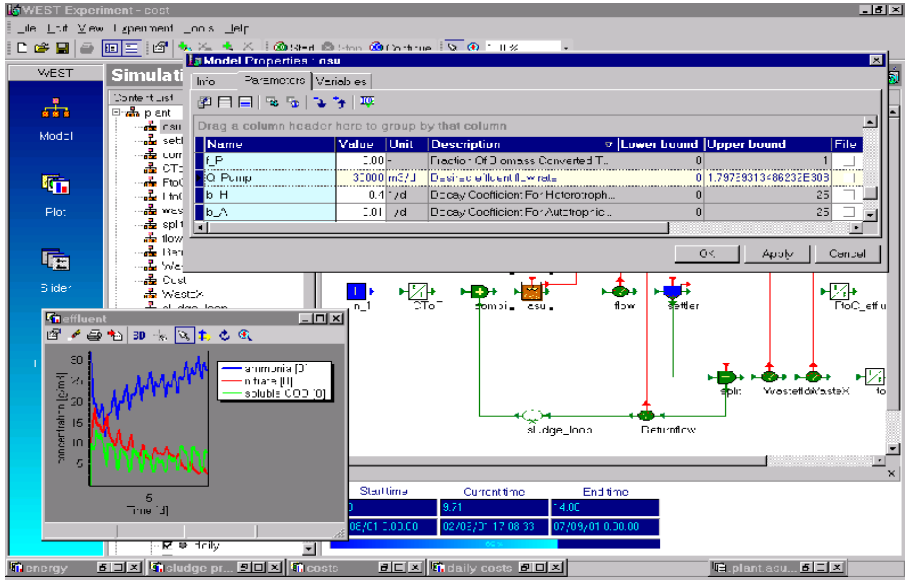
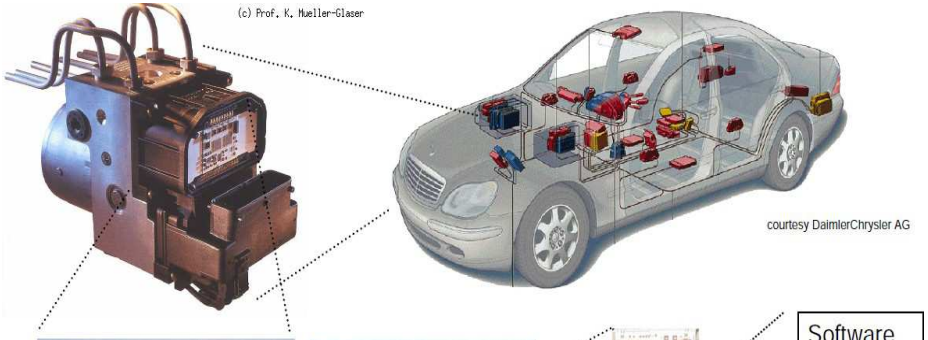
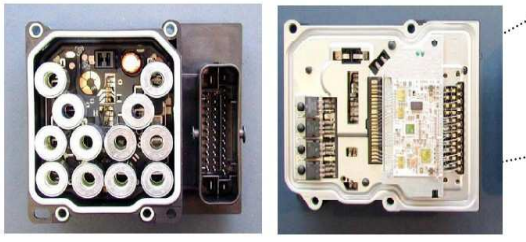
Software Intensive Systems: Dealing with Complexity

Hans Vangheluwe



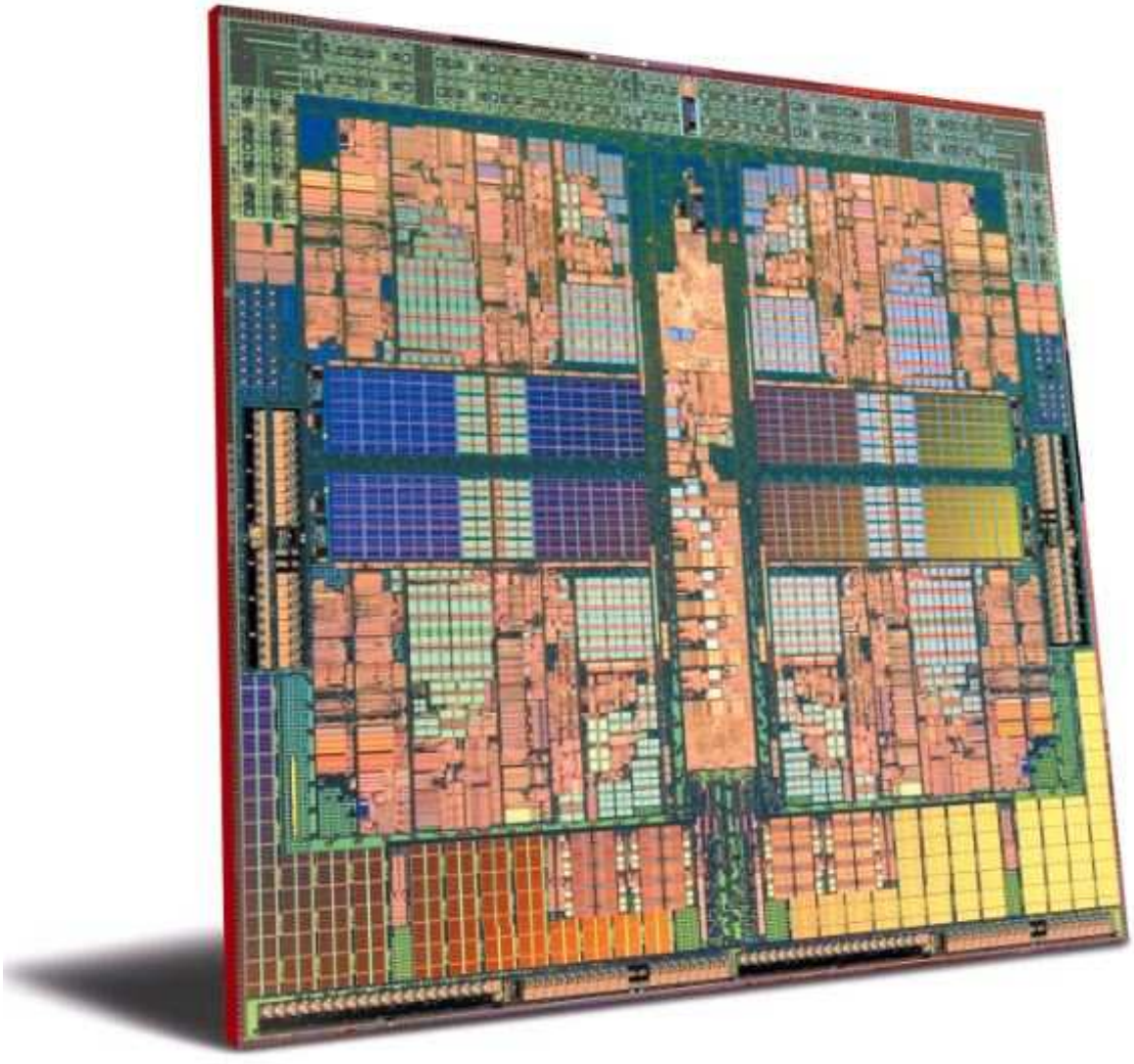
Google™

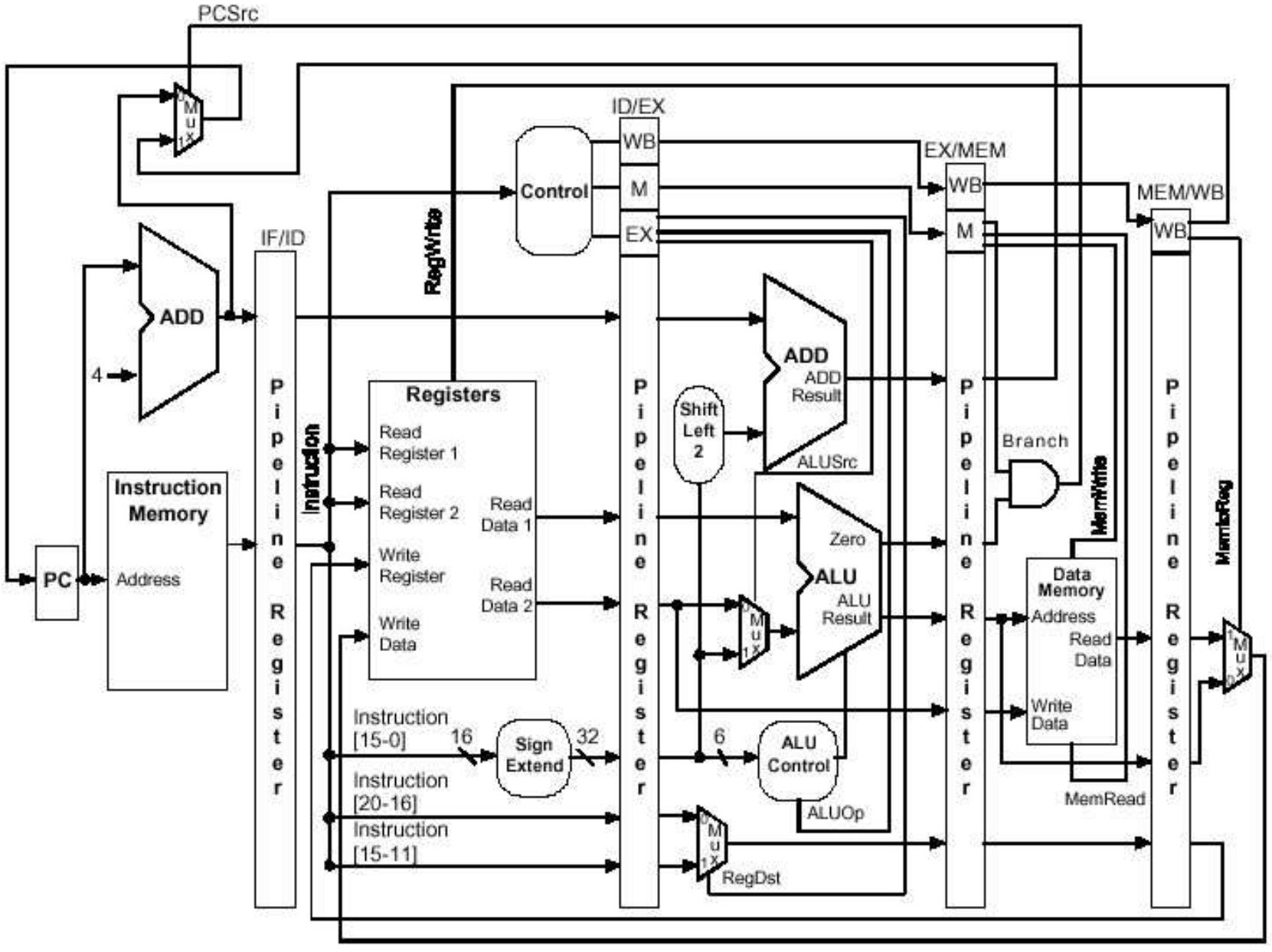


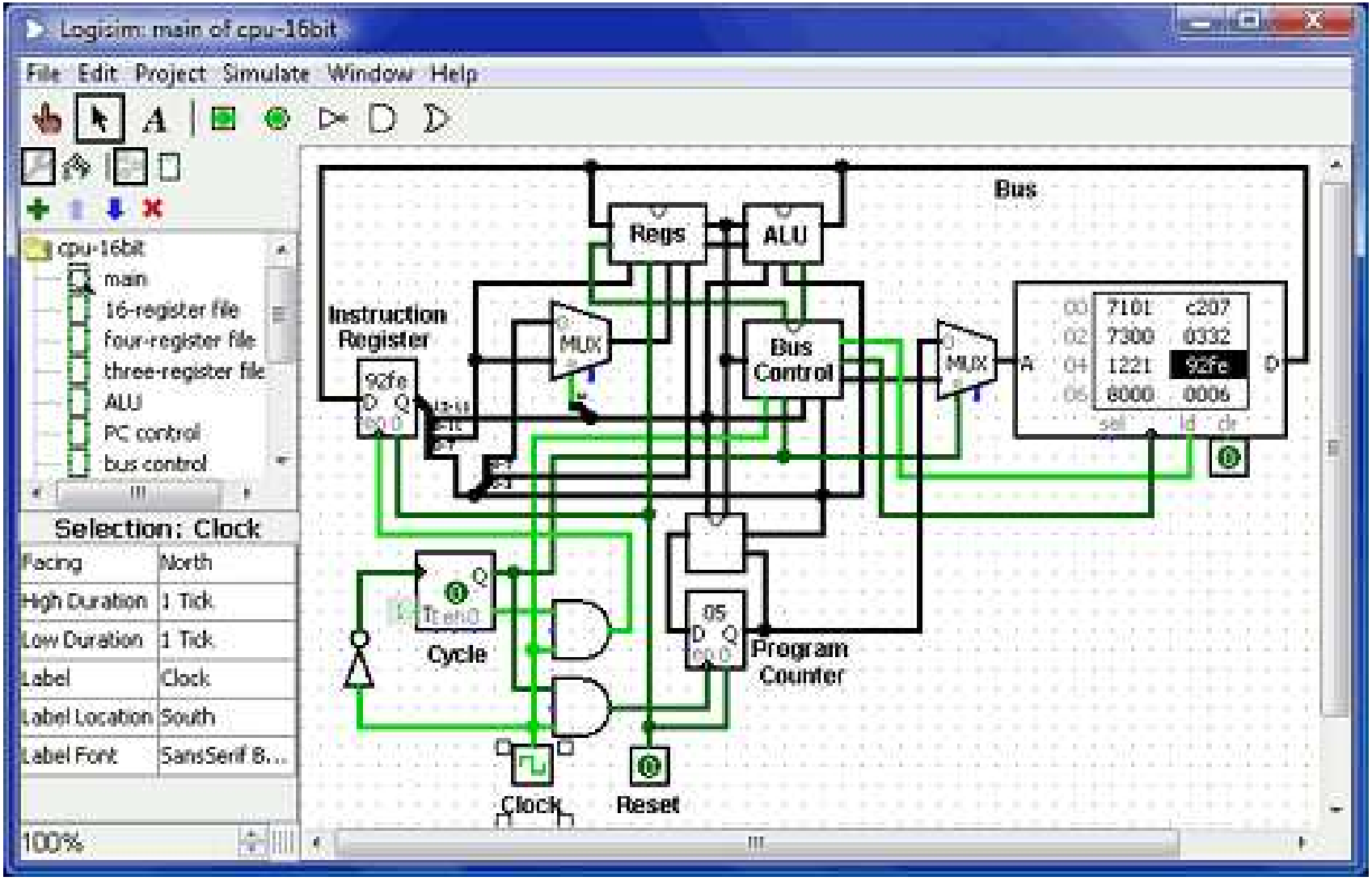


Software is part of ECU

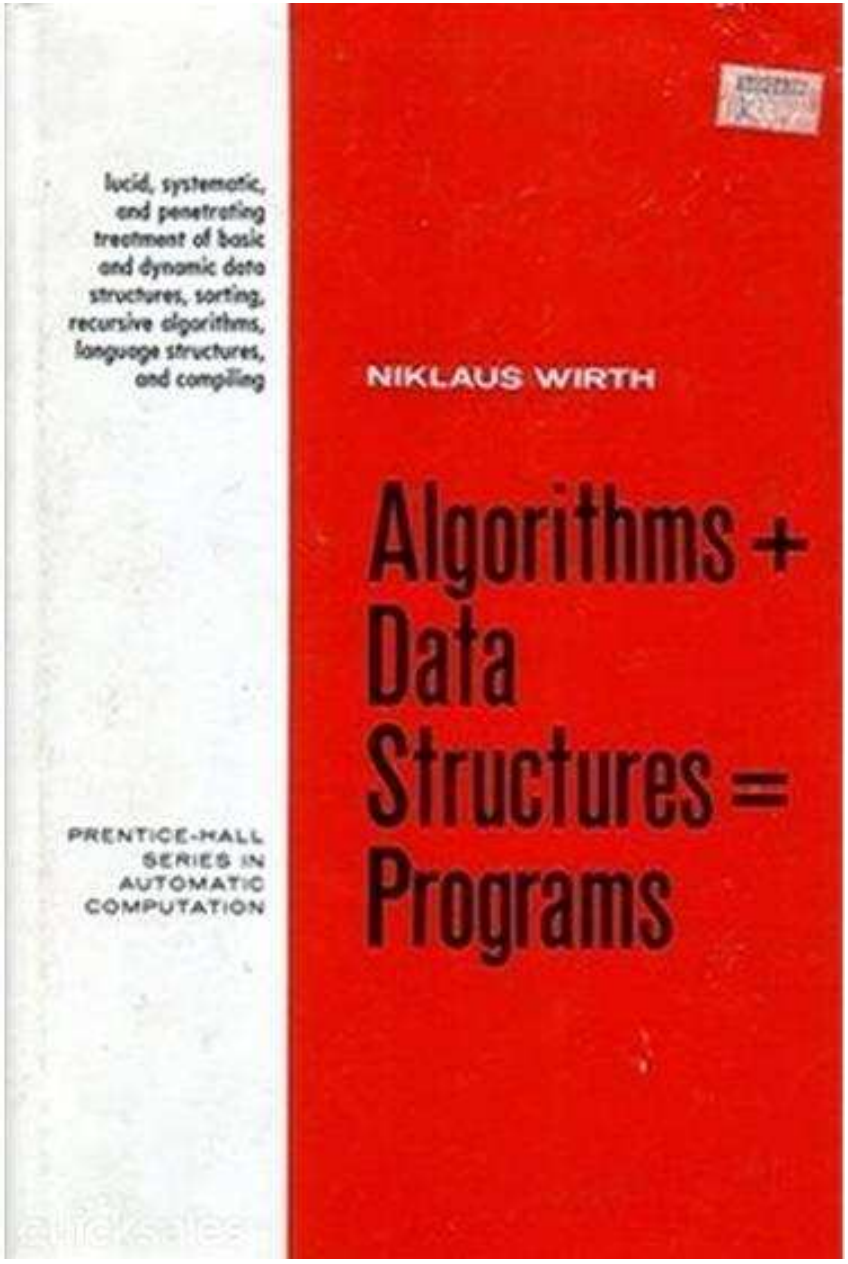
ECU's are part of mechatronic systems for measurement and control



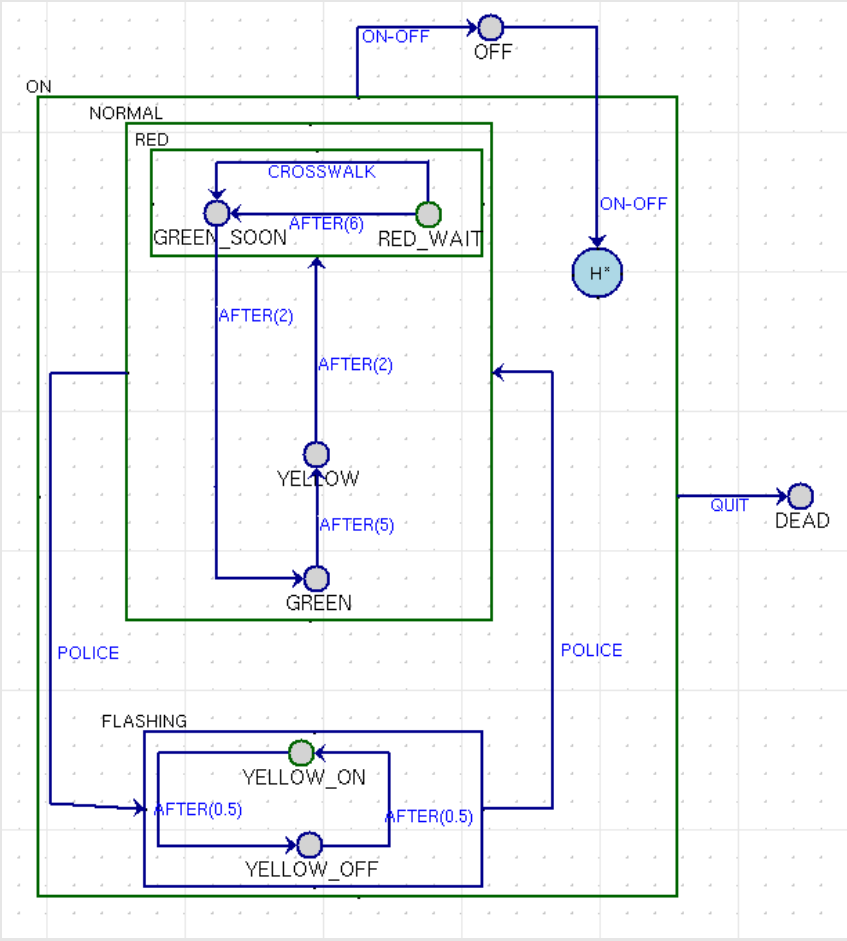

PCSrc
M u x
ID/EX
EX/MEM
MEM/WB
WB
M
EX
Control
RegWrite
IF/ID
ADD
4
Pipeline Register
Instruction
Registers
Read Register 1
Read Register 2
Write Register
Write Data
Read Data 1
Read Data 2
Instruction Memory
Address
PC
Shift Left 2
ADD
ADD Result
ALUSrc
Branch
MemWrite
Zero
ALU
ALU Result
Data Memory
Address
Read Data
Write Data
MemRead
MemtoReg
Instruction [15-0]
16
Sign Extend
32
6
ALU Control
ALUOp
Instruction [20-16]
Instruction [15-11]
RegDst


