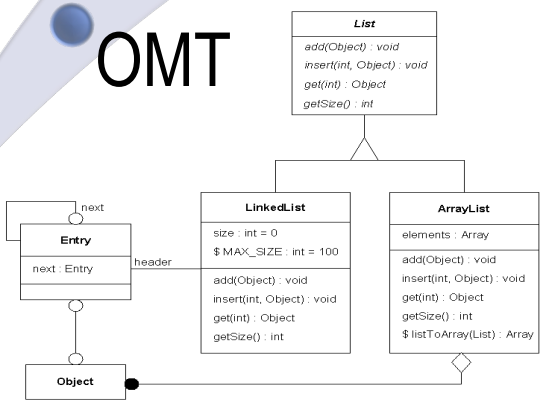
Copyright © 2006 Data Access Technologies, Inc., used with permission



Mikelo Skarabo, CC BY-SA 4.0, via Wikimedia Commons



Esap at the English-language Wikipedia, CC BY-SA 3.0, via Wikimedia Commons



User:AutumnSnow, CC BY-SA 3.0, via Wikimedia Commons

Why Do I Love Modeling?

- Helps me understand the world through the eyes of experts
- Helps me simplify incomprehensible detail
- Reduces rework in documents, schemas, and code
  - Helps with naming and organizing
  - Helps separate domain constraints from implementation constraints
  - No boilerplate code!
- Can generate schemas, and code, e.g.
  - ModelPro
  - AndroMDA
  - xUML
- But…

My Disappointments

- Modeling is not mainstream for software developers
- Most software developers are unmotivated to learn how to model
- Modeling leads to simplifications that seem obvious (but aren't)
- Most models focus on technical details and data structures
- Most models useful only for one system
- Some models are inert and not updated over time
- Some models are polluted with transformation directives
- What's worse than disappointment?

Hostility

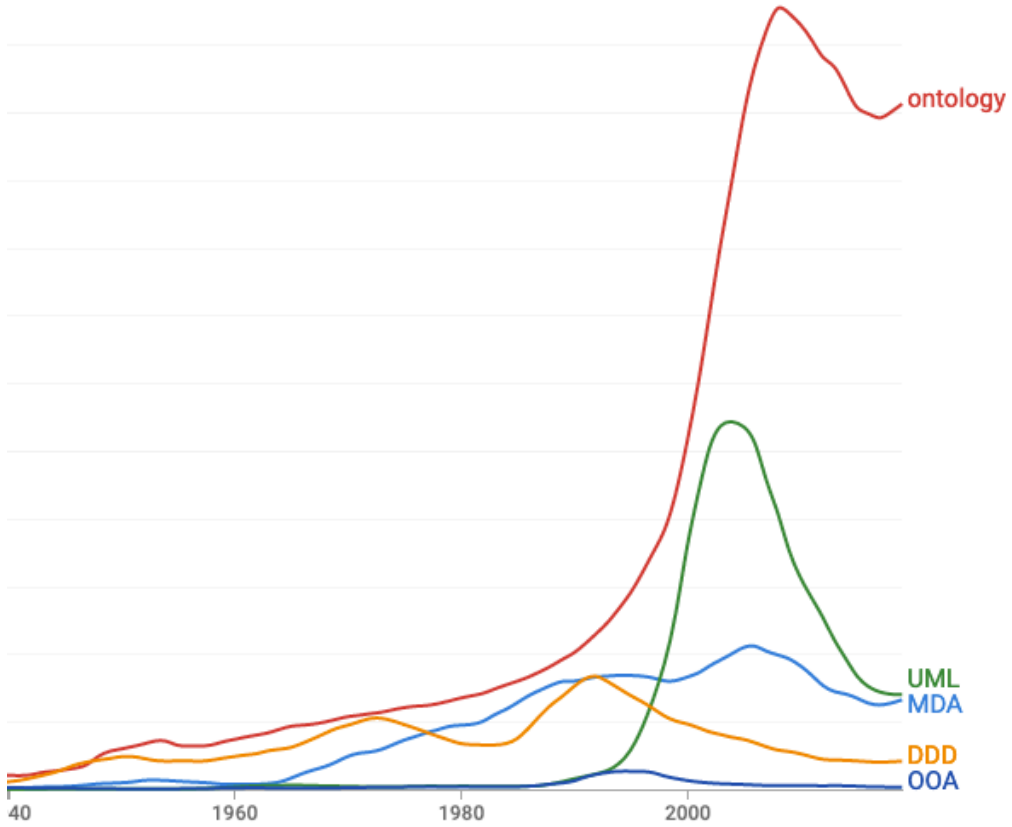
- Unfashionable
- Kind of tried it once, but it didn’t work
- Snide comments at developer conferences
- Not worth the effort
- Hard to understand
- Too “waterfall” for Extreme Programming / Agile (although orthogonal concern)
- Tools too expensive
- The code is the model
- Models are just sketches
- Sometimes they had proud ignorance of analysis and design principles
- Yet, some are not so hostile…

