Malina

SOFTWARE CORP.

The Should Be, Could Be, and (Likely) Will Be of Model-Based Engineering

Bran Selic

Malina Software Corp.
CANADA

selic@acm.org

ALL BLACKS®

Deep Thoughts on Prediction

- "Prediction is very hard, especially about the future"
  [Yogi Berra, baseball player]

- "It would appear that we have reached the limits of what is possible to achieve with computer technology, although one should be careful with such statements, as they tend to sound pretty silly in 5 years."
  [John von Neumann, computer pioneer (cca. 1949)]

And, One More...

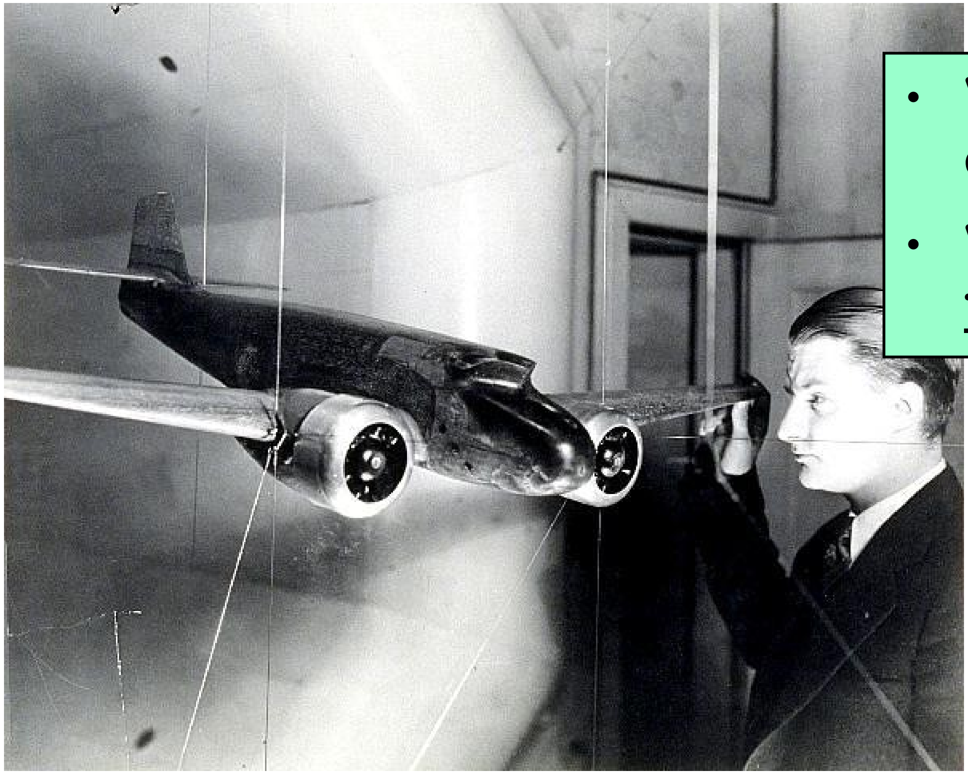
“The best way to predict the future is to invent it”,
[Alan Kay, computer scientist]

But, what if we invented it, and no one came...?

What Modeling Is

Engineering Models

- **ENGINEERING MODEL:** ***A selective representation of some system that captures accurately and concisely all of its essential properties of interest for a given set of concerns (viewpoint)***



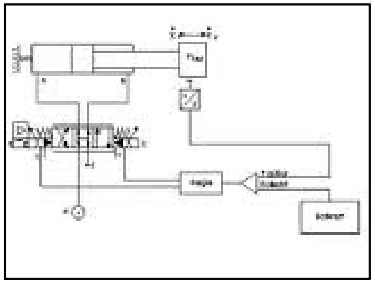
- We don't see everything at once
- What we do see is adjusted to human comprehension

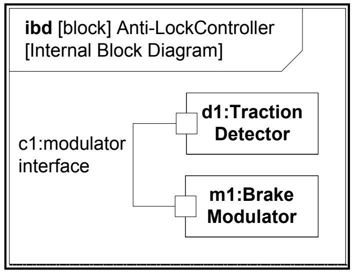
Engineering Models and Their Purposes

- Descriptive models
  - To help us understand (reason about) complex systems
  - To communicate understanding and design intent to others
  - To predict the interesting characteristics of systems
- Prescriptive models
  - To specify systems
  - …and, in special cases, to implement systems

Models, Models Everywhere

- …including on computers!