Logisim: main of cpu-16bit
File Edit Project Simulate Window Help
cpu-16bit
main
16-register file
four-register file
three-register file
ALU
PC control
bus control
Selection: Clock
Facing North
High Duration 1 Tick
Low Duration 1 Tick
Label Clock
Label Location South
Label Font SansSerif 8...
100%
Bus
Regs
ALU
Instruction Register
MUX
Bus Control
MUX
92fe
7101 c207
7300 0332
1221 92fc
8000 0006
Cycle
05
Program Counter
Clock
Reset


getConnection()
connection.
tSQL = "SELECT
statement.
next()) {


lucid, systematic, and penetrating treatment of basic and dynamic data structures, sorting, recursive algorithms, language structures, and compiling
NIKLAUS WIRTH
Algorithms + Data Structures = Programs
PRENTICE-HALL SERIES IN AUTOMATIC COMPUTATION

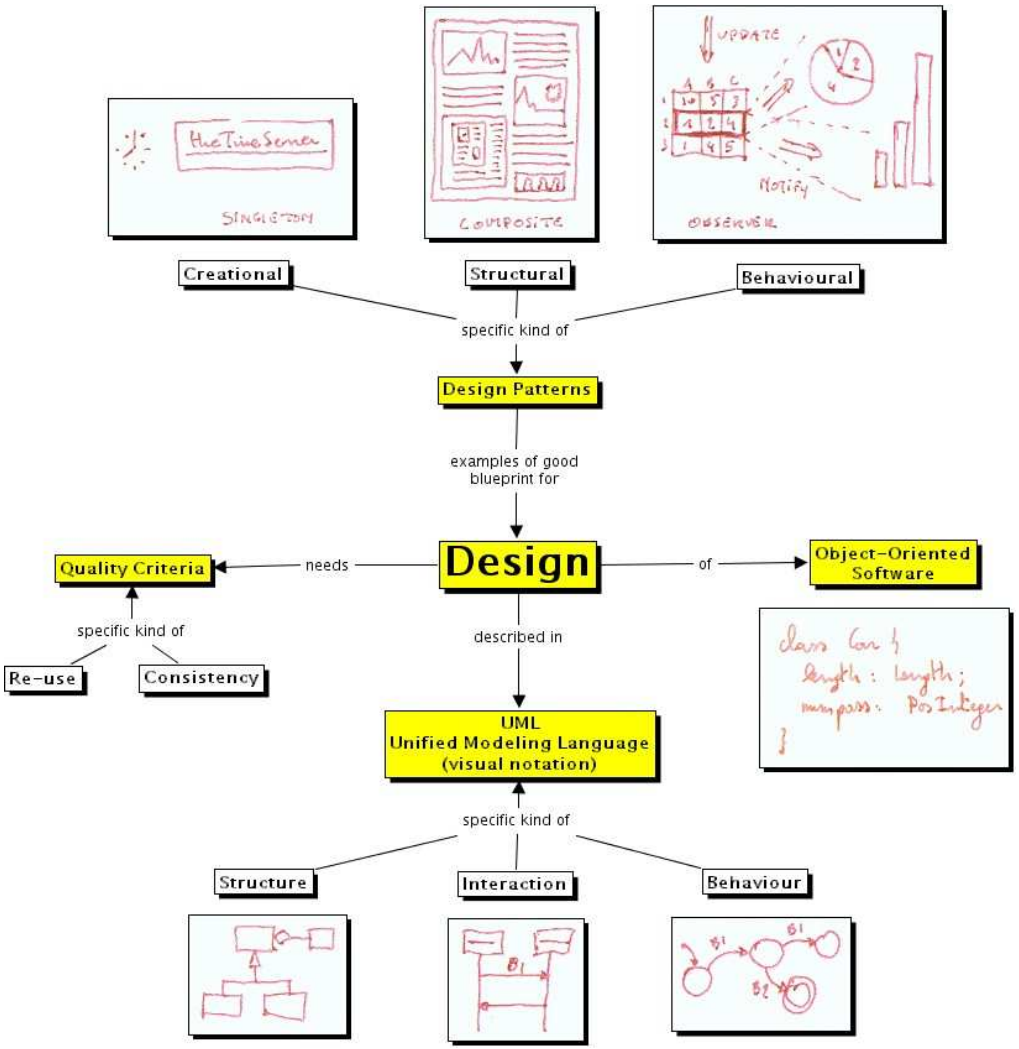
Model, don’t code (though code is a model too)



beware of **leaky abstractions**

“All non-trivial abstractions, to some degree, are leaky.” (Joel Spolsky)

https://www.joelonsoftware.com/articles/LeakyAbstractions.html


SINGLETON
COMPOSITE
UPDATE
Notify
Creational
Structural
Behavioural
specific kind of
Design Patterns
examples of good blueprint for
Quality Criteria
needs
Design
of
Object-Oriented Software
specific kind of
Re-use
Consistency
described in
UML
Unified Modeling Language
(visual notation)
specific kind of
Structure
Interaction
Behaviour

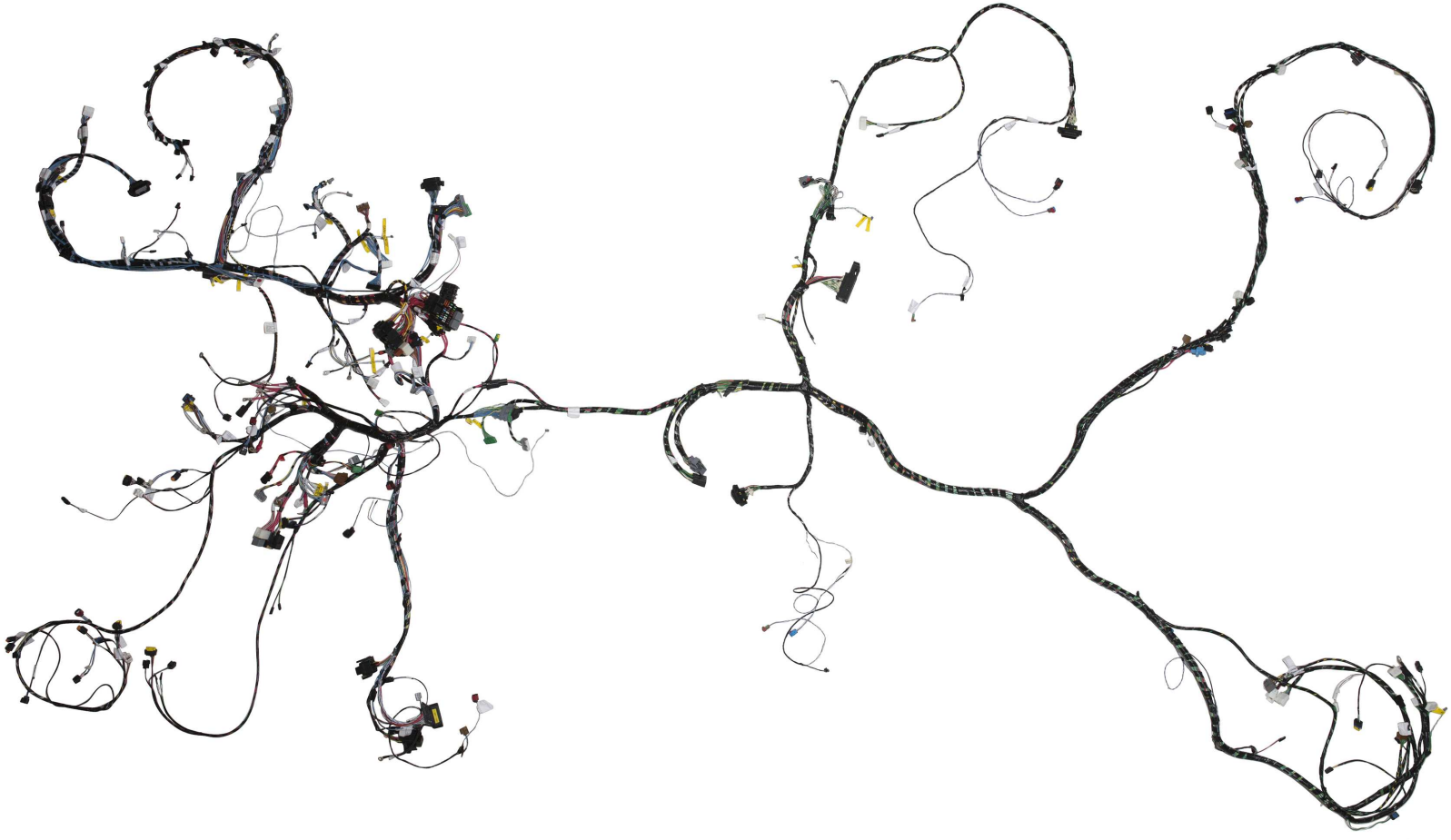

BOSCH
F 01R 00D 125
Made in China
3000018A40


MOTORAUTHORITY
WWW.MOTORAUTHORITY.COM

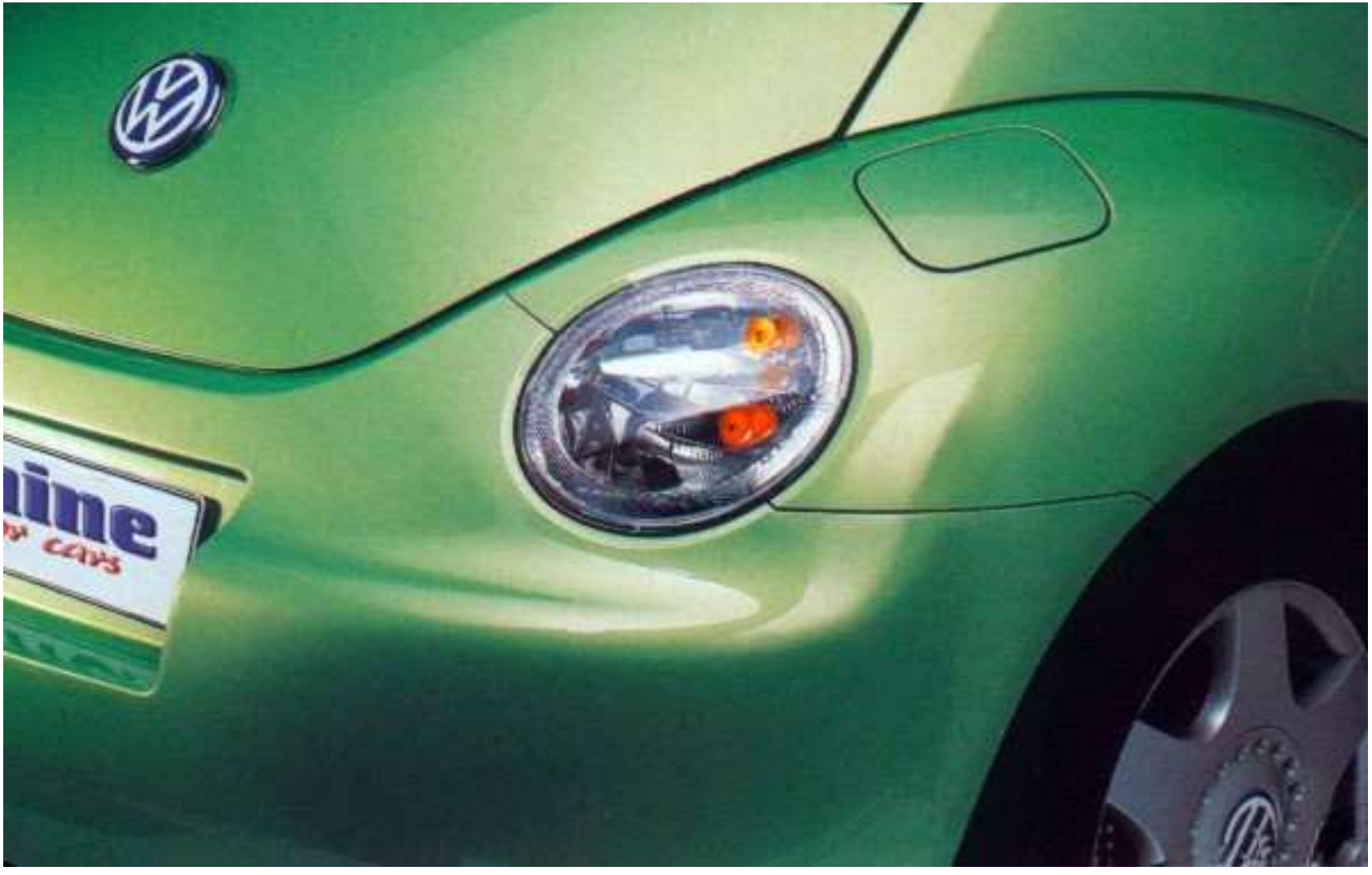
Mercedes S Class: >100 ECUs, >20 million LOC

VW Phaeton: wiring harness length > 2km, copper weight > 30kg

http://blog.nxp.com/the-wiring-harness-the-smart-communication-network-in-every-car/



http://blog.nxp.com/the-wiring-harness-the-smart-communication-network-in-every-car/



VW recalls 790,000 vehicles because of brake lights

Updated 2/26/2007 3:45 PM ET

Enlarge Volkswagen

2001-2007 New Beetles are part of the recall. An earlier recall for the same issue affected 1998-2002 Beetles.

WASHINGTON (AP) — Volkswagen of America said Monday it would recall 790,000 vehicles because of problems with the brake light switch.

The recall involves several vehicles: 1999-2006 model years of the Golf and GTI, 2001-2005 Jettas, 2001-2007 New Beetles and the 2004 R32. It expands upon a recall announced last year of some Jettas and New Beetles because of the same defect.

Volkswagen told the National Highway Traffic Safety Administration that the brake light switches in the vehicles could malfunction if they were improperly installed.

The automaker said the light could either remain on or not function, which would fail to provide other motorists with the proper braking signal and potentially lead to a crash.

In some vehicles with automatic transmissions, a faulty brake light could work in tandem with the shift interlock to immobilize the vehicle and require towing, said VW spokesman Keith Price.

Last year, VW recalled 362,000 Jetta and New Beetle sedans because of similar problems with the brake lights. That recall affected Jettas from the 1999-2002 model years and New Beetles from the 1998-2002 model years.

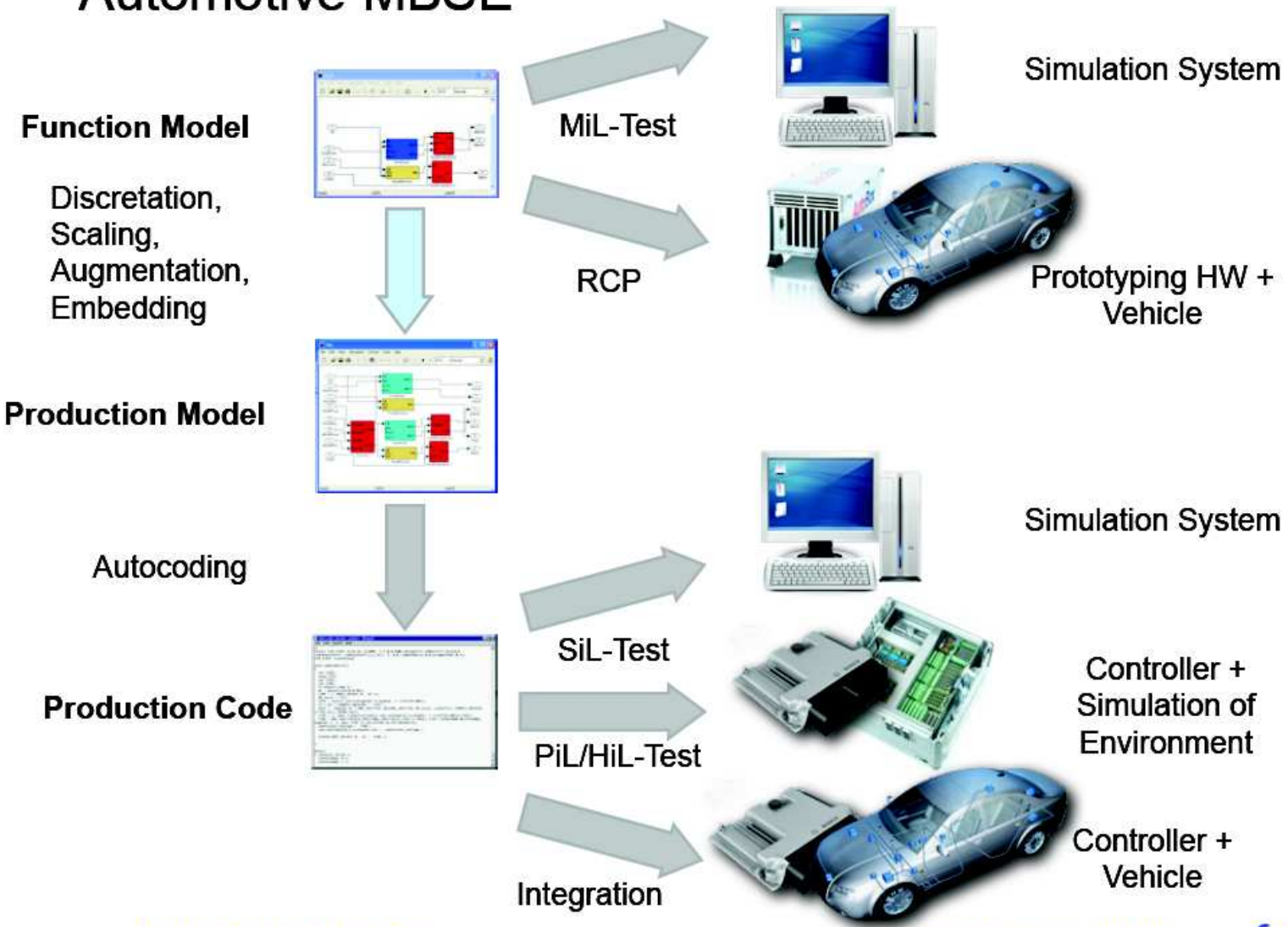
Price said the latest recall is an extension of the previous one because the company "found that there was a broader pool of vehicles that had the defective part."

He said owners of 2001-2002 Jettas and New Beetles who already had the repairs completed following last year's recall would not need to return for a second time.

VW dealers will install the newly designed brake light switch free of charge. The recall is expected to begin in late April and owners may contact VW with questions at 800-822-8987.

Posted 2/26/2007 11:02 AM ET

Automotive MBSE



Dealing with Complexity

Dealing with Complexity

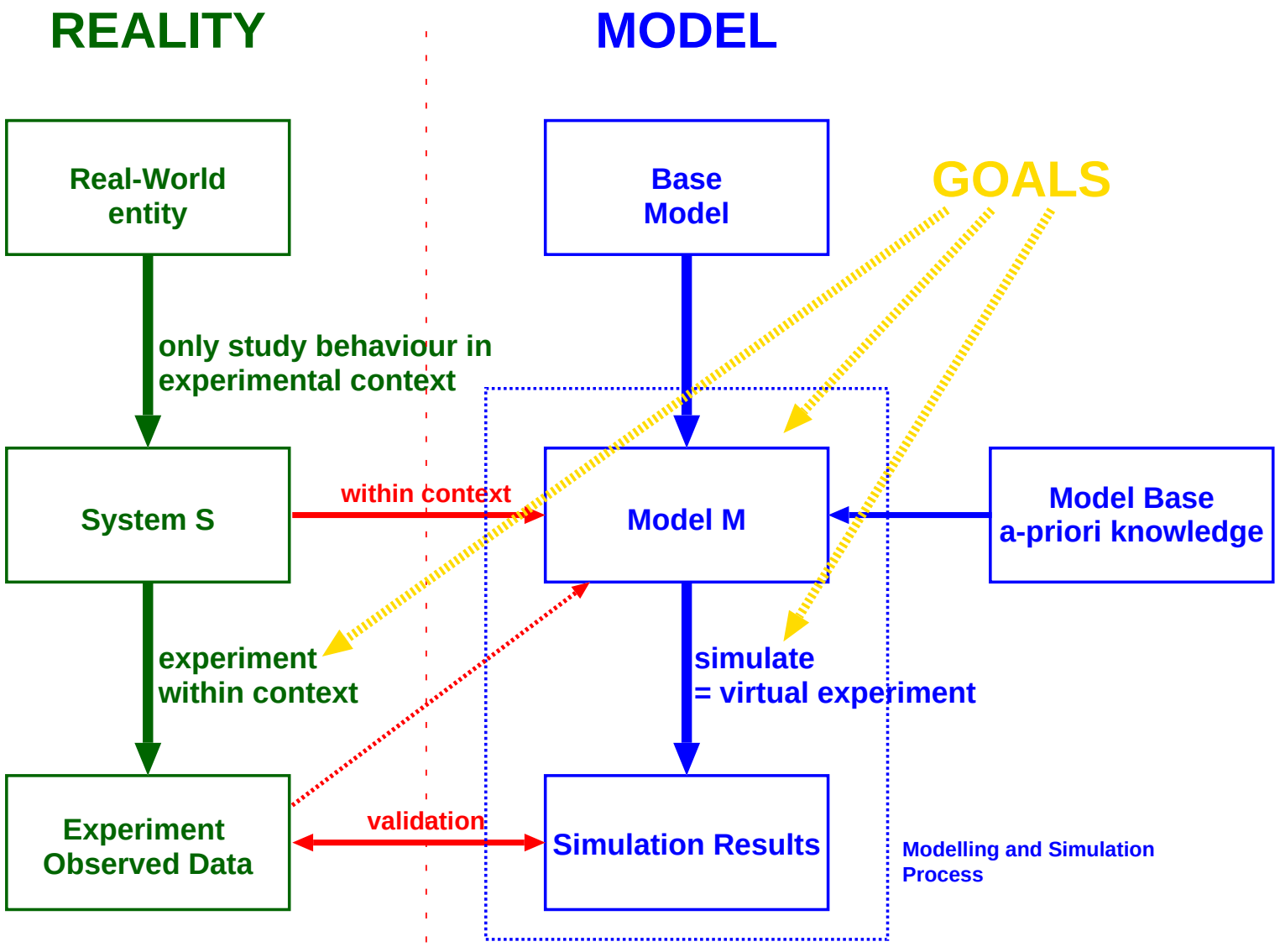
Model Everything . . . Explicitly

Dealing with Complexity

Model Everything . . . Explicitly
for **design** (Engineering) and **analysis** (Science)

A model is a depiction, representing the original.
A model is a reduction, capturing relevant aspects.
A model has a purpose, defining its use.