Possibility

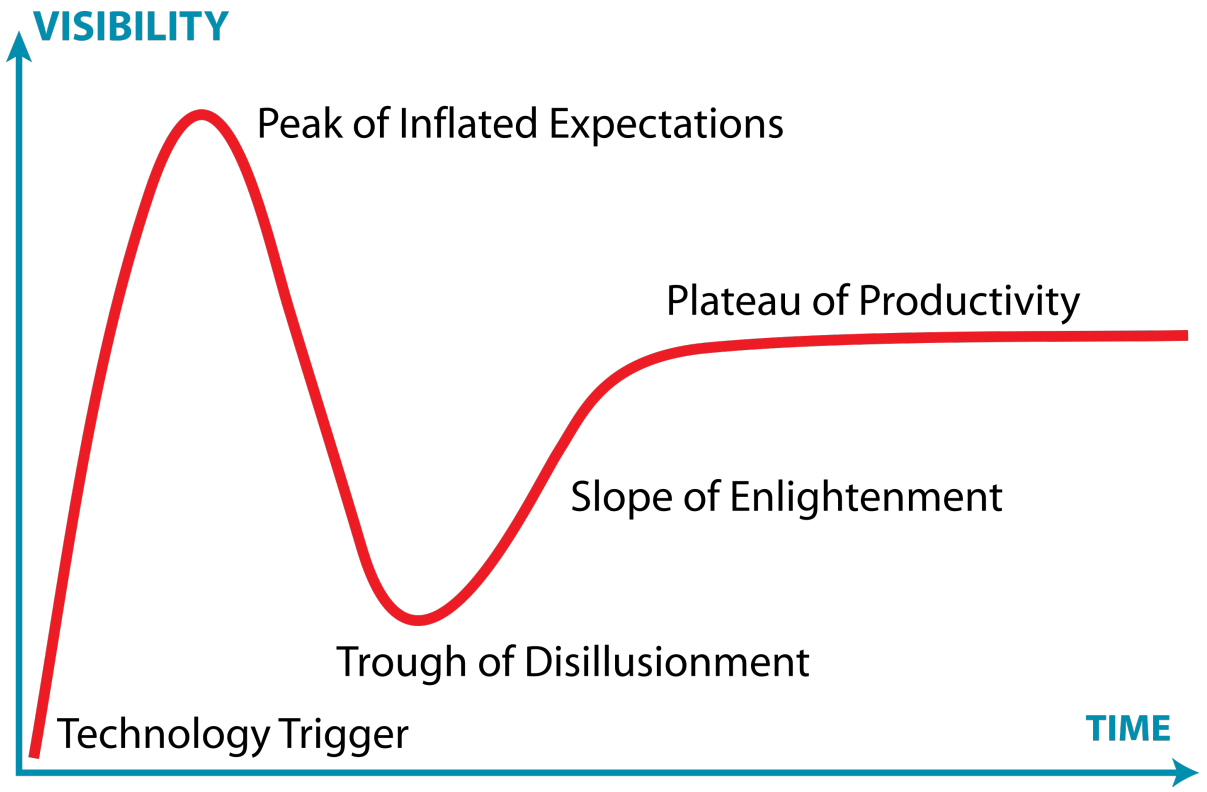
- Uptick in ontologies, semantics, and conceptual modeling
  - Huge demand for semantics from system engineering (SysMLv2)
  - (Lots to learn from SEs!)
  - Industrial Ontology Foundry (IOF)
  - Financial Industry Business Ontology (FIBO)
  - ISO 21838 (Information technology — Top-level ontologies)
    - Part 2: Basic Formal Ontology (BFO)
    - Part 3: Descriptive ontology for linguistic and cognitive engineering (DOLCE)
  - Open Biomedical Ontology (OBO) Foundry
  - OntoUML / UFO / gUFO
  - Conceptual ontology consultancies, such as The Shed Group
- Standards for simulation and execution
  - fUML, Alf, PSSM – all we need for standardized xUML!
- Ontologies and Executable UML are not new!