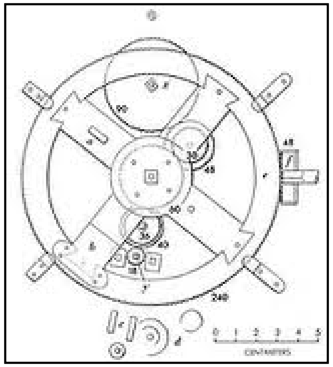


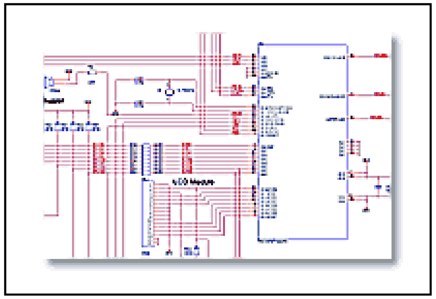

ibd [block] Anti-LockController [Internal Block Diagram]

c1:modulator interface

d1:Traction Detector

m1:Brake Modulator





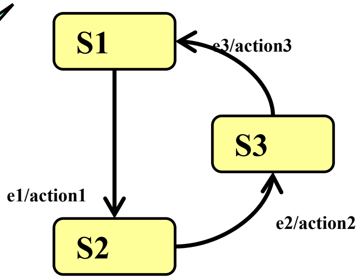
The flexibility and automation potential of computers has provided a crucial boost to the effectiveness of engineering models

The World of Computer-Based Models

- Based on two time-proven notions:

(1) ABSTRACTION

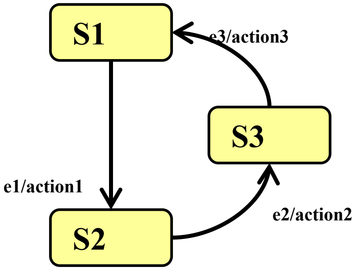
Realm of __modeling__ __languages__



```
switch (state) {
   case'1:action1;
          newState('2');
          break;
   case'2:action2;
          newState('3');
          break;
   case'3:action3;
          newState('1');
          break;}
```

(2) AUTOMATION

Realm of __tools__ and __model__ __transforms__



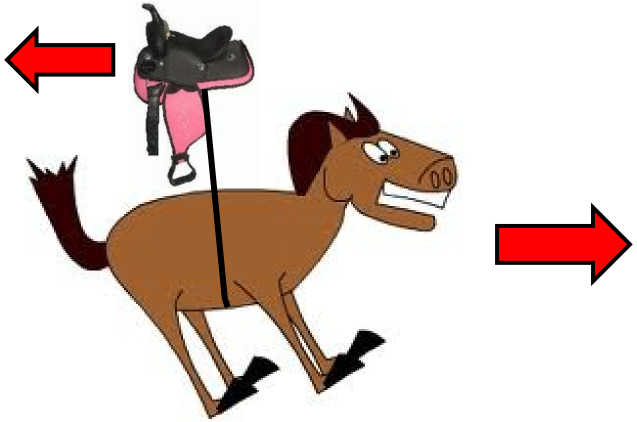
```
switch (stat
   case'1:ac
          newState('2');
          break;
   case'2:action2;
          newState('3');
          break;
   case'3:action3;
          newState('1');
          break;}
```

But, is this really "model-driven"?

On Naming…A Case of Inverted Priority

SPORTS: Referee-driven competitions ??

- **Engineering is driven by models in the same way that a saddle drives a horse**



Modeling is neither a design paradigm nor a methodology!