Herbert Stachowiak



Bernard P. Zeigler. *Multi-faceted Modelling and Discrete-Event Simulation*. Academic Press, 1984.

Simulation . . . when too costly/dangerous

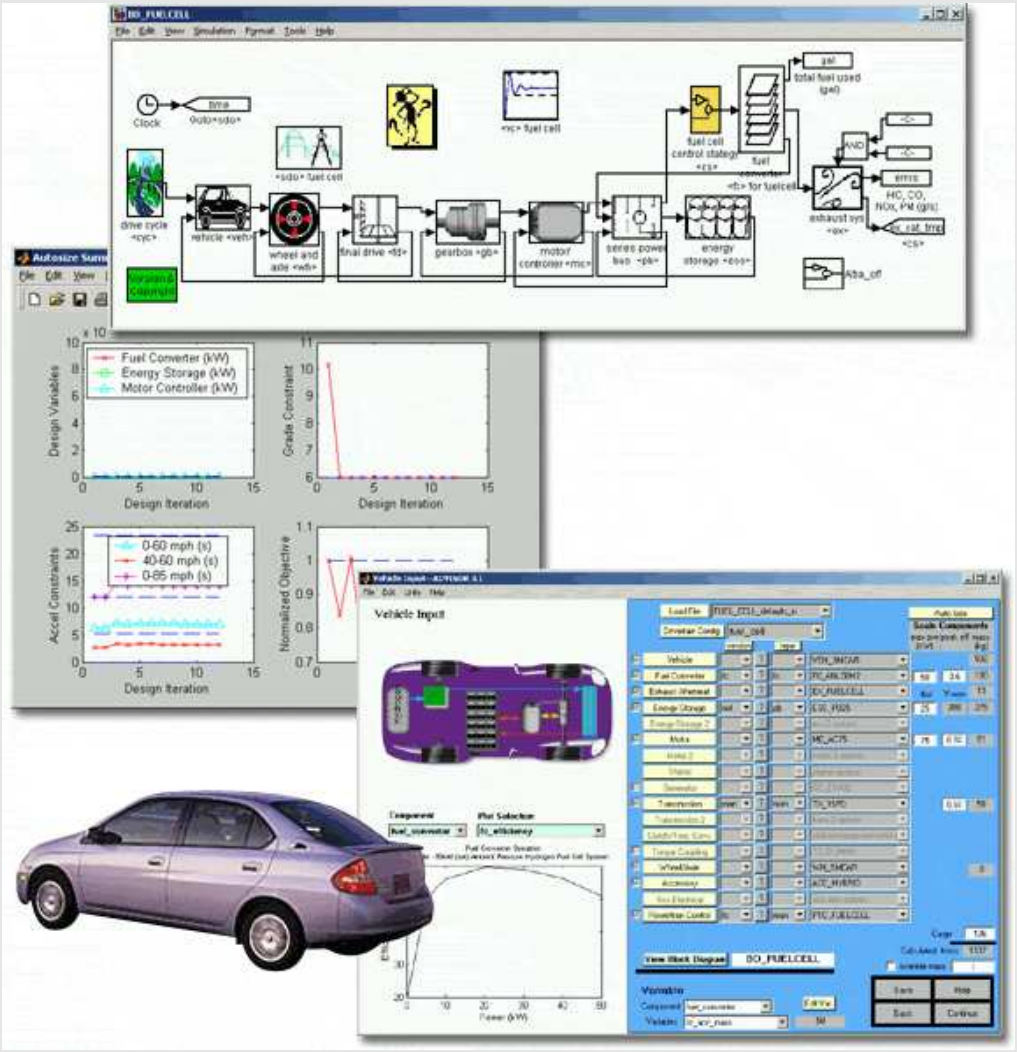
analysis $\leftrightarrow$ design

Simulation . . . real experiment not ethical

“physical” simulation, training

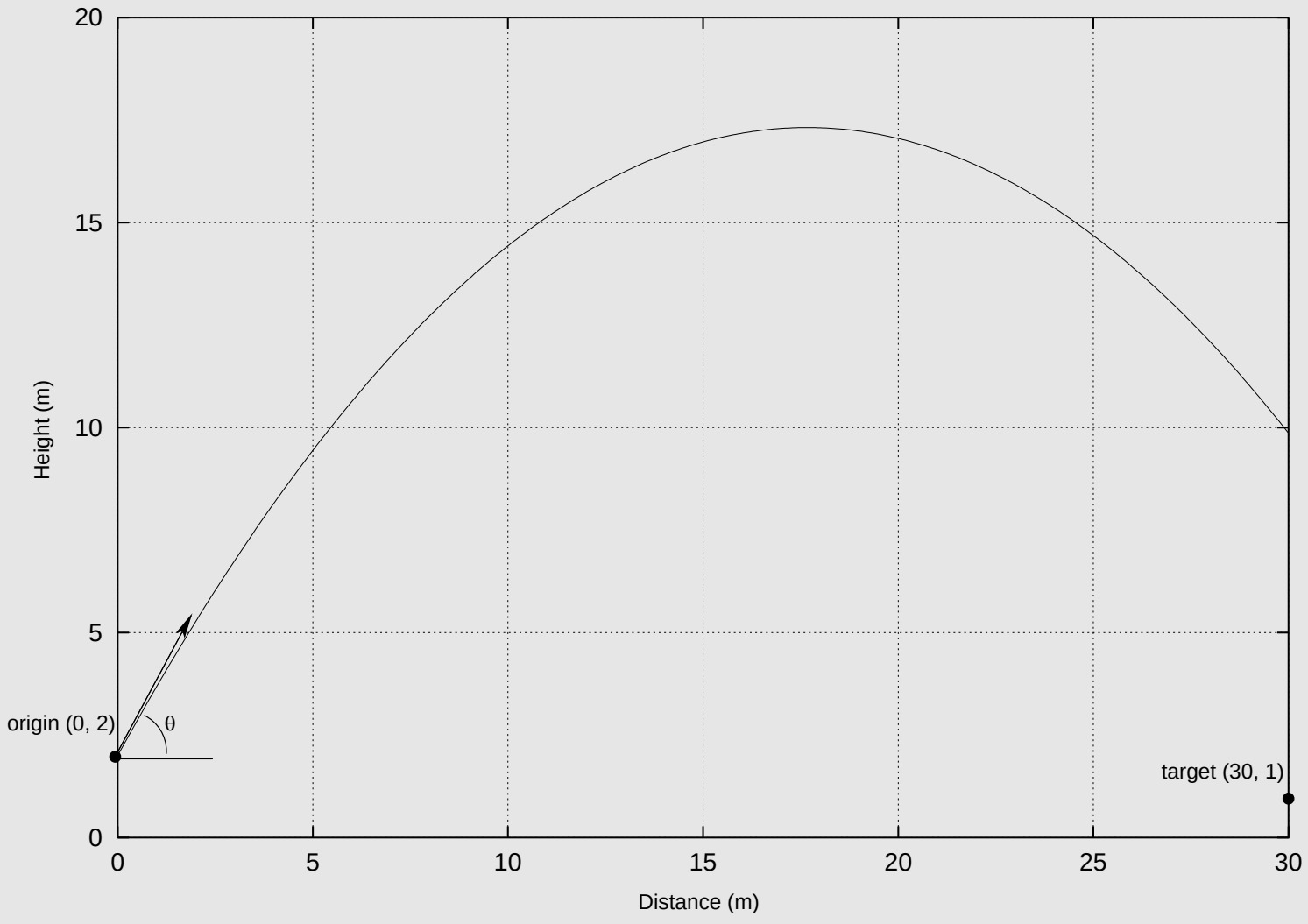
Simulation . . . evaluate alternatives

Simulation . . . “Do it Right the First Time”

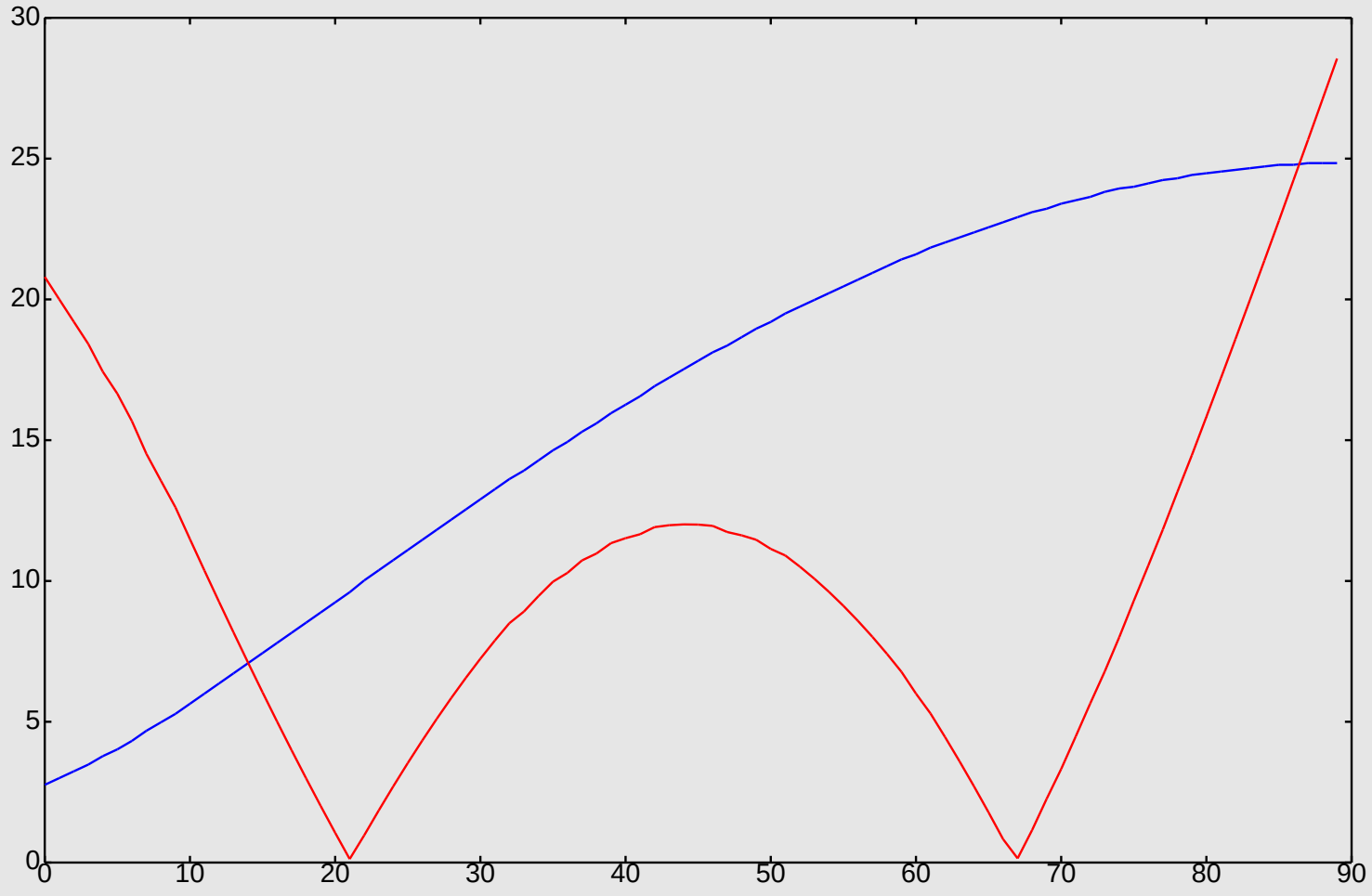


essence: “shooting” problems

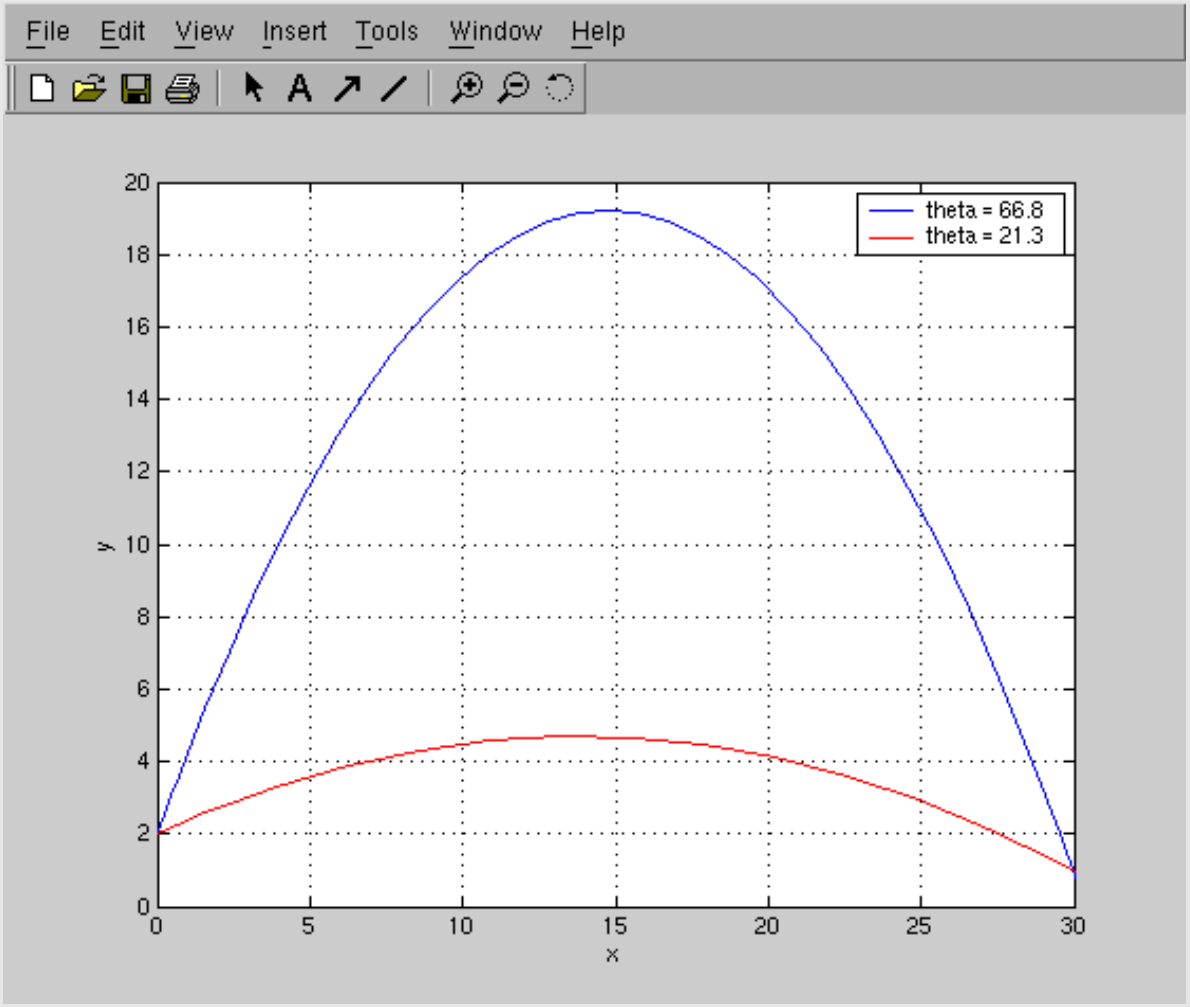
defining a “hit”



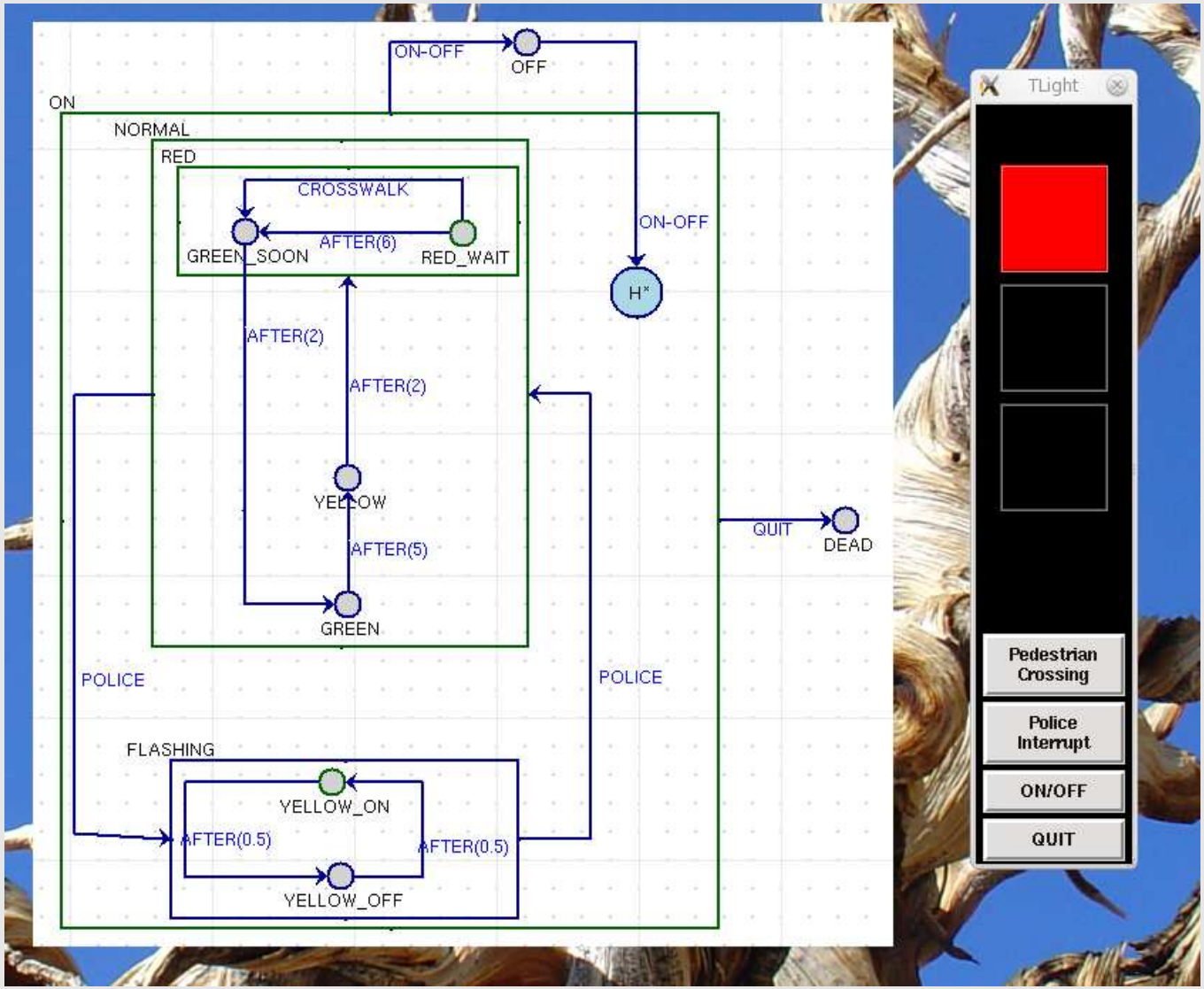
optimizing a “performance metric”



optimal solution. . . s



Modelling/Simulation . . . and code/app Synthesis



The spectrum of uses of models

- Documentation, Communication

The spectrum of uses of models

- Documentation, Communication
- Formal Verification of Properties
  (all models, all behaviours)

The spectrum of uses of models

- Documentation, Communication
- Formal Verification of Properties
  (all models, all behaviours)
- Model Checking of Properties
  (one model, all behaviours)

The spectrum of uses of models

- Documentation, Communication
- Formal Verification of Properties
  (all models, all behaviours)
- Model Checking of Properties
  (one model, all behaviours)
- Test Generation

The spectrum of uses of models

- Documentation, Communication
- Formal Verification of Properties
  (all models, all behaviours)
- Model Checking of Properties
  (one model, all behaviours)
- Test Generation
- Simulation (one model, one behaviour)
  . . . for calibration, optimization, . . .