Everything Old Is New Again

- Ontology
- Object-Oriented Analysis
- Executable UML
- Domain Driven Development



Source: Google Books Ngram Viewer, https://books.google.com/ngrams



Source: Jeremykemp at English Wikipedia, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=10547051

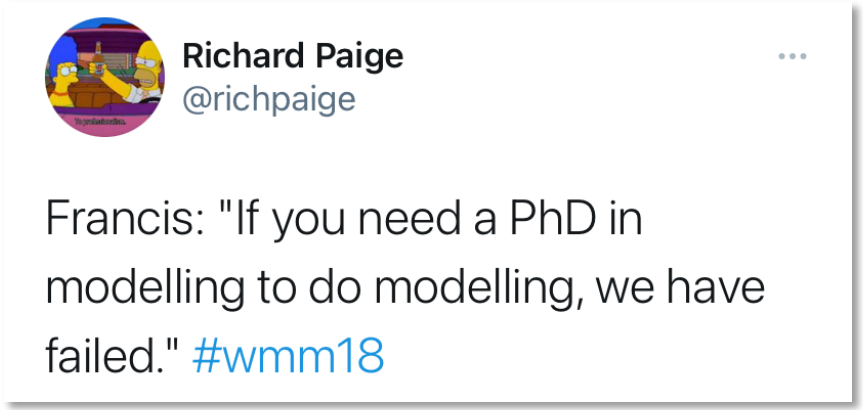
Mind The Gap!

- The state of practice lags the state of the art
  - Insufficient empirical evidence
  - Rollout left to industry
  - Chicken and egg tool problem in the marketplace
- Incumbent work force
  - Comfortable working in a particular way
  - Prefer doing over learning
  - Getting code to work creates a “high”
  - Cannot afford a loss of productivity
  - Prefers evolutionary changes, not revolutionary changes
  - Needs fashionable skills on their resumes

Image: GIANLU, CC0, via Wikimedia Commons

Bridge The Gap!

- Convincing and motivating case studies
- Progressive benefits (an easy to learn 20% yields 80% impact)
- Widely accepted, respected, understood, and practiced methodology
- Enjoyable, “gamified” experiences
- Productive tools for development and interoperability
- It might start with reusable, conceptual ontologies…