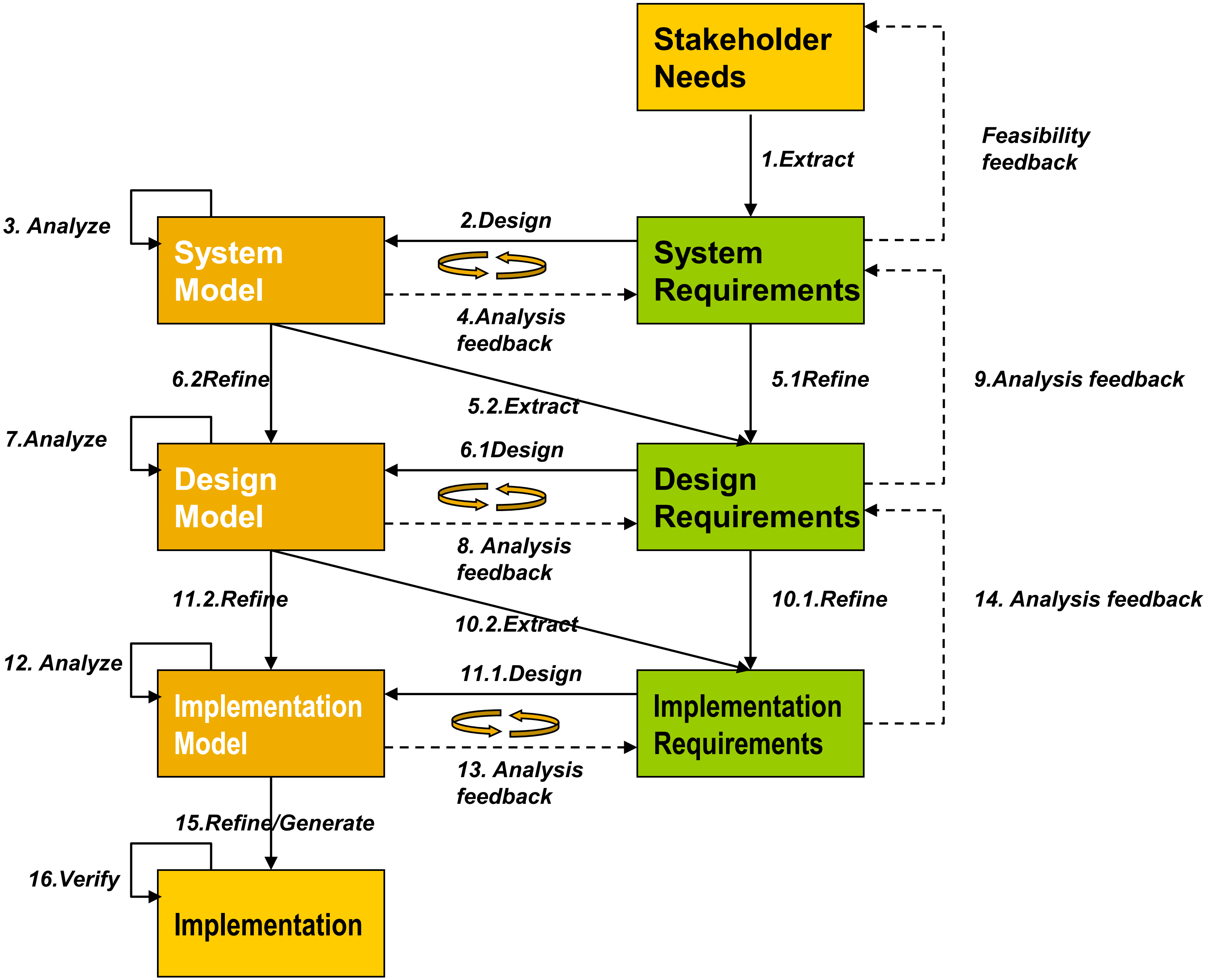
In the Year 2020: What Modeling Should Be

In the Year 2020…

- We can expect much more sophisticated software instensive systems than today:
  - More complex functionality
  - Systems that are highly integrated with their environment (environment aware, reactive, adaptive)
  - Increased demands on:
    - Safety
    - Availability
    - Reliability
    - Computational power

There is no doubt that traditional methods and computing technologies will not be able to cop