The spectrum of uses of models

- Documentation, Communication
- Formal Verification of Properties
  (all models, all behaviours)
- Model Checking of Properties
  (one model, all behaviours)
- Test Generation
- Simulation (one model, one behaviour)
  . . . for calibration, optimization, . . .
- Application Synthesis (software, FPGA, 3D printing,
  production line control, . . .)

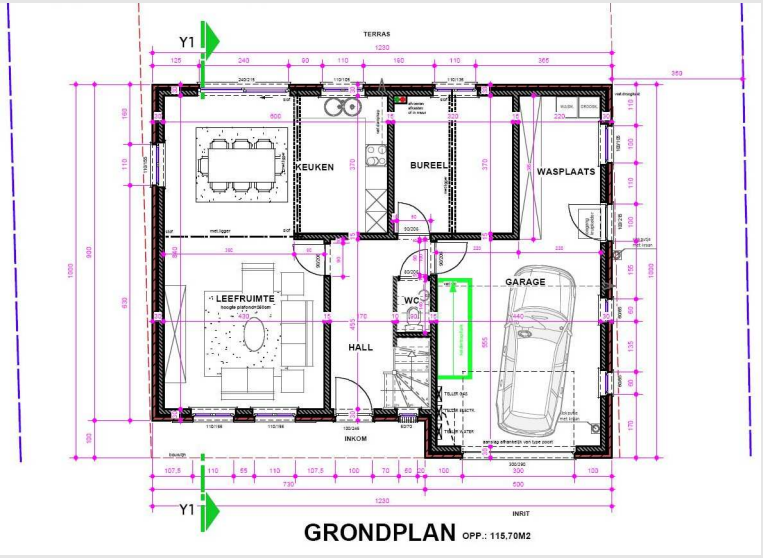
Requirements (“What?”)

- Detached or Semi-detached
- Style (classical, modern, . . .)
- Number of Floors
- Number of rooms of different types (bedrooms, bathrooms, . . .)
- Garage, Storage, . . .
- Cellar
- . . .

Requirements (“What?”)

- Detached or Semi-detached
- Style (classical, modern, . . .)
- Number of Floors
- Number of rooms of different types (bedrooms, bathrooms, . . .)
- Garage, Storage, . . .
- Cellar
- . . .

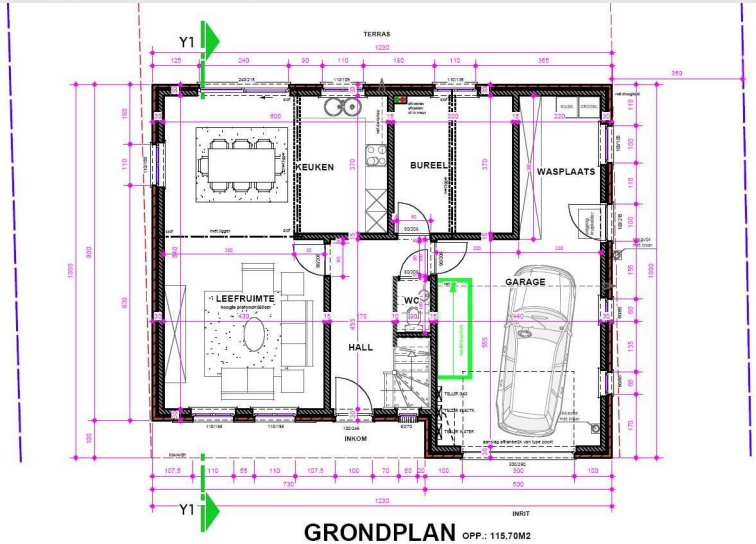
Design (“How?”)



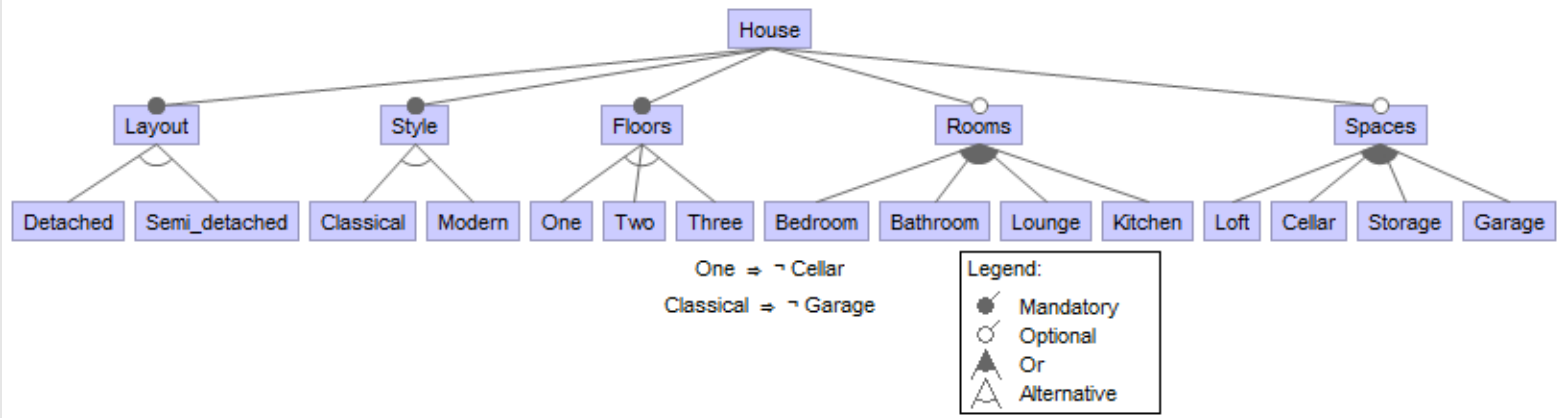
Requirements (“What?”)

- Detached or Semi-detached
- Style (classical, modern, . . .)
- Number of Floors
- Number of rooms of different types (bedrooms, bathrooms, . . .)
- Garage, Storage, . . .
- Cellar
- . . .

Design (“How?”)



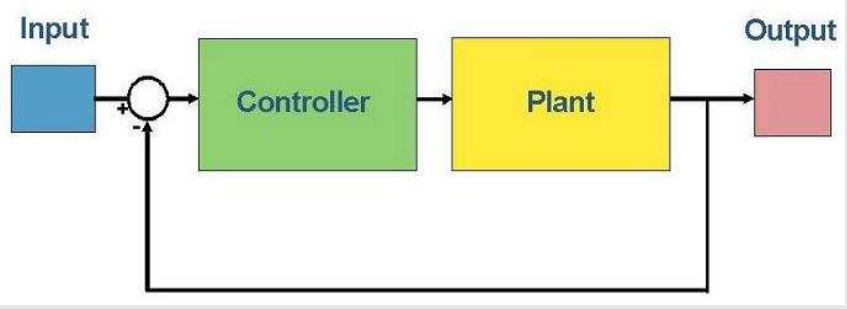
Requirements (“What?”): Feature Model of a Product Family



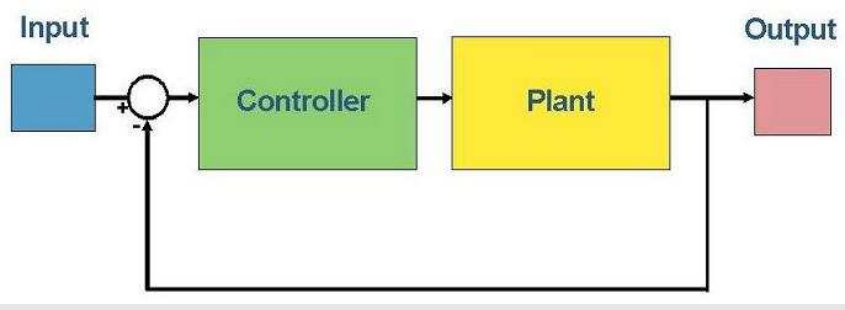
System Boundaries

- **System** to be built/studied
- **Environment** with which the system interacts

System vs. “Plant”

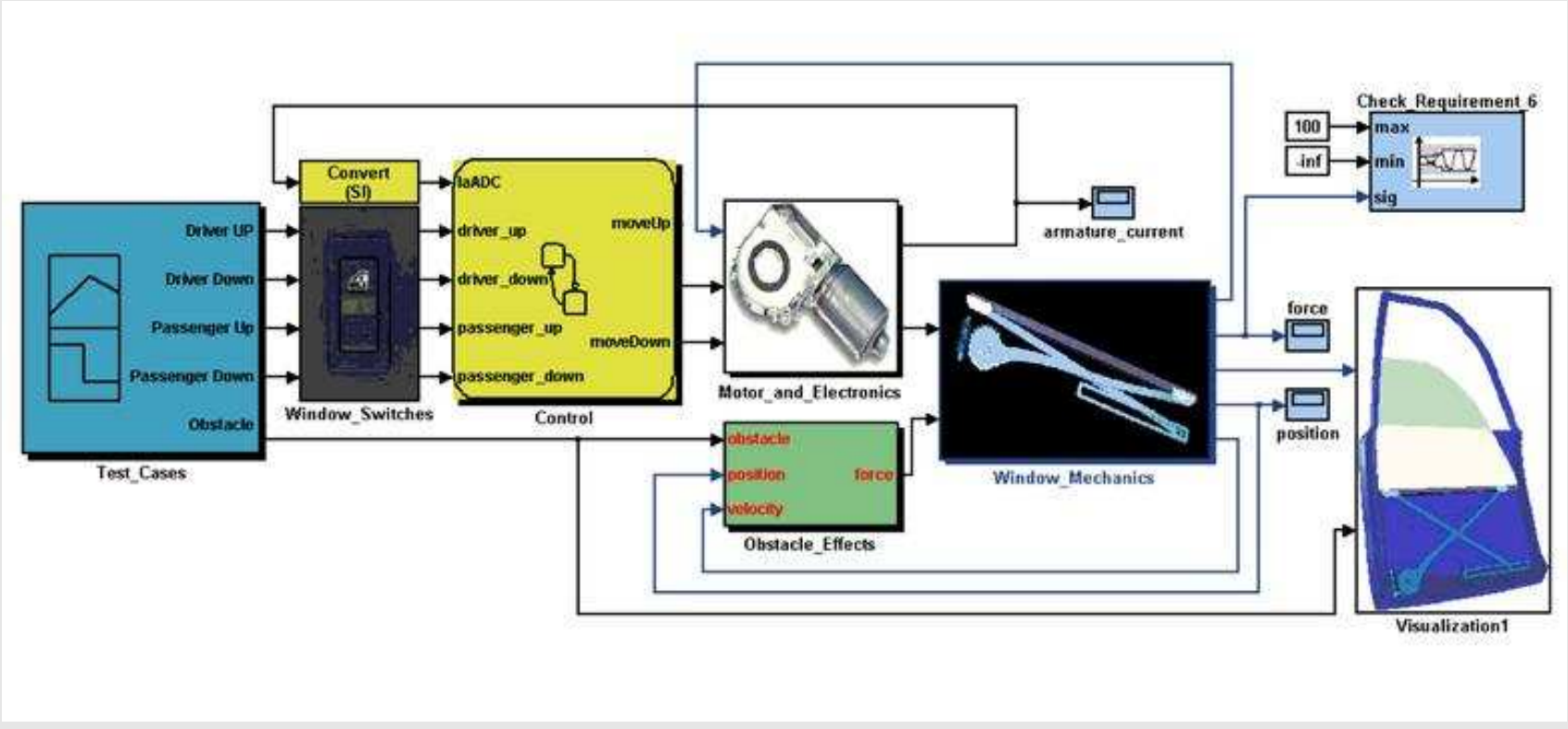


System vs. “Plant”



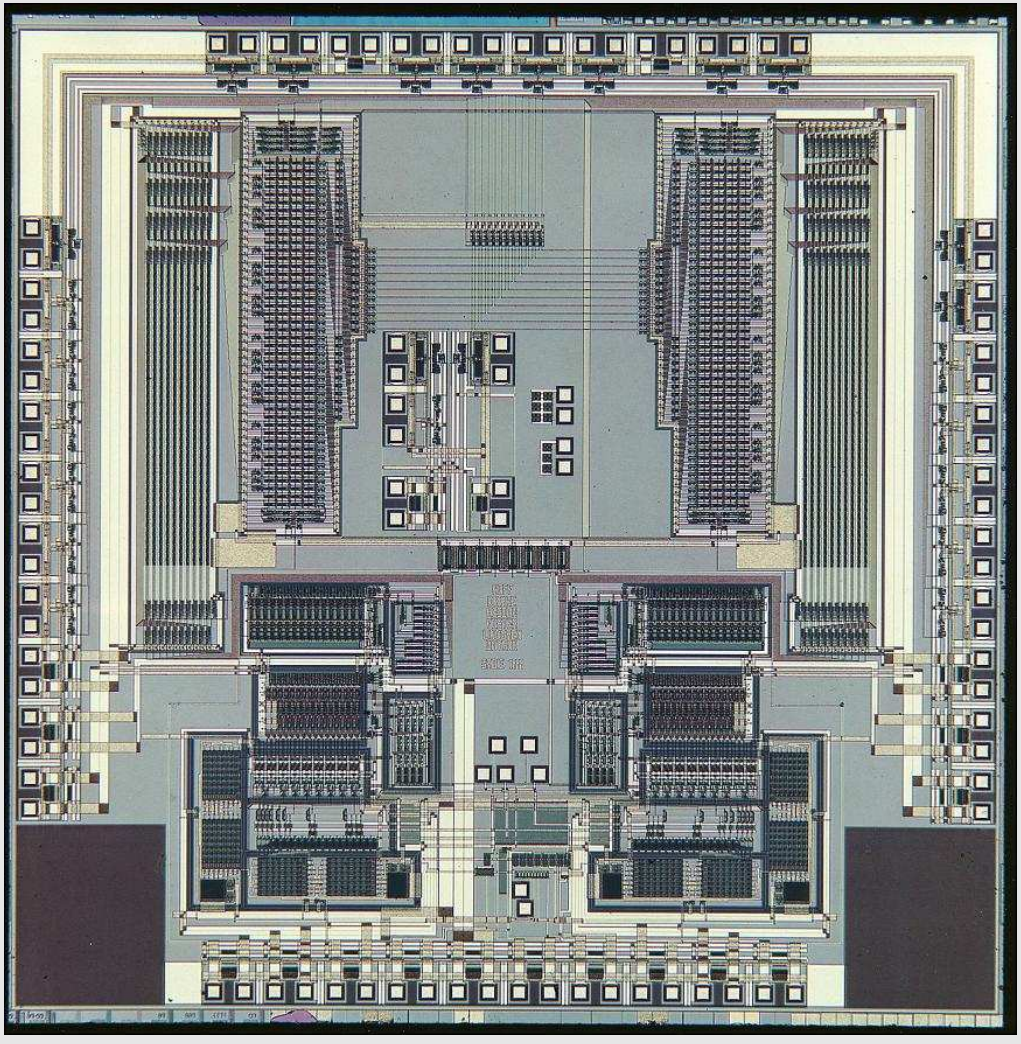
“Plant”?!

System vs. “Plant”