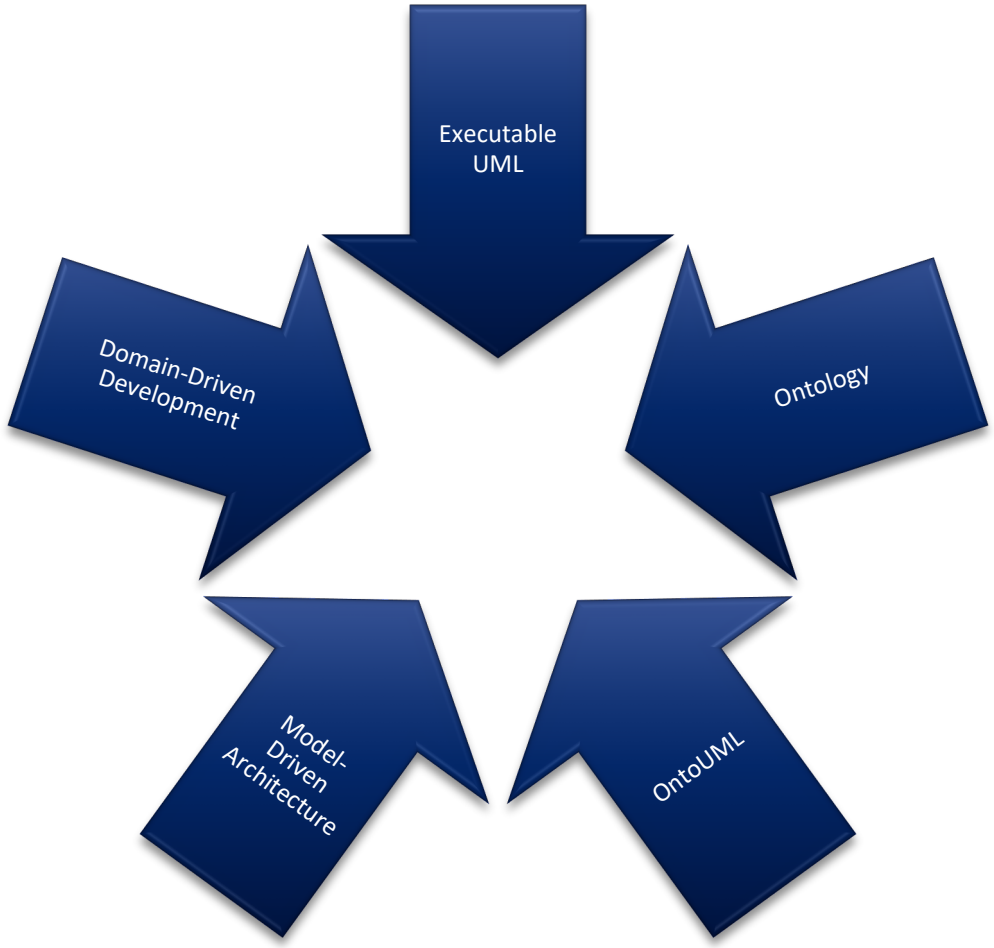


The Existing Landscape (Across the Gap)

- **OntoUML / UFO /gUFO**
  - Has empirical evidence
  - Well-foundedness prevent mistakes
  - Tools since at least the 2010's; new tool in development
- **Basic Formal Ontology (BFO)**
  - Hundreds of ontologies in use
  - Many more in development at the Industrial Ontology Foundry (IOF)
  - OBO Foundry and tools since the 2000's
- **SysML**
  - Huge demand from systems engineers
  - Lots to learn from system engineers
  - SysMLv2 on the way
  - Tools since the 2010's
- **xUML**
  - Shlaer/Mellor tools since the 1980's
  - xUML tools since the 2000's
  - fUML / Alf tools since the 2010's

