Software Intensive Systems: Dealing with Complexity

Hans Vangheluwe
<table>
<thead>
<tr>
<th>Software?</th>
<th>Model Everything!</th>
<th>Compl. Causes</th>
<th>Dealing with Compl.</th>
<th>MPM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The image shows a detailed view of a semiconductor chip, indicating the complexity and precision of modern hardware components.
Software?

Model Everything!

Compl. Causes

Dealing with Compl.

MPM
Algorithms + Data Structures = Programs

Niklaus Wirth

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
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

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


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 Jetta's 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 Jetta's 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.

Copyright 2007 The Associated Press. All rights reserved. This material may not be published, broadcast, rewritten or redistributed.
Automotive MBSE

Function Model
- Discretion, Scaling, Augmentation, Embedding

Production Model
- Autocoding

Production Code

MiL-Test

RCP

SiL-Test

PiL/HiL-Test

Simulation System

Prototyping HW + Vehicle

Controller + Simulation of Environment

Controller + Vehicle

Integration

Simulation System

Controller + Vehicle
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
Simulation ... when too costly/dangerous

analysis ↔ 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

\[\text{theta} = 66.8\]

\[\text{theta} = 21.3\]
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
- ...

---

**Model Everything!**

**Compl. Causes**

**Dealing with Compl.**

**MPM**

---

**Software?**
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”**

![Diagram showing system components: Input, Controller, Plant, Output](image)

**“Plant”?!**

![Image of a chemical plant](image)
System vs. “Plant”

Number of Components
Crowds: diversity, interaction
Diversity of Components: Power Window
Non-compositional/Emergent Behaviour

non-compositionality of networks leads to emergent behaviour

separation  cohesion  alignment

www.red3d.com/cwr/boids/ (Craig Reynolds)
Emergent Behaviour
Engineered Emergent Behaviour

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


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

Levels of Abstraction/Views: Morphism

- Model
  - simulation
  - detailed
    - (technical) level
  - trajectory
  - traj_t
- abstraction
- M_t
- M_d
- abstraction
- M_d
- abstract
  - (decision) level
  - traj_d
**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 (Minimizing Accidental Complexity)
Most Appropriate Formalism (Minimizing Accidental Complexity)

www.planeshift.it

Massively Multiplayer Online Role Playing games need Non-Player Characters (NPCs)
TankWars: high level

Sensors
- Radar
- Turret
- FuelTank

Analyzers
- InRangeDetector
- ObstacleDetector
- WaypointDetector

Memorizers
- EnemyTracker
- FuelStationMap
- RepairStationMap
- ObstacleMap

Strategical Deciders
- PilotStrategy

Tactical Deciders
- AttackPlanner
- RefuelPlanner
- RepairPlanner
- ExplorePlanner
- EscapePlanner

Executors
- Steering
- TurretSteering

Coordinators
- TurretTankMovementCoordinator

Actuators
- MotorControl
- TurretControl

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

- P.Calculating
- Wait4Cons
- Produce
- Put in Buffer
- Buffer
- Buffer−p
- C.Calculating
- Wait4Prod
- Rem.from buffer
- Consume
GPSS model of Telephone Exchange
Multiple Formalisms: Power Window
Components in Different Formalisms

Controller, using Statechart(StateFlow) formalism
Mechanics subsystem
Multiple Views/Concerns/Aspects
Wireless Home Entertainment System
Multiple (consistent !) Views (in ≠ 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**)

**Multiple Views/Concerns/Aspects**
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.


Special Issue: Grand Challenges for Modeling and Simulation.