www.mathworks.com/products/demos/simulink/PowerWindow/html/PowerWindow1.html

Number of Components

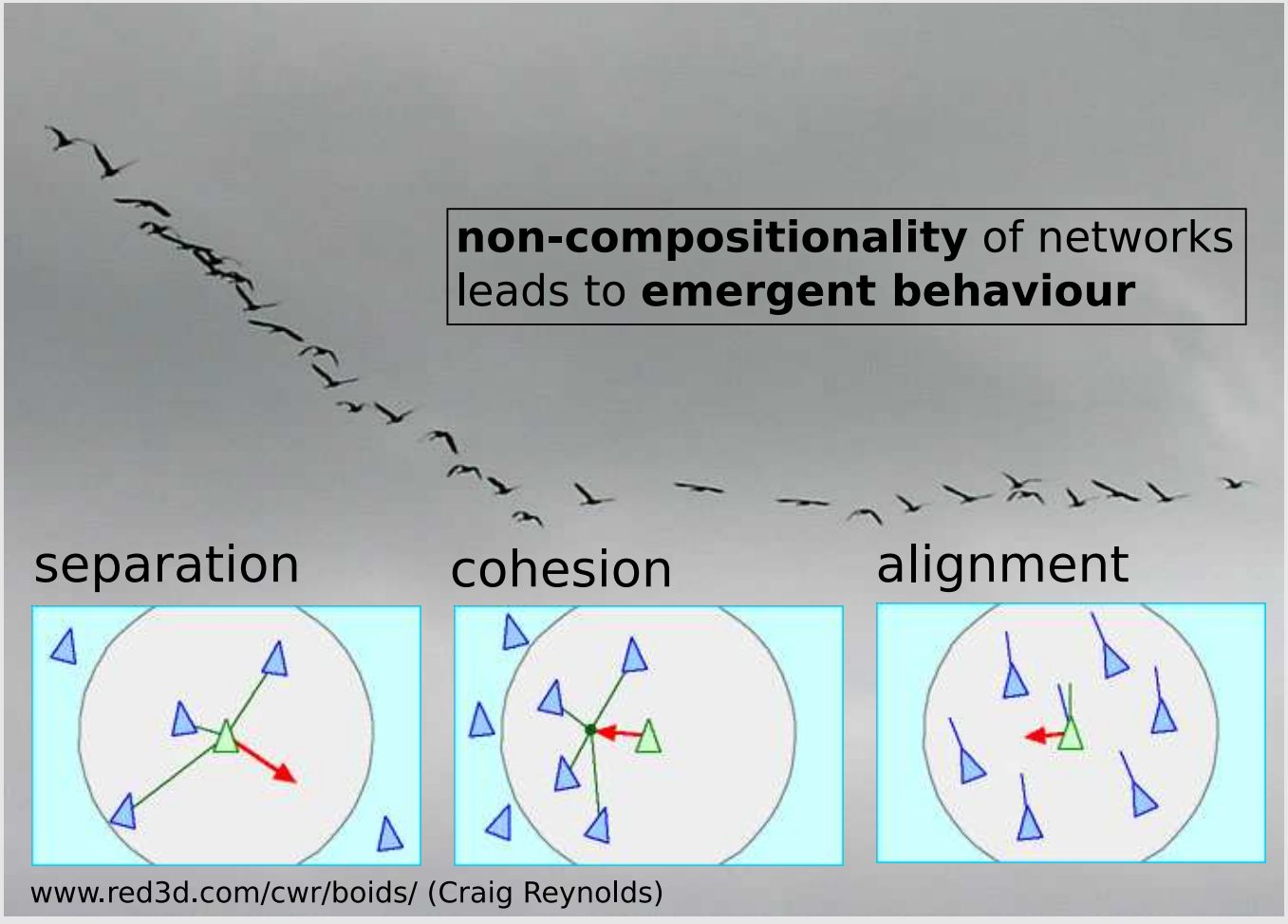


Crowds: diversity, interaction

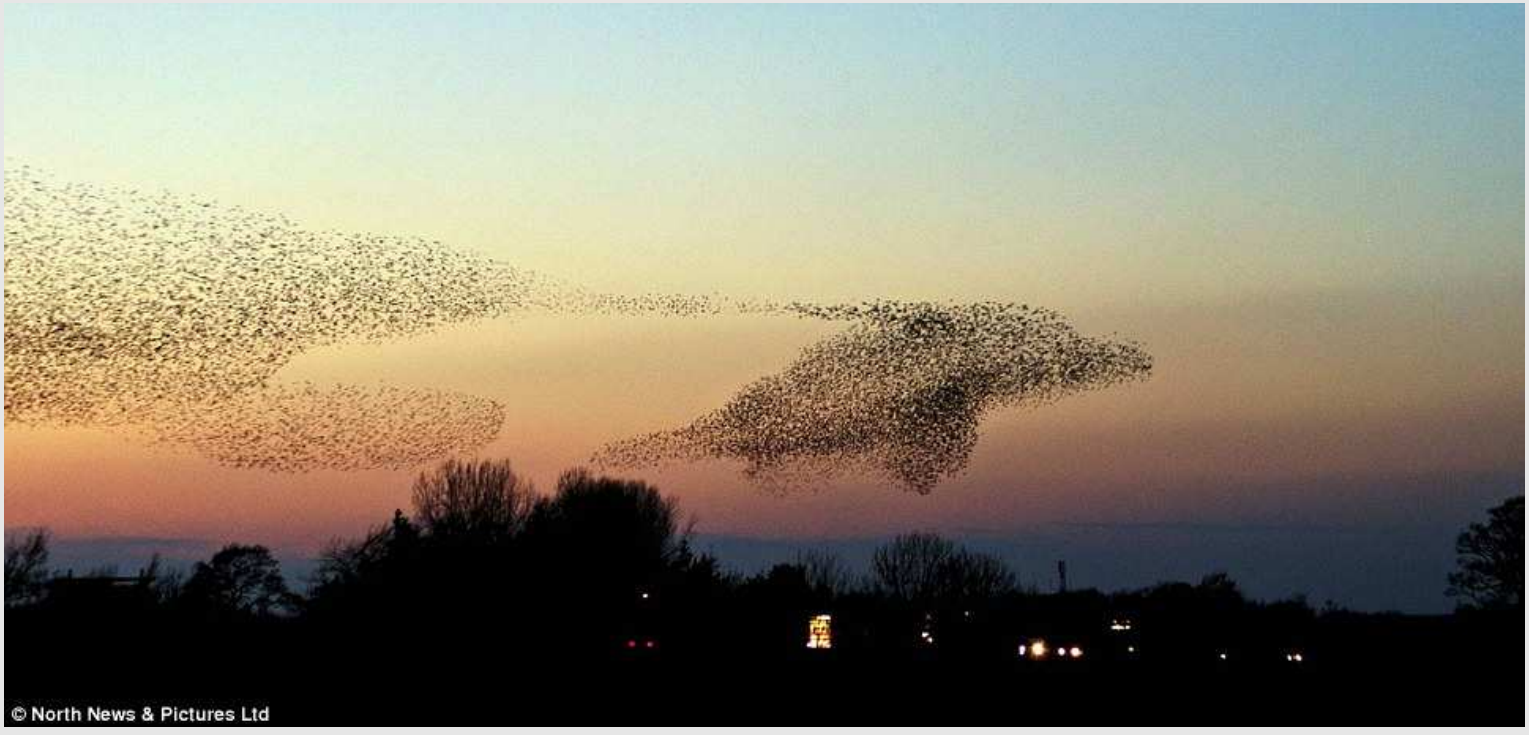
www.3dm3.com

Diversity of Components: Power Window

Non-compositional/Emergent Behaviour



Emergent Behaviour

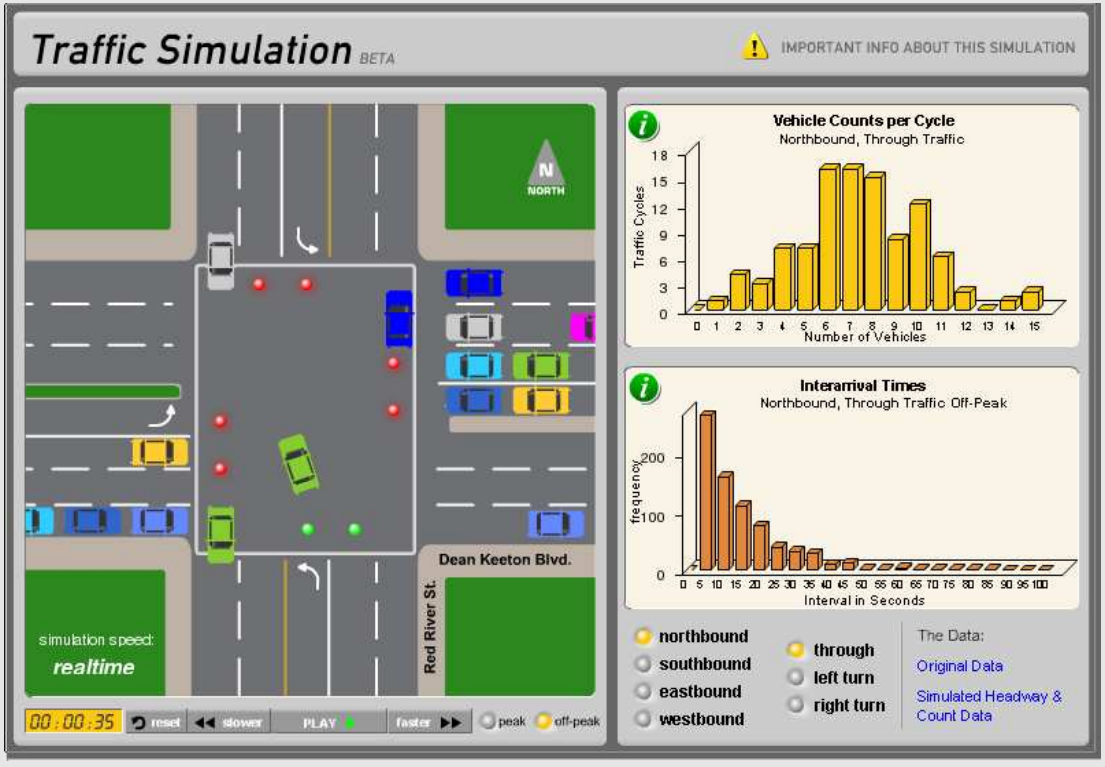


Engineered Emergent Behaviour

Robert Bogue. *Swarm intelligence and robotics.* Industrial Robot: An International Journal. 35(6):488 - 495, 2008.

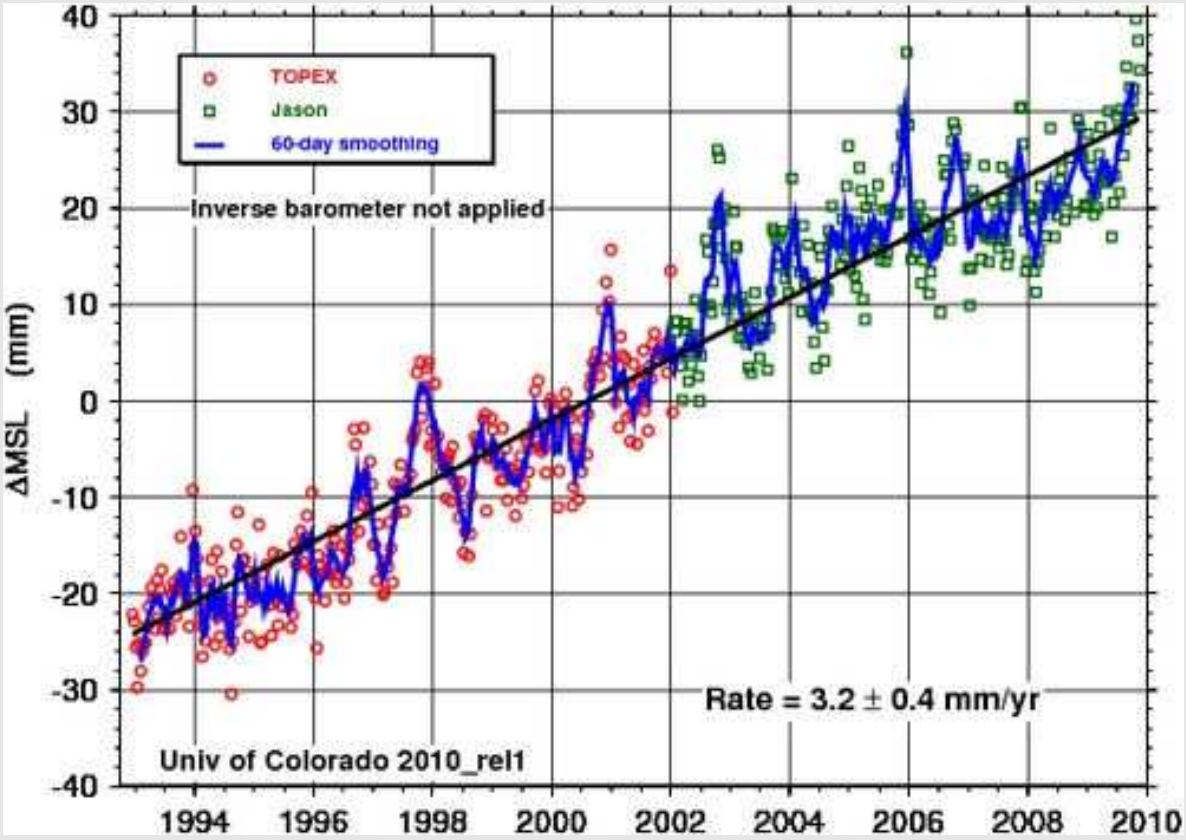
Uncertainty

Often related to level of abstraction:
for example continuous vs. discrete

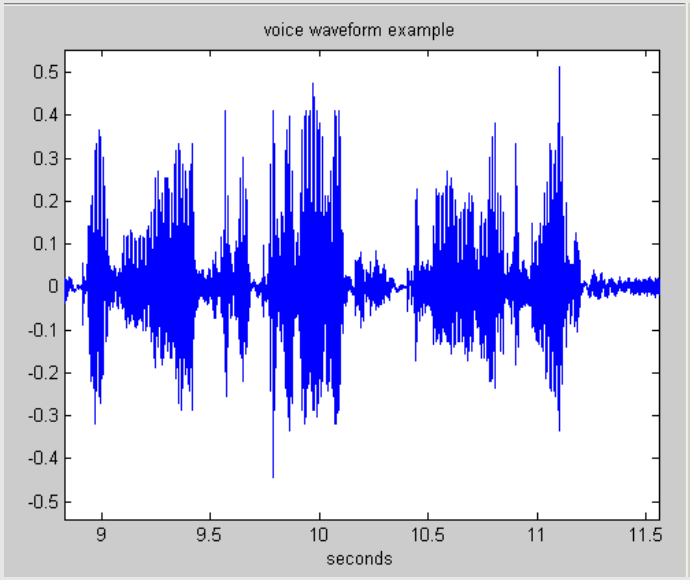


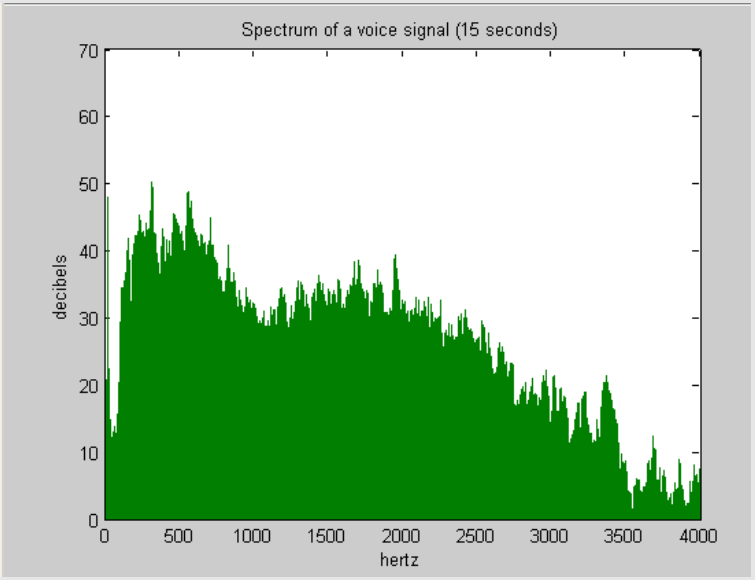
www.engr.utexas.edu/trafficSims/

Question: is the deviation from the trend periodic?



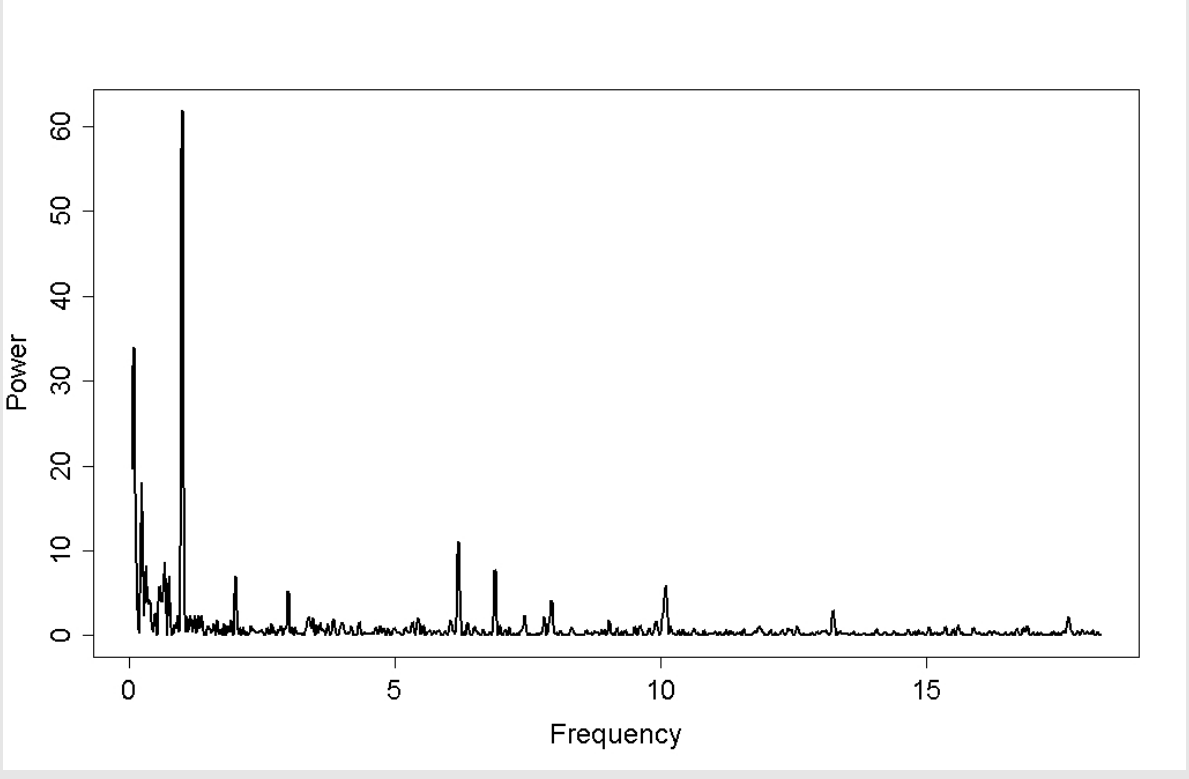
Fourier Transform: time-domain $\rightarrow$ frequency-domain





$$f(t) = \int_0^{+\infty} [a(\lambda)cos(2\pi\lambda t) + b(\lambda)sin(2\pi\lambda t)]d\lambda$$

Answer: transform to make the solution obvious

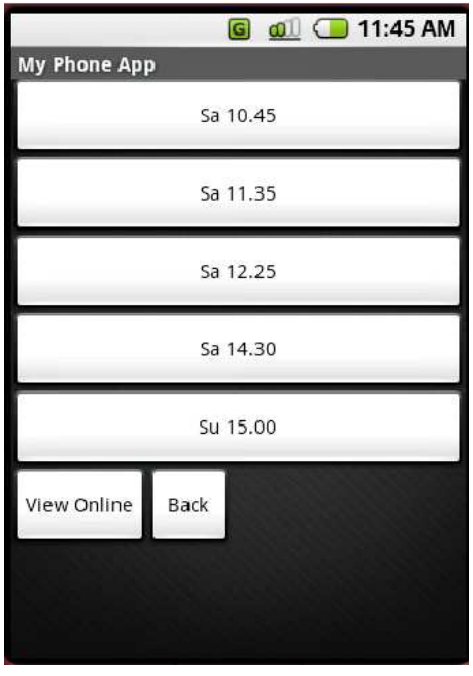


Guiding principle (∼ physics: principle of minimal action)

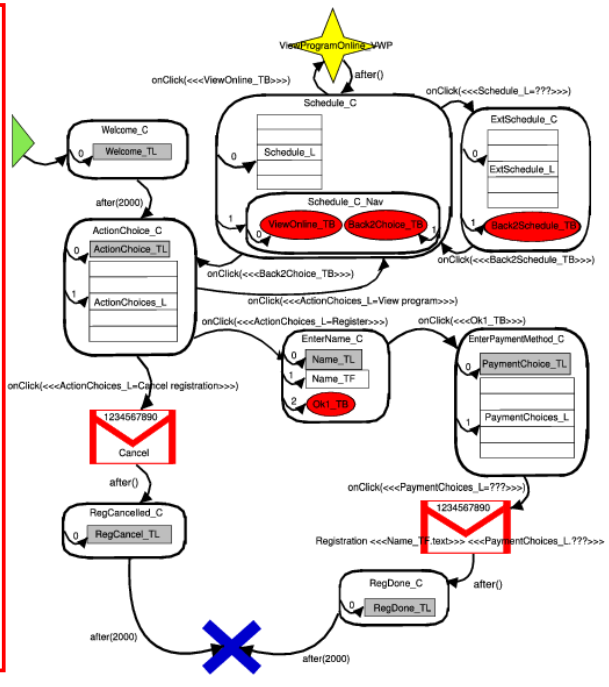
minimize **accidental complexity**,
only **essential complexity** remains

Fred P. Brooks. No Silver Bullet – Essence and Accident in Software Engineering.
Proceedings of the IFIP Tenth World Computing Conference, pp. 1069–1076, 1986.

`http://www.lips.utexas.edu/ee382c-15005/Readings/Readings1/05-Broo87.pdf`



```
package android.ger_code.PhoneApp;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnKeyListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.EditText;
import java.util.HashMap;


public class PhoneApp extends Activity
{
    private HashMap<String,CharSequence> pastEntries = new HashMap<String,
    private String currentAction = "";
    private PhoneAppStateChart appStateChart;
    public static PhoneApp instance;

    public void sendEventToStateChart(String event)
    {
        appStateChart.handleEvent(event);
    }

    public void onCreate(Bundle icicle)
    {
        super.onCreate(icicle);
        instance = this;
        appStateChart = new PhoneAppStateChart();
        appStateChart.handleEvent("START");
    }

    //NOTE: using this method causes a crash when the application exits
    public void onDestroy()
    {
        if( isFinishing() )
            Atom3Communicator.highlightCurrentState("!!!");
    }

    public void onWindowFocusChanged(boolean hasFocus)
    {
        super.onWindowFocusChanged(hasFocus);
        if( hasFocus )
        {
            sendEventToStateChart("onComplete(<<<"+currentAction+">>>)");
            currentAction = "";
        }
    }

public void action_SendCancelRegMsg_SM()
{
    String dest = "1234567890";
    String message = "Cancel";
    Intent i = new Intent(Intent.ACTION_SENDTO, Uri.parse("mailto:"+dest+"
    i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    i.setFlags(Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
    startActivityForResult(i,1);
    currentAction="SendCancelRegMsg_SM";
}
```



Dealing with Complexity: some approaches

Dealing with Complexity: some approaches

- multiple abstraction levels

Dealing with Complexity: some approaches

- multiple abstraction levels
- optimal formalism

Dealing with Complexity: some approaches

- multiple abstraction levels
- optimal formalism
- multiple formalisms

Dealing with Complexity: some approaches

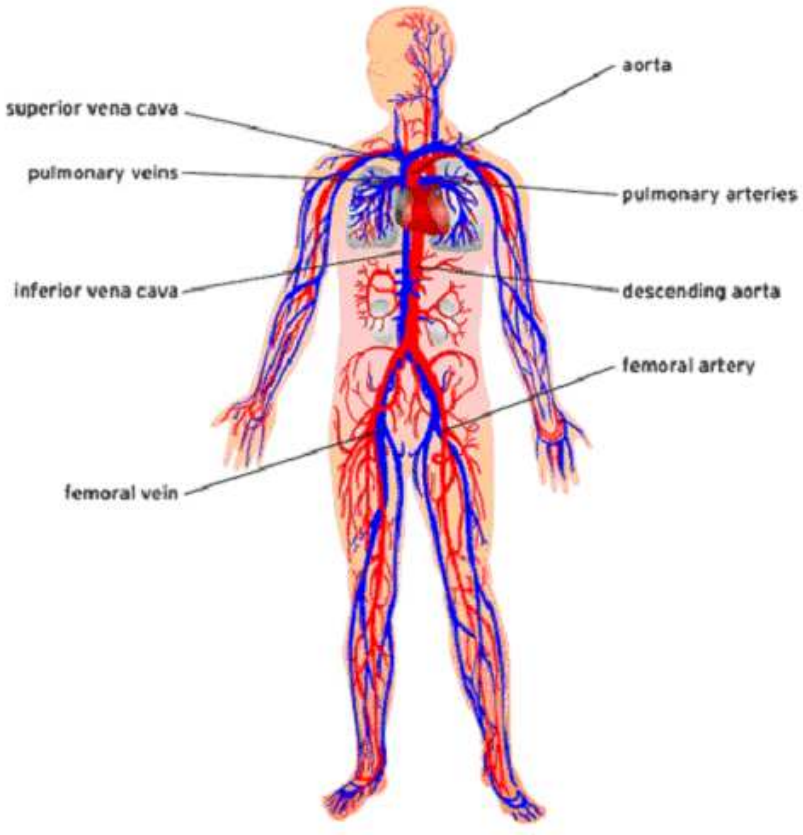
- multiple abstraction levels
- optimal formalism
- multiple formalisms
- multiple views