"Future" Engineering Processes


Stakeholder Needs

1.Extract

Feasibility feedback

3. Analyze

System Model

2.Design

System Requirements

4.Analysis feedback

6.2Refine

5.2.Extract

5.1Refine

9.Analysis feedback

7.Analyze

Design Model

6.1Design

Design Requirements

8. Analysis feedback

11.2.Refine

10.2.Extract

10.1.Refine

14. Analysis feedback

12. Analyze

Implementation Model

11.1.Design

Implementation Requirements

13. Analysis feedback

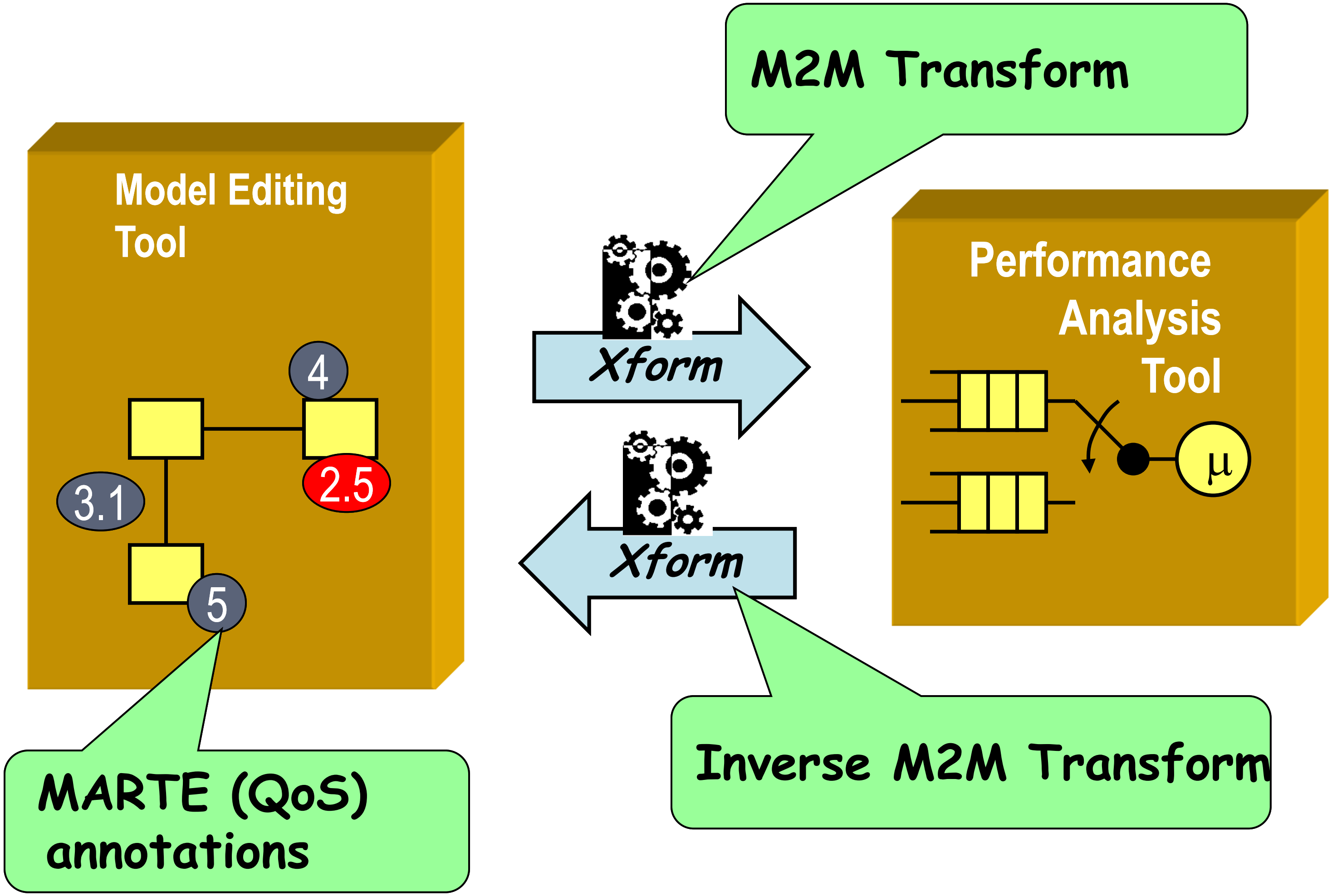
15.Refine/Generate

16.Verify