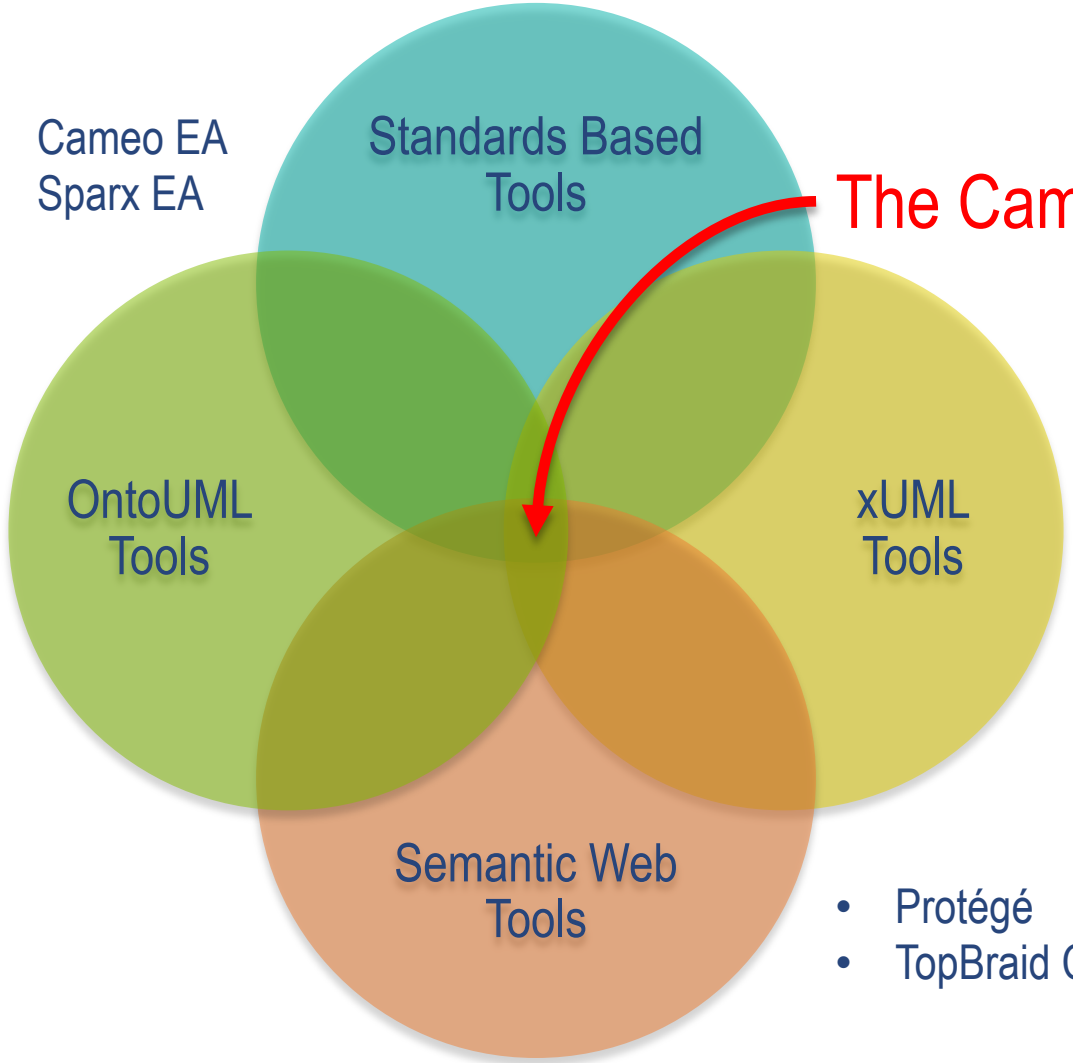
Emerging: explicit modeling of the *problem* domain!

A Vision of the Future



- Per-domain reference ontology
  - Use an upper ontology (e.g., BFO / UFO / DOLCE)
  - Model the *reality* of a domain
  - Reify qualities (to allow multiple value spaces)
  - Reify material relations (to uncover truth-makers)
- Per-solution analysis and design
  - Reduce scope and adjust per information demand
  - Use a state machine and events per concept
  - Use an action language to respond to an event
  - Simulate
  - Generate code and schemas
  - Semantically interoperate

Changing the Modeling Landscape



- Cameo EA
- Sparx EA

The Cameo Concept Modeler

- OLED
- Menthor
- (Name TBD)

- fUML / Alf reference implementation
- Cameo Simulation Toolkit
- Moka / Papyrus UML
- Pópulo
- xtUML (Bridgepoint)

- Protégé
- TopBraid Composer

In Summary

- I love modeling, despite some disappointments and hostilities
- Let's learn from those disappointments
- I see a widening gap between state of practice and state of the art
- However, there has been an uptick in ontology modeling
- Let's focus more on the *problem domain*
- I have a tool and a vision that could make modeling mainstream
- "If you want to go fast, go alone; but if you want to go far, go together." – *African proverb*
- Please join me on my modeler's journey
- And **mind the gap**!

The End