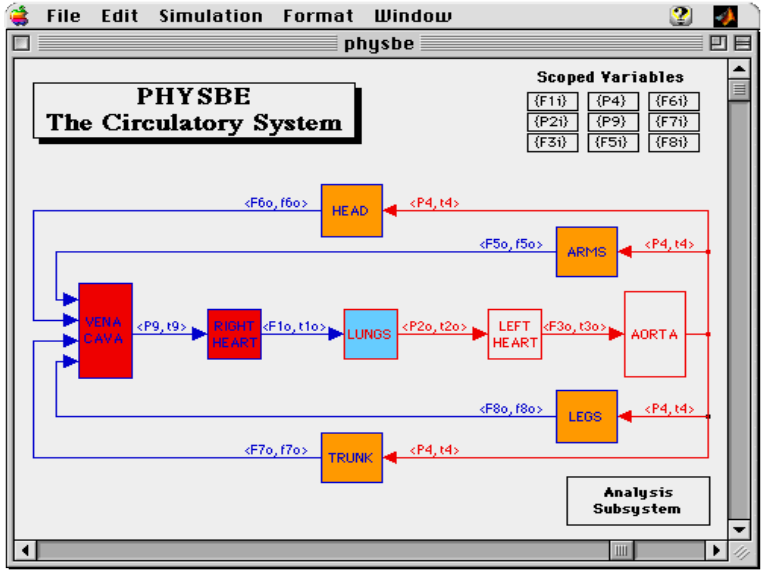
Dealing with Complexity: some approaches

- multiple abstraction levels
- optimal formalism
- multiple formalisms
- multiple views

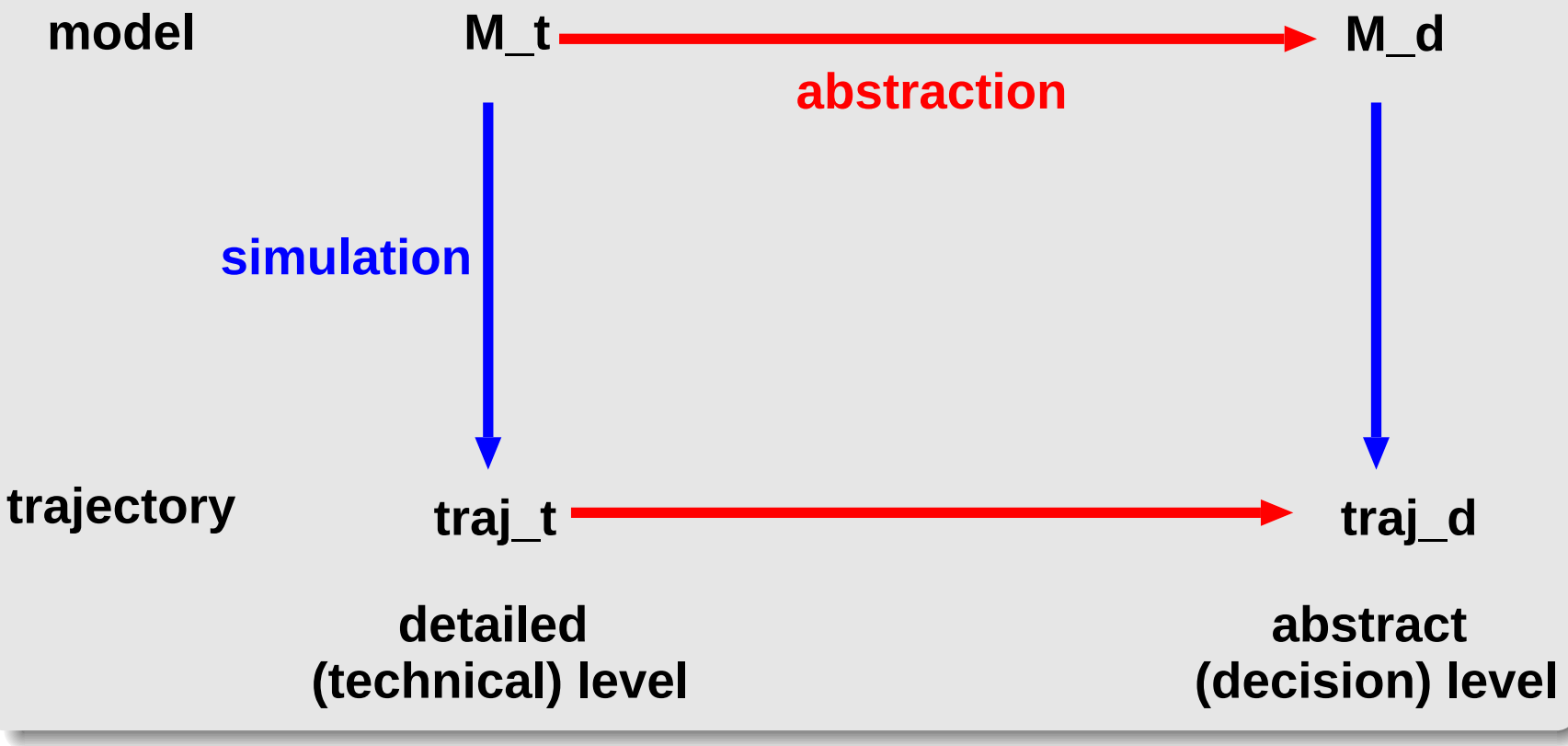
Modularity!

Different Abstraction Levels – properties preserved





Levels of Abstraction/Views: Morphism



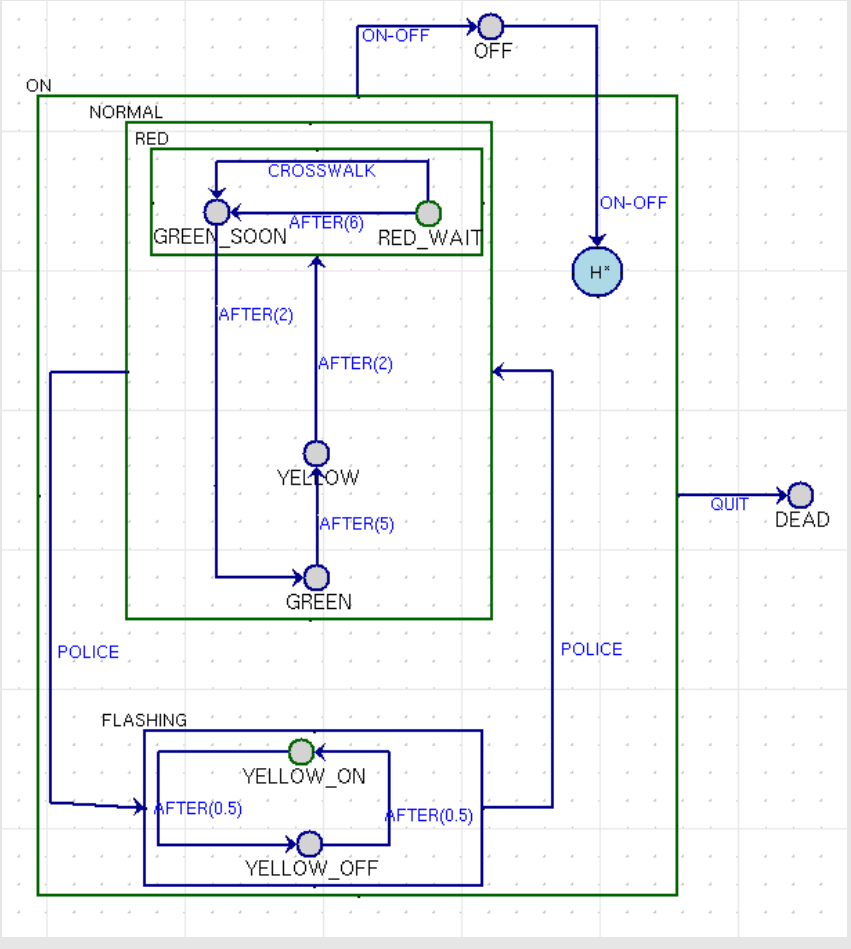
Abstraction Relationship

foundation: the *information* contained in a model M.
Different *questions* (properties) $P = I(M)$ which can be asked concerning the model.
These questions either result in true or false.

Abstraction and its opposite, *refinement* are *relative to a non-empty set of questions* (properties) P.

- If M_1 is an *abstraction* of M_2 with respect to P, for all $p \in P$: $M_1 \models p \Rightarrow M_2 \models p$. This is written $M_1 \sqsupseteq_P M_2$.
- M_1 is said to be a *refinement* of M_2 iff M_2 is an *abstraction* of M_1. This is written $M_1 \sqsubseteq_P M_2$.

Most Appropriate Formalism

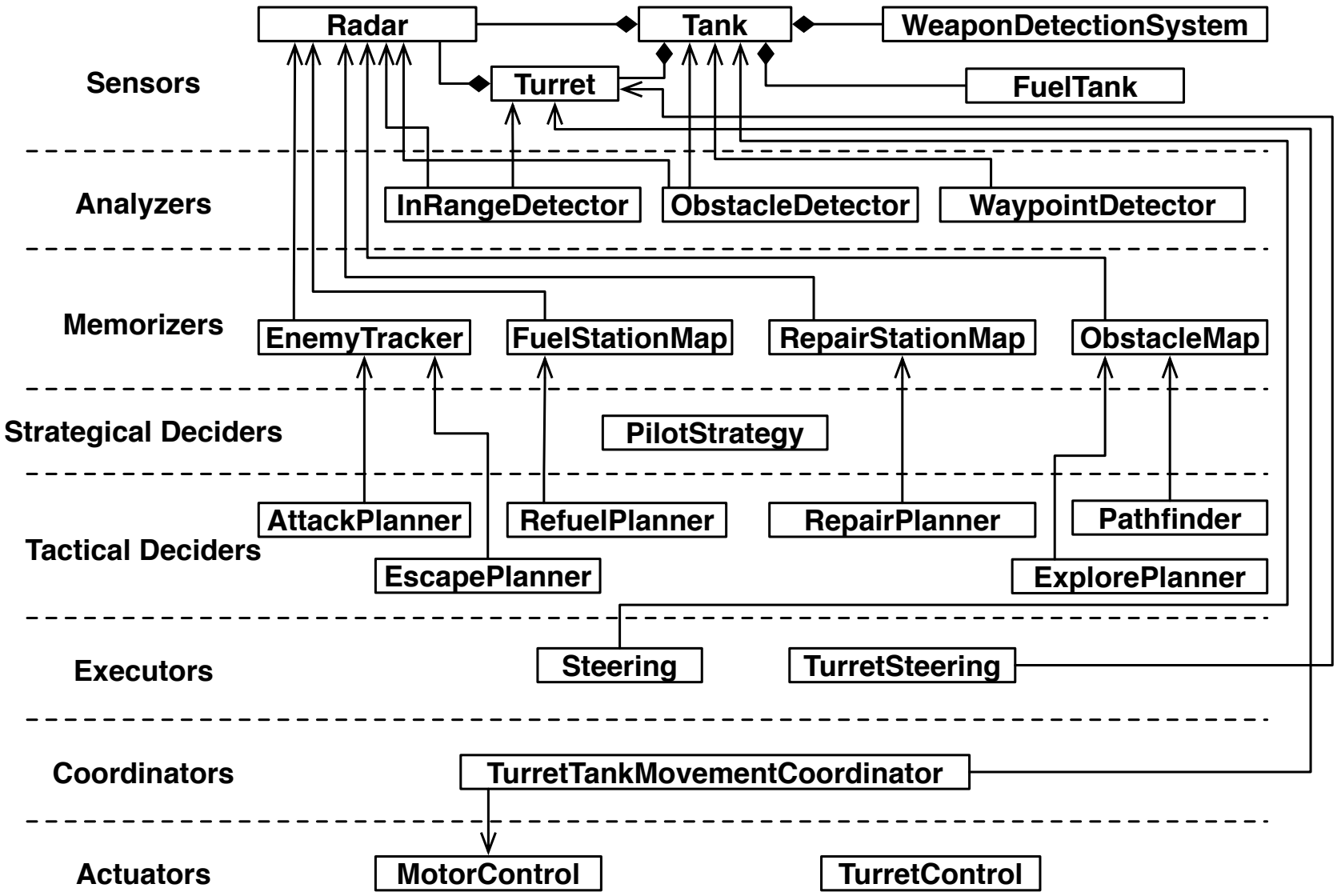


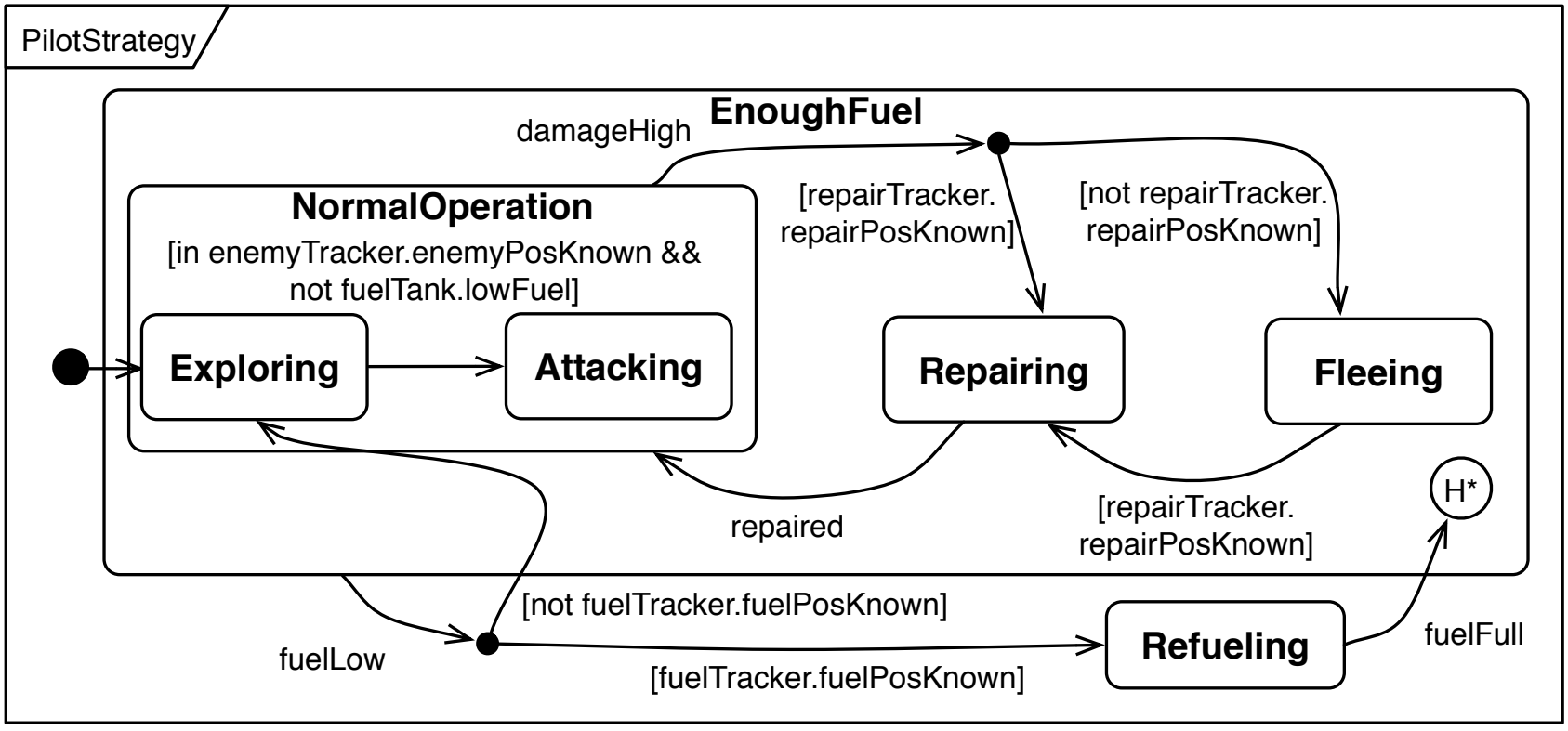
Most Appropriate Formalism (Minimizing Accidental Complexity)

www.planeshift.it

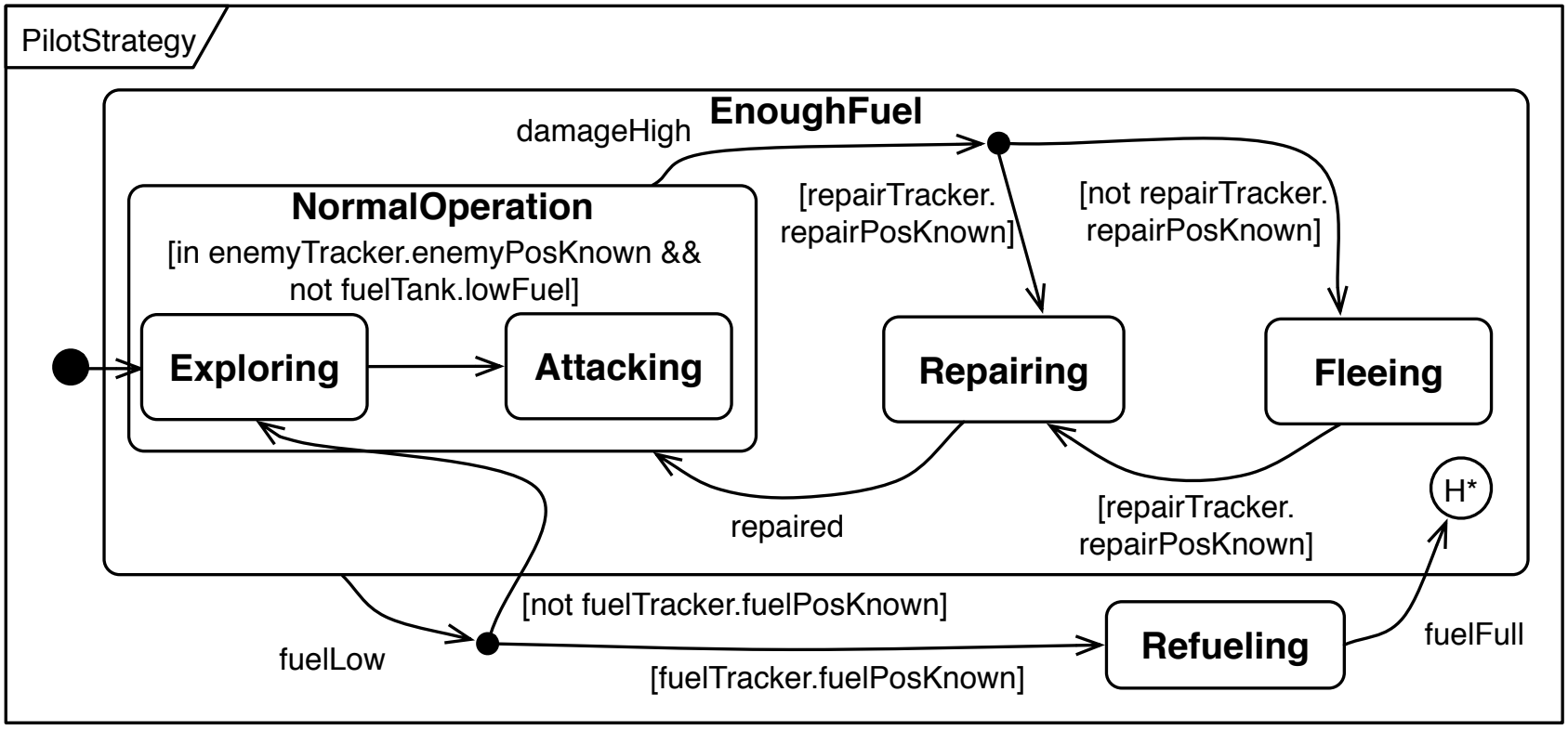
Massively Multiplayer Online Role Playing games need Non-Player Characters (NPCs)

TankWars: high level





Jörg Kienzle, Alexandre Denault, Hans Vangheluwe. Model-Based Design of Computer-Controlled Game Character Behavior. MoDELS 2007: 650-665