Implementation

Automated Model Analysis Example

- Transform a model into a queueing network model for predictive analysis

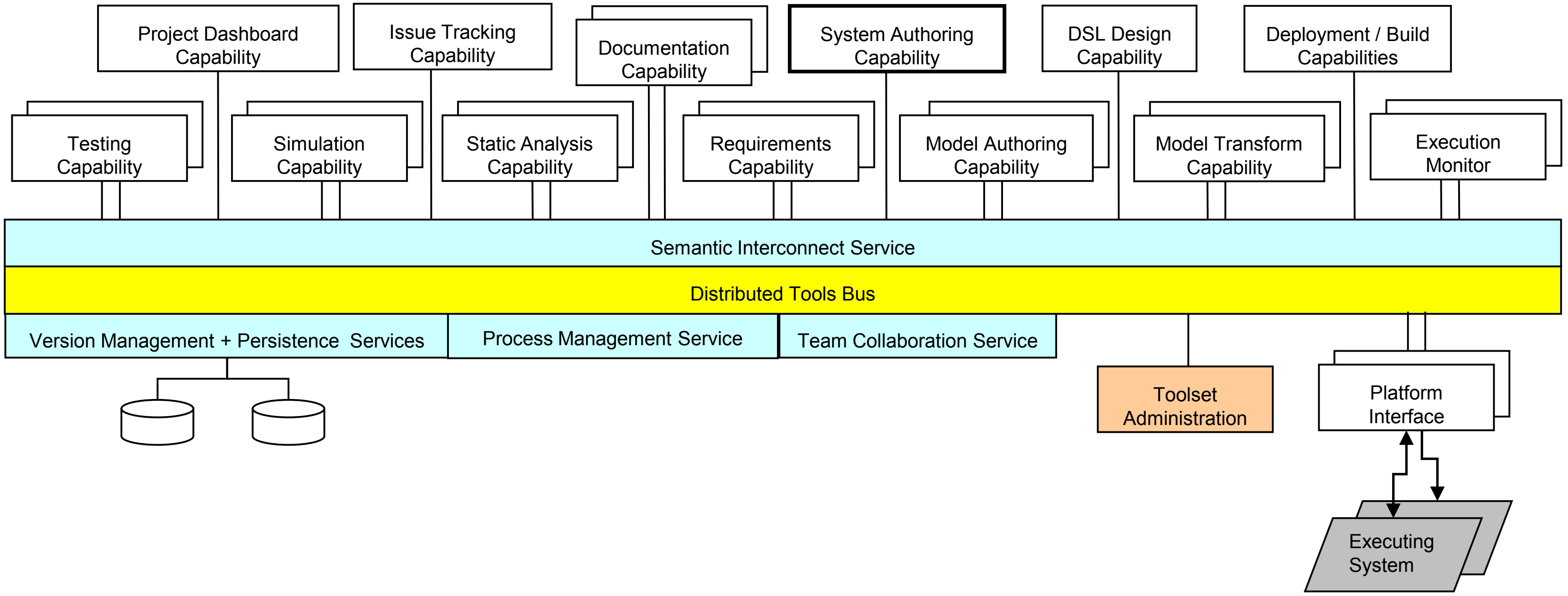


The Role and Nature of Tools

- **Tools are responsible for providing <u>automation</u>**
- **We will have a new generation of tools that are:**
  - Highly usable
  - Highly adaptable
  - Highly specialized (capabilities versus tool packages)
  - Eminently flexible (e.g., adjustable degree of formality, execution of incomplete models)
  - Interworkable (through standards)