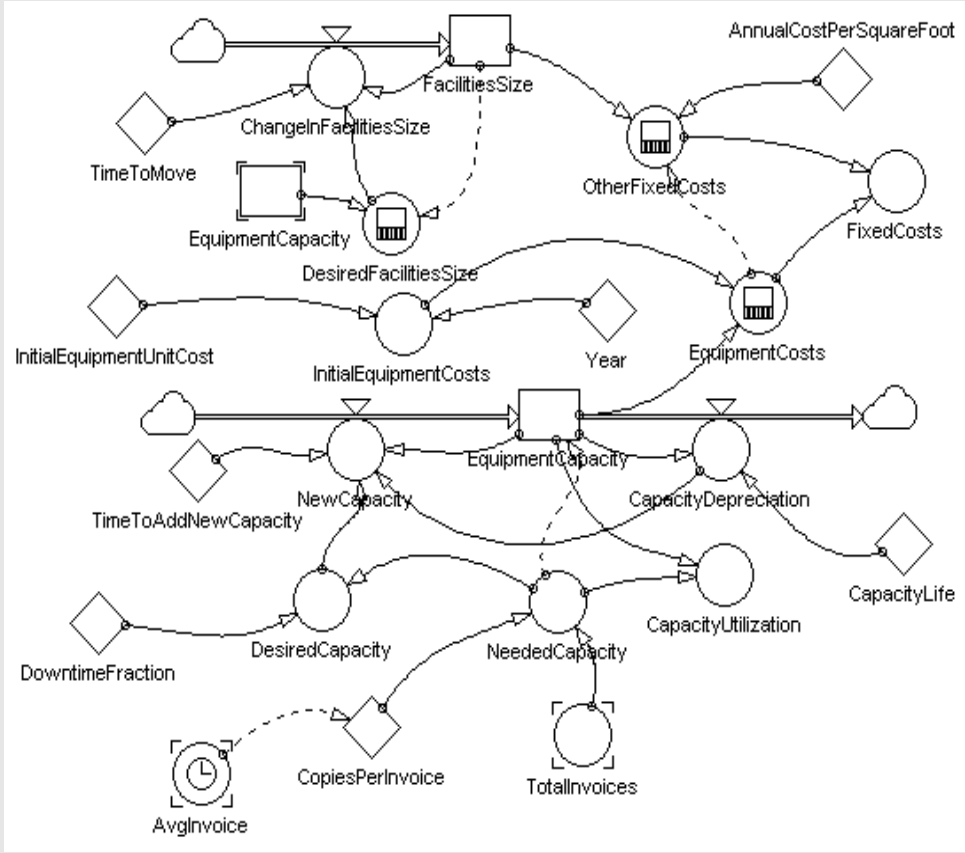


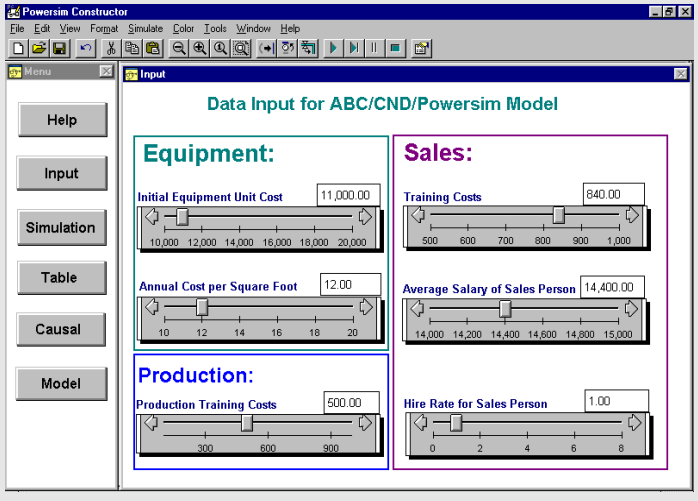
Jörg Kienzle, Alexandre Denault, Hans Vangheluwe. Model-Based Design of Computer-Controlled Game Character Behavior. MoDELS 2007: 650-665

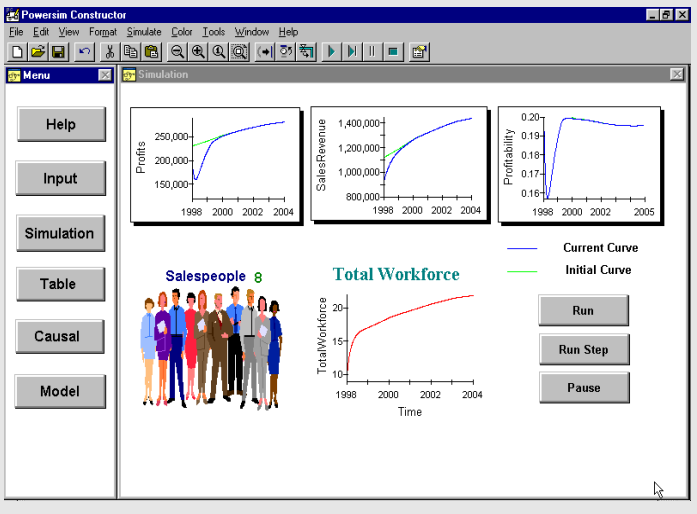
Could have used production rules instead of Statecharts

Eugene Syriani, Hans Vangheluwe: Programmed Graph Rewriting with DEVS. AGTIVE 2007: 136-151

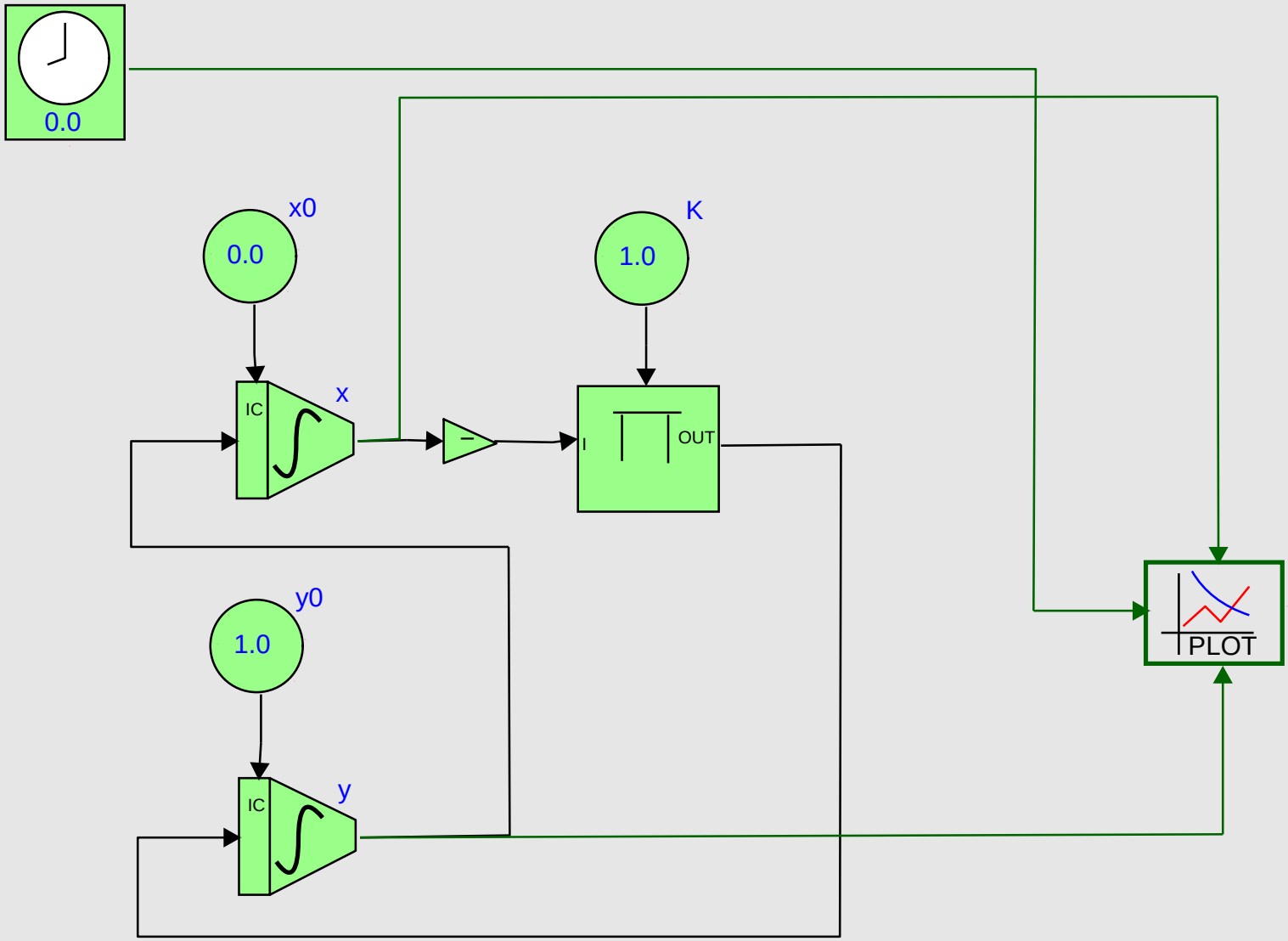
“Management Flight Simulator” using Forrester System Dynamics model



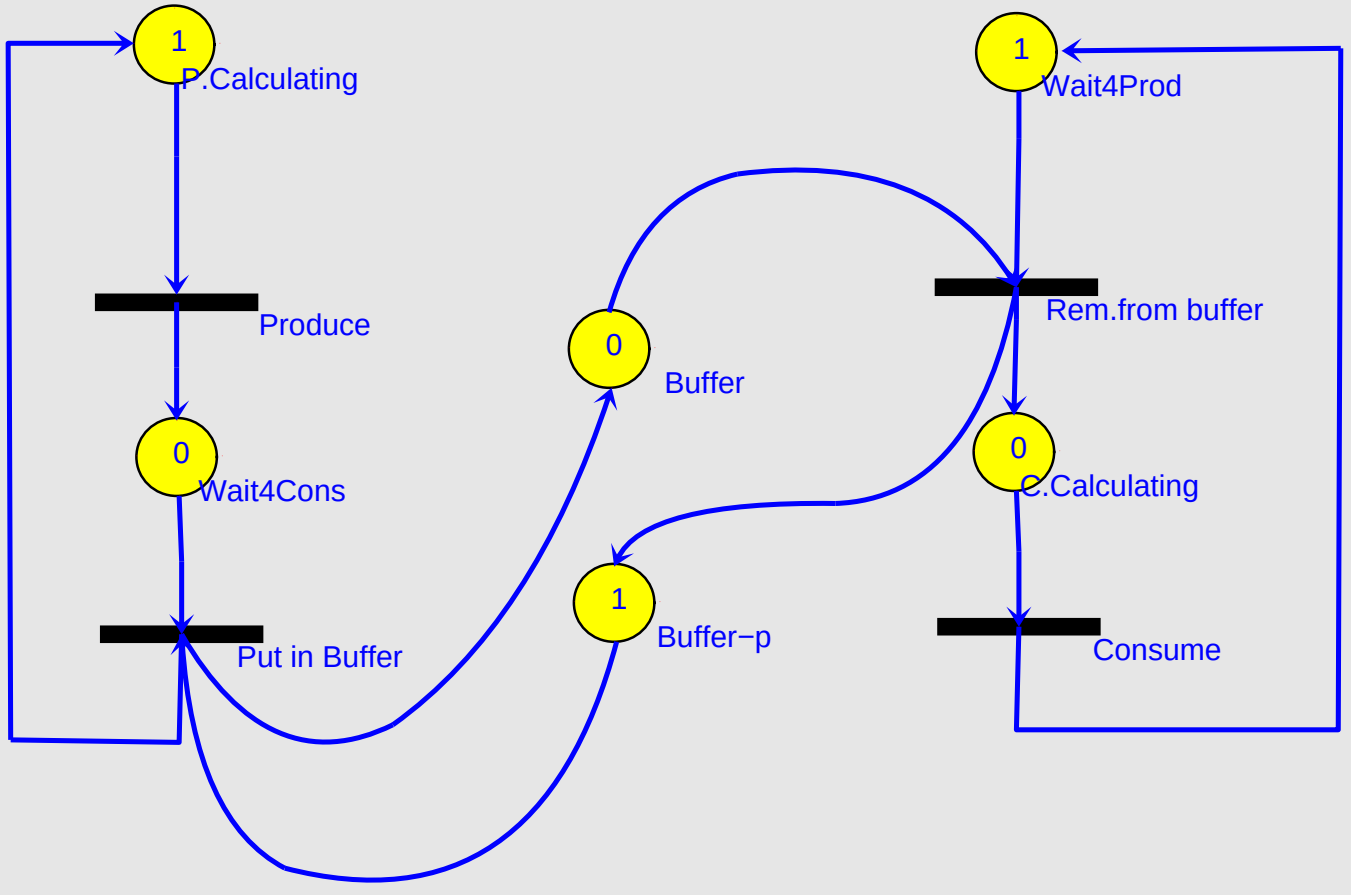




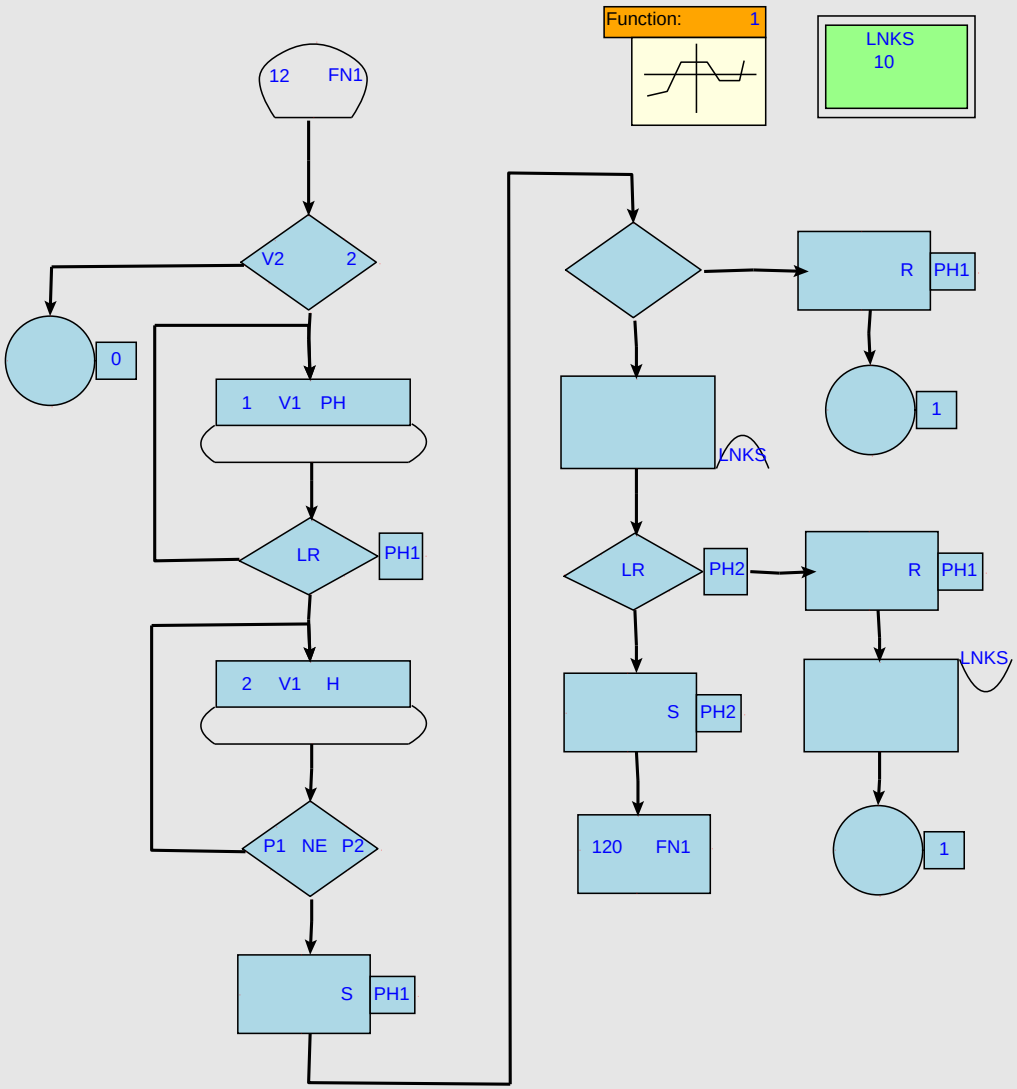
Causal Block Diagram model of Harmonic Oscillator



Petri Net model of Producer – Consumer

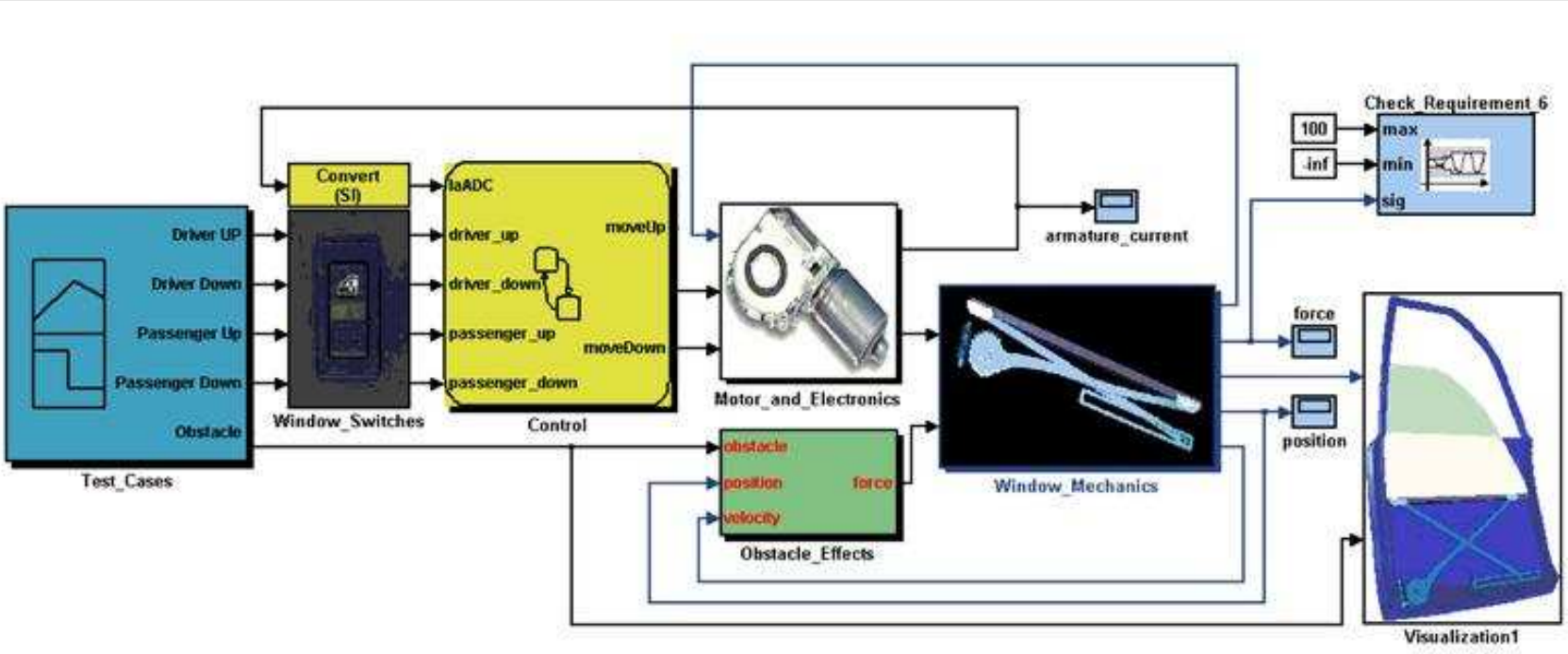


GPSS model of Telephone Exchange



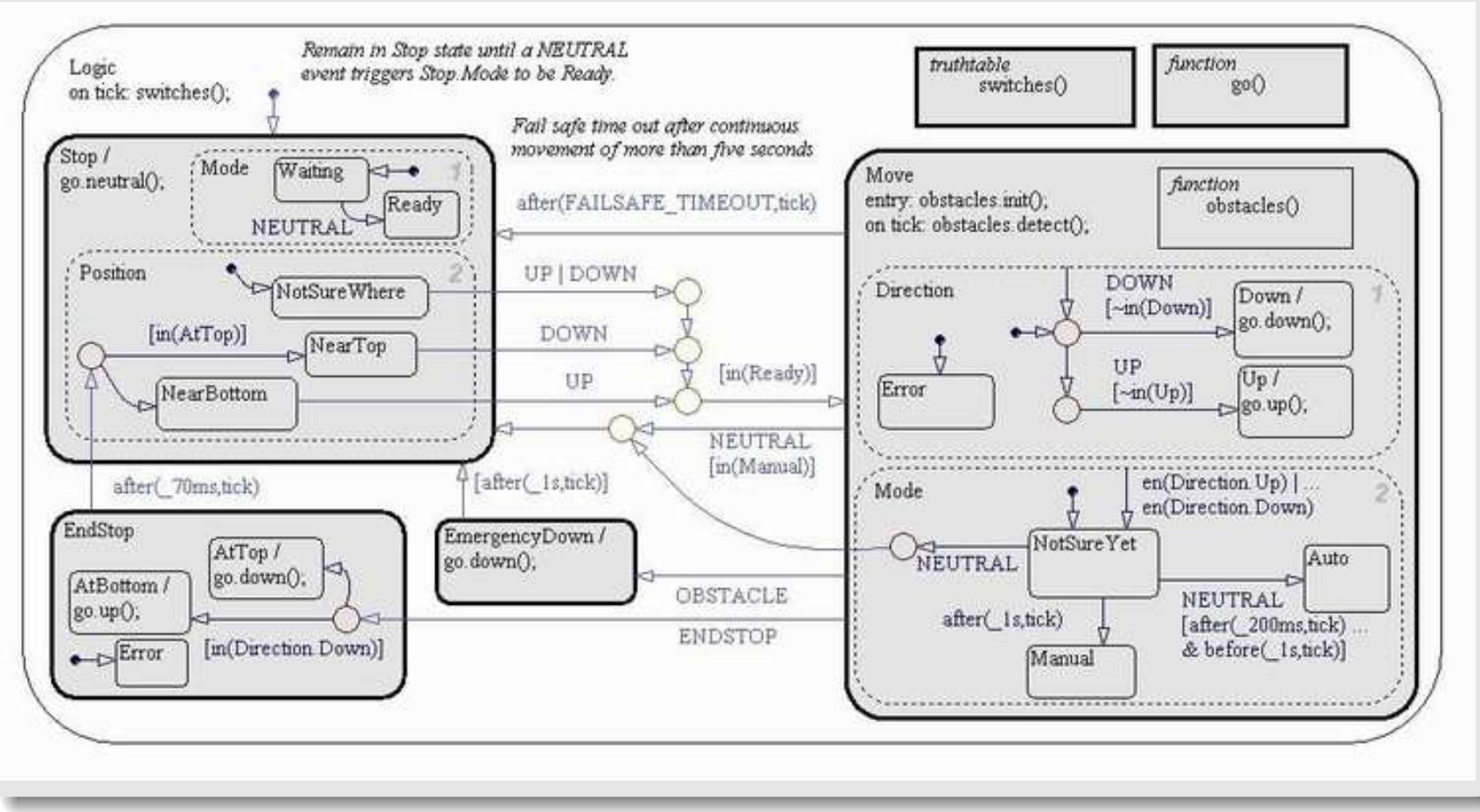
Multiple Formalisms: Power Window

Components in Different Formalisms

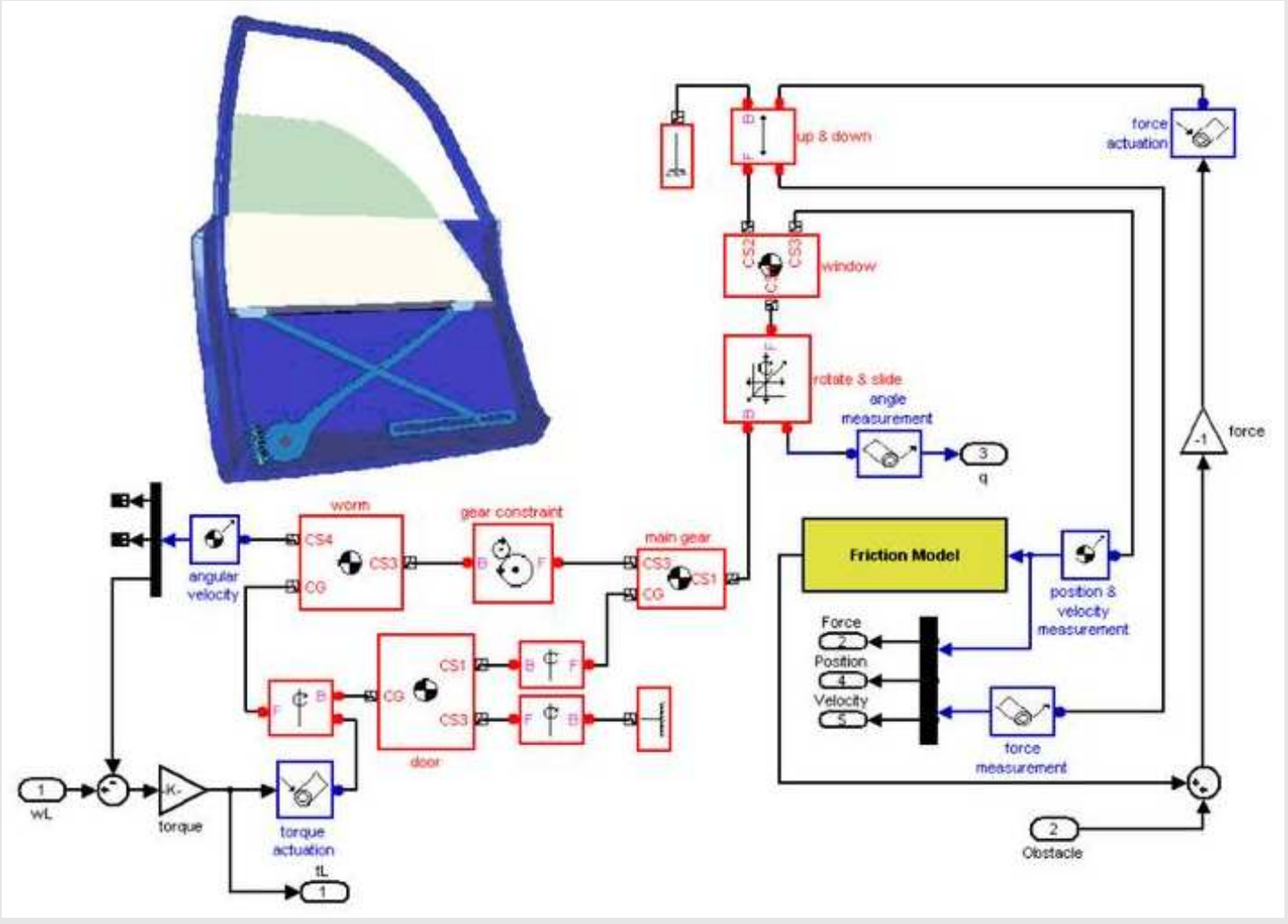


www.mathworks.com/products/demos/simulink/PowerWindow/html/PowerWindow1.html

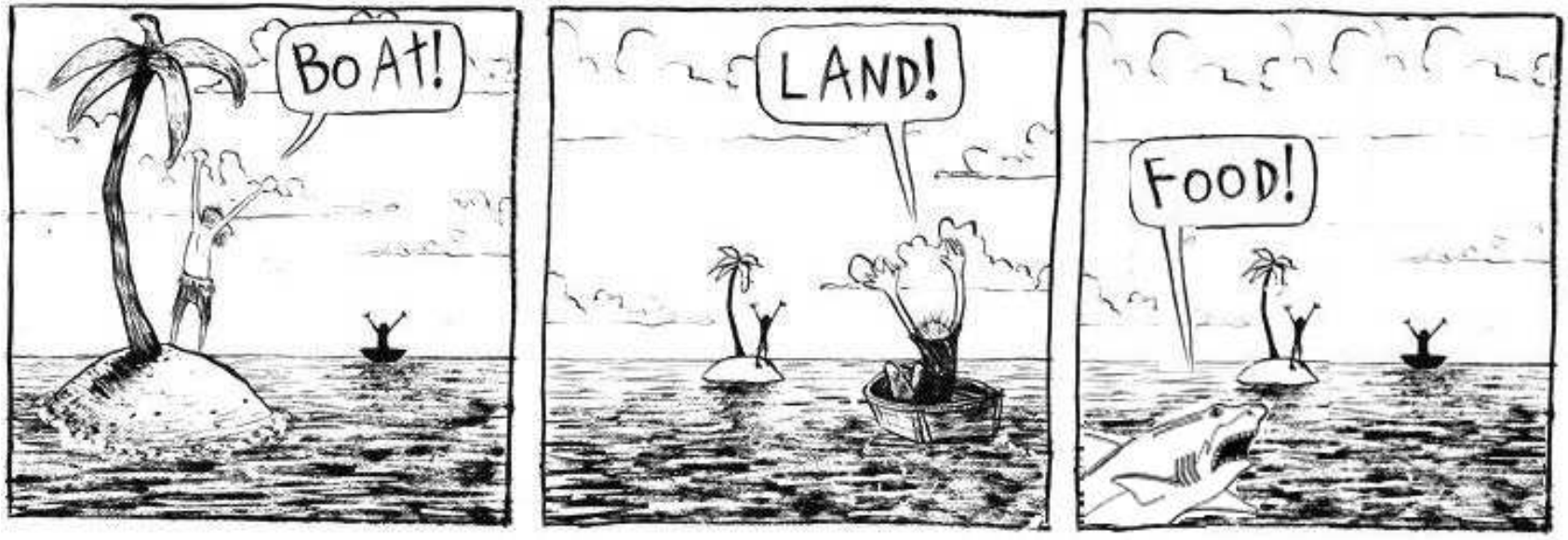
Controller, using Statechart(StateFlow) formalism



Mechanics subsystem

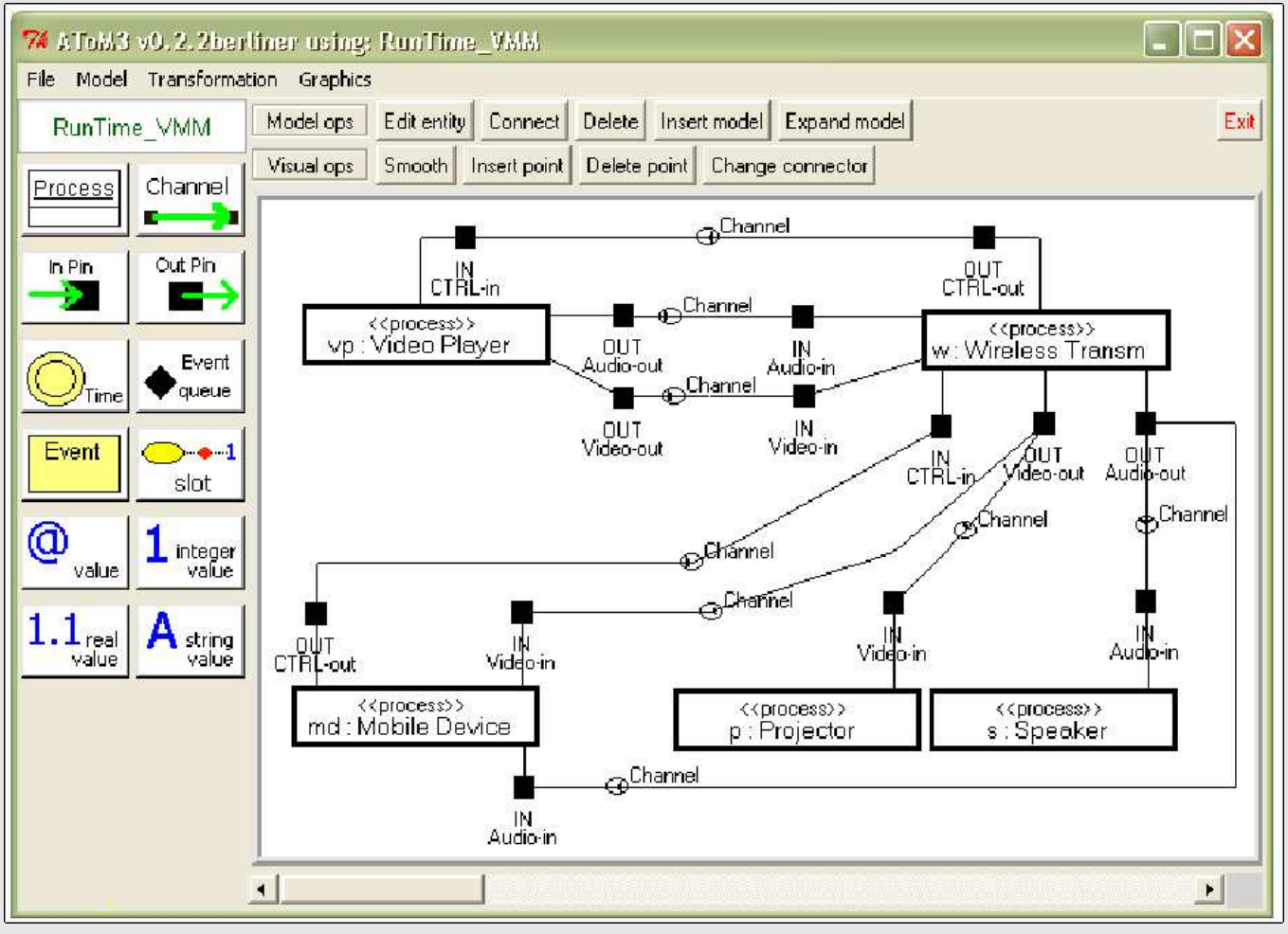


Multiple Views/Concerns/Aspects

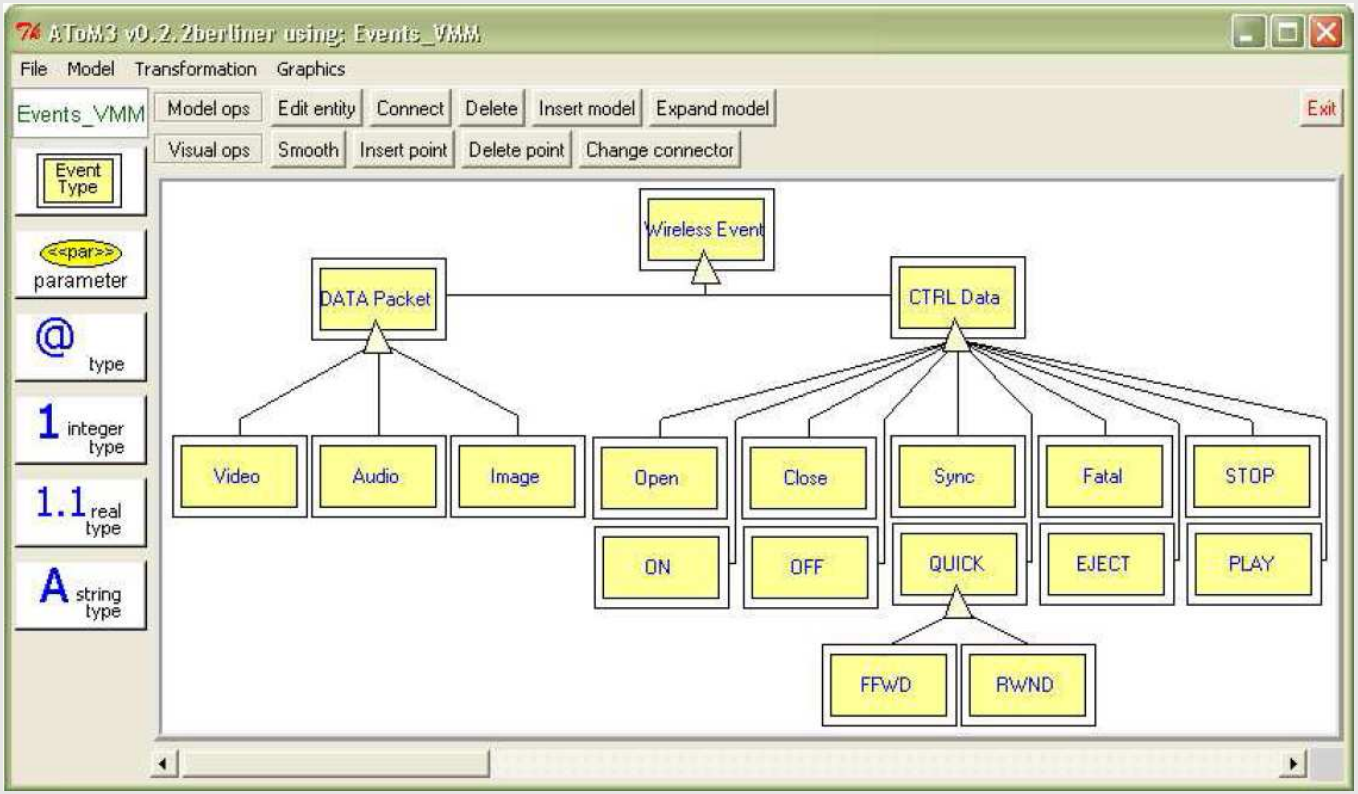


Wireless Home Entertainment System

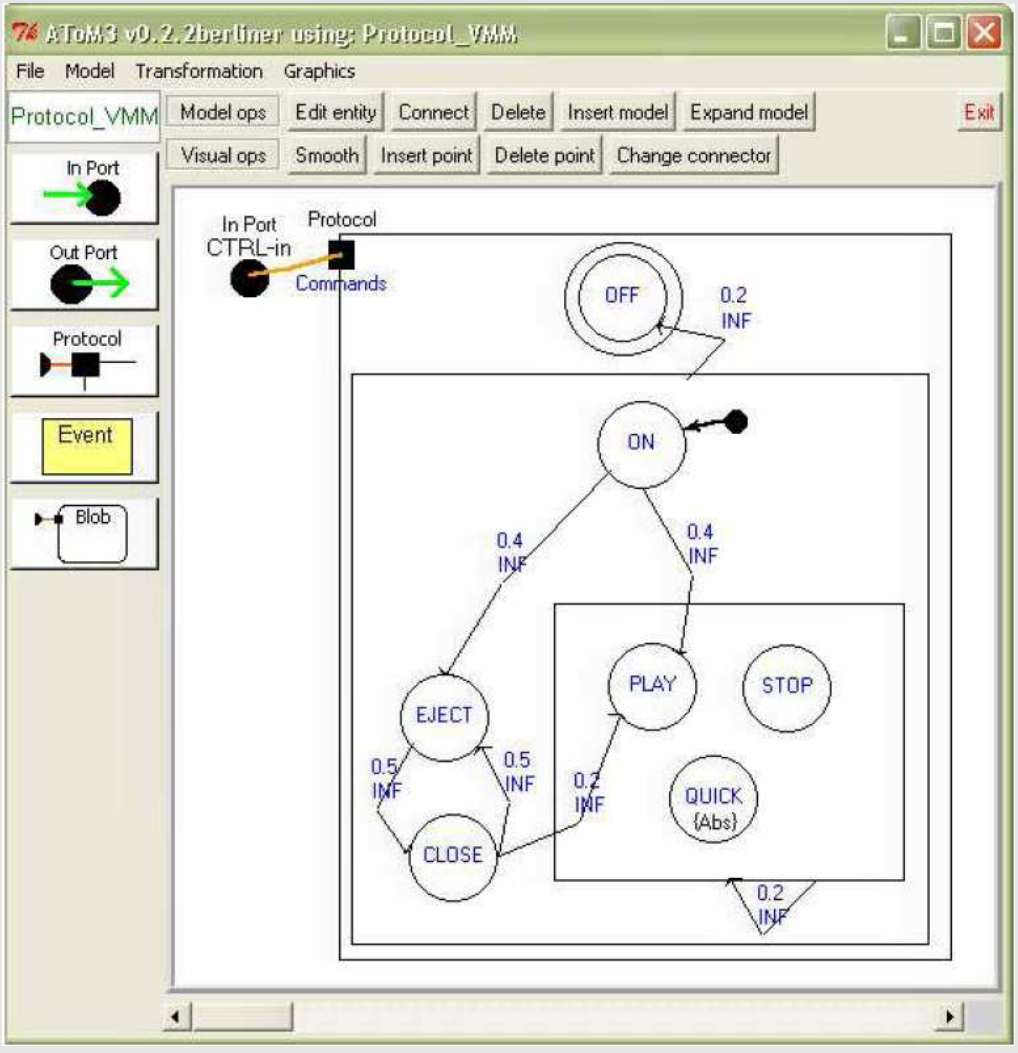
Multiple (consistent !) Views (in $\neq$ Formalisms)



View: Events Diagram



View: Protocol Statechart



No Free Lunch!

Solutions often introduce
their **own accidental complexity**

No Free Lunch!

Solutions often introduce
their **own accidental complexity**

- multiple abstraction levels (need **morphism**)

No Free Lunch!

Solutions often introduce
their **own accidental complexity**

- multiple abstraction levels (need **morphism**)
- optimal formalism (need **precise meaning**)

No Free Lunch!

Solutions often introduce
their **own accidental complexity**

- multiple abstraction levels (need **morphism**)
- optimal formalism (need **precise meaning**)
- multiple formalisms (need **relationship**)

No Free Lunch!

Solutions often introduce
their **own accidental complexity**

- multiple abstraction levels (need **morphism**)
- optimal formalism (need **precise meaning**)
- multiple formalisms (need **relationship**)
- multiple views (need **consistency**)

No Free Lunch!

Solutions often introduce
their **own accidental complexity**

- multiple abstraction levels (need **morphism**)
- optimal formalism (need **precise meaning**)
- multiple formalisms (need **relationship**)
- multiple views (need **consistency**)

Multi-Paradigm Modelling
(*model* everything, minimize *accidental complexity*)

- at the most appropriate **level of abstraction**
- using the most appropriate **formalism(s)**
  Class Diagrams, Differential Algebraic Equations, Petri Nets, Bond Graphs, Statecharts, CSP, Queueing Networks, Sequence Diagrams, Lustre/Esterel, ...
- with **transformations** as first-class models

Pieter J. Mosterman and Hans Vangheluwe.

Computer Automated Multi-Paradigm Modeling: An Introduction. Simulation 80(9):433–450, September 2004.

Special Issue: Grand Challenges for Modeling and Simulation.