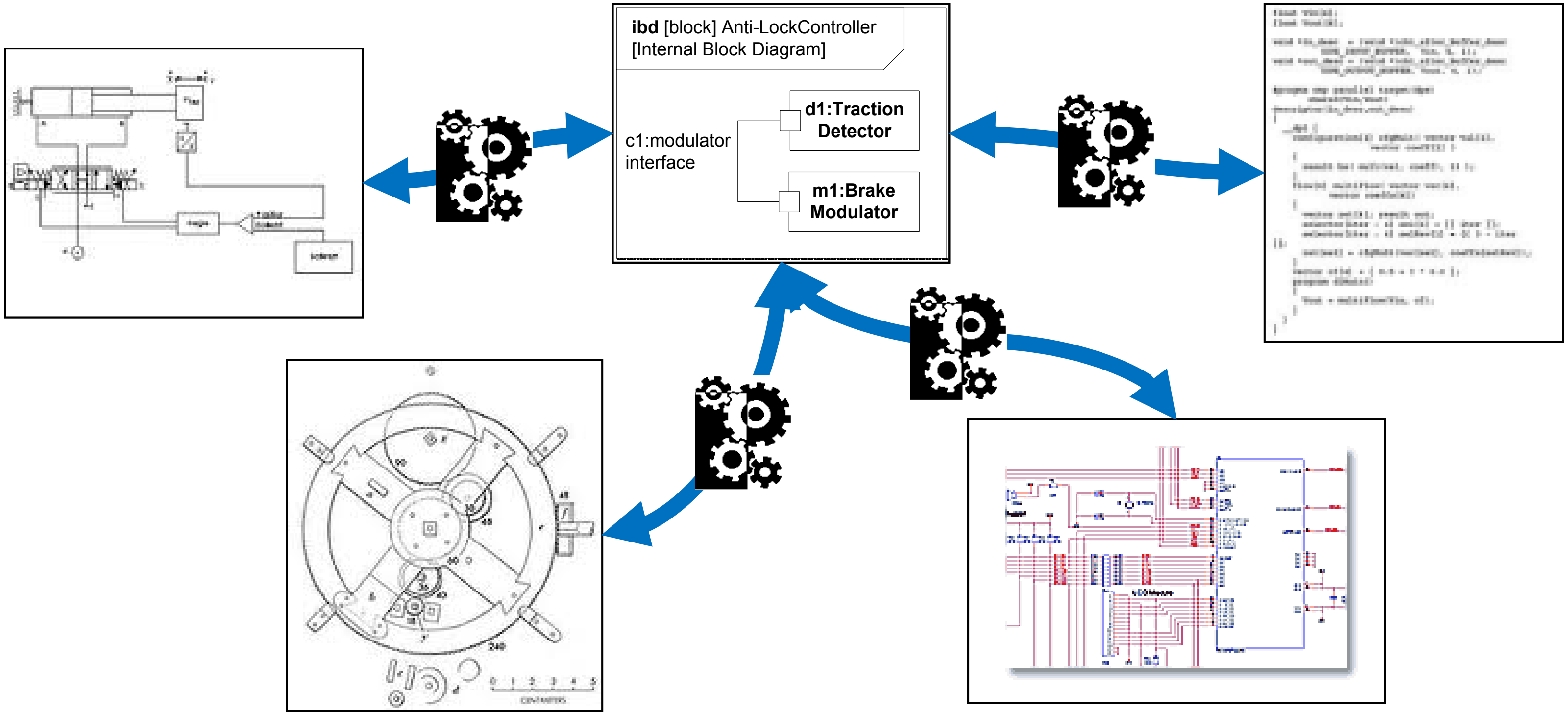
Tools Landscape

- **Moving away from isolated point tools to <u>tool environments</u> with pluggable tool capabilities**
  - With strong collaborative development and project management features
  - Support for semantically-aware model interchange


Project Dashboard Capability
Issue Tracking Capability
Documentation Capability
System Authoring Capability
DSL Design Capability
Deployment / Build Capabilities
Testing Capability
Simulation Capability
Static Analysis Capability
Requirements Capability
Model Authoring Capability
Model Transform Capability
Execution Monitor
Semantic Interconnect Service
Distributed Tools Bus
Version Management + Persistence Services
Process Management Service
Team Collaboration Service
Toolset Administration
Platform Interface
Executing System

Model Integration – Systems of Models

- **Connected via model transforms and various traceability mechanisms**



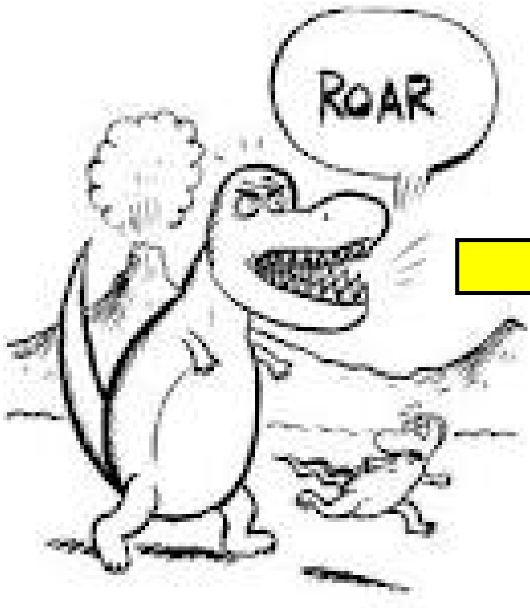
ibd [block] Anti-LockController [Internal Block Diagram]

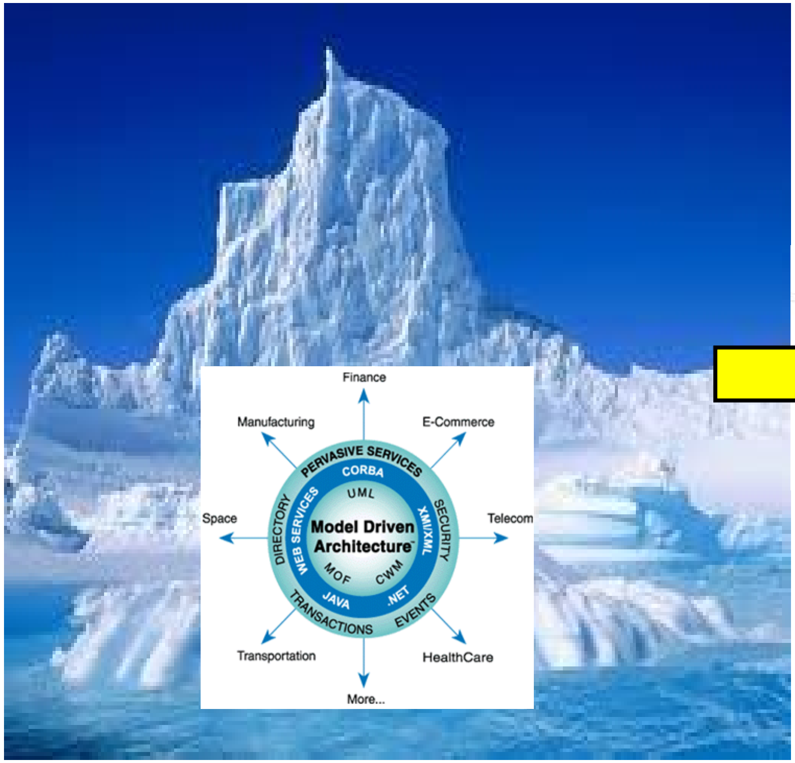
c1:modulator interface

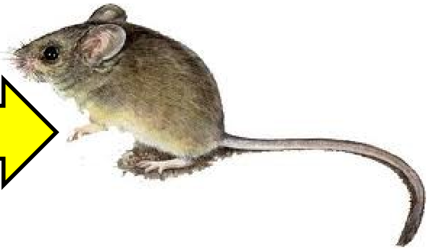
d1:Traction Detector

m1:Brake Modulator

What Modeling Could Be (in the Year 2020)







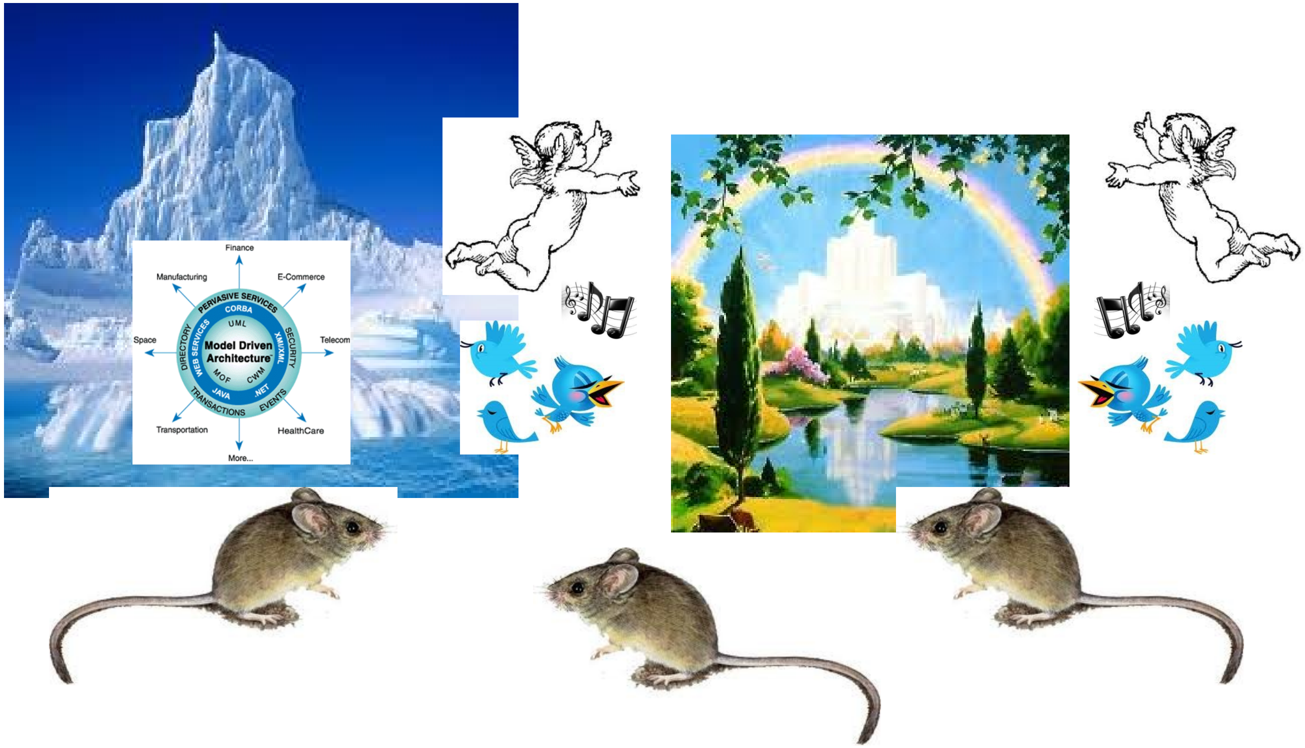
The Case of AI

- 1956 – Field officially founded at Dartmouth College Conference (Minsky, McCarthy, Newell, Simon)
  - Pattern recognition, chess playing, robotics, speech, etc.
- 1960's – Intense government funding (e.g., US DoD)
  - New AI labs being established everywhere
  - "…machines will be capable, within 20 years, of doing any work a man can do" [Herbert Simon]
  - "…within a generation…the problem of creating 'artificial intelligence' will be substantially solved" [Marvin Minsky]
- 1974 – US and UK governments cut off funding to AI research ["AI winter"]
- 1980's – expert systems
- 1990's – "behind the scenes" successes (data mining, medical diagnosis, etc.)
- 2011 – IBM Watson wins Jeopardy! game show

Analysis of AI Example

- **Causes:**
  - Overpromise and underestimation of technical challenges
  - Limitations of computing hardware
- **Led to:**
  - Disappointment
  - Research funding shifting to other more credible domains
- **Gradual recovery due to:**
  - Pig-headedness of a relatively small number of enthusiasts
  - Faster, cheaper, more powerful hardware
  - Low key pragmatic approach (application to concrete industrially relevant issues, avoidance of messianic marketing) ⇒ indisputable and significant success stories
  - Evolution of a greater theoretical underpinning

What Modeling is Likely to Be (and what we can do about it)



Most Likely Scenario

- **Following a relatively mild and brief "MDD Winter"**
  - Many contributing factors: economic, cultural, practical
  - But, mild because
  1. It is the only pragmatic solution
  2. Industry is waking to the potential with some notable published successes
  3. Pig-headed enthusiasts
- **We will see the emergence of a reliable technical/engineering discipline**
  - A proper theoretical framework
  - New and better modeling languages and model transformations
  - Useful and highly usable tools with high degrees of automation

What is to be Done

- **Key research areas**
  - Modeling language design and theory
  - Tools, tools, and tools
  - Human factors
  - Model transforms
  - Model analysis methods
- **Advice to researchers:**
  - Recognize and build on the good ideas of others
  - Avoid starting your own religion, consolidate
  - Work with industry
- **And, drop the useless "model-driven" label that merely serves to foster useless theological sectarianism**

More Words of Wisdom…

“So that we may say the door is now opened, for the first time, to a new method fraught with numerous and wonderful results, which in future years will command the attention of other minds.”
[Galileo Galilei, 16th century]

“What is not fully understood is not possessed.”
[Johann Wolfgang Goethe, German philosopher]