Embedding Formal Methods into Systems Engineering

Helmut Veith
FORSYTE Group, TU Darmstadt
School of Computer Science, Carnegie Mellon University

for(syte) TECHNISCHE UNIVERSITÄT DARMSTADT Carnegie Mellon

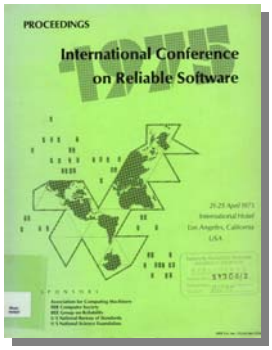
1

TECHNISCHE UNIVERSITÄT DARMSTADT for(syte)

Weight ?
Mechanical engineering process ?
Repair logistics ?
ECU distribution ?
Component Reuse ?
Deployment strategy ?
Intellectual Property ?
Certification ?
Product Lines ?
Power consumption ?
Variant Management
Execution Time ?
Physical Environme
Memory use ?
Obsoliscence ?
Safety ?
Dependability ?
Cost per car ?
HW or SW ?
FPGA or ASIC ?
Multicore ?

Experience from Collaborations with Automotive / Avionic Industry

H. Veith 5/2008



Correctness Concerns and, Among Other Things, Why They are Resented.
E. Dijkstra

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

… a clear separation of concerns emerges: we might call them the mathematical concerns about correctness and the engineering concerns about efficiency.

... the discovery how to separate them rigorously in our thinking is relatively young, and even when aware of this possibility, we often fall back into our old bad habits.

H. Veith 5/2008

Abstraction

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction

high level programming language

↓ compilation

executable

↓ execution

hardware

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction

high level programming language

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction

high level programming language

↓ compilation

executable

↓ execution

hardware

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction

math. model

synthesis

high level programming language

compilation

executable

execution

hardware

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction

math. model

synthesis

high level programming language

compilation

executable

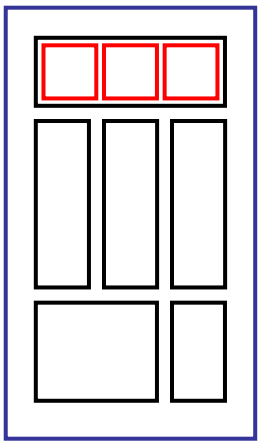
execution

hardware

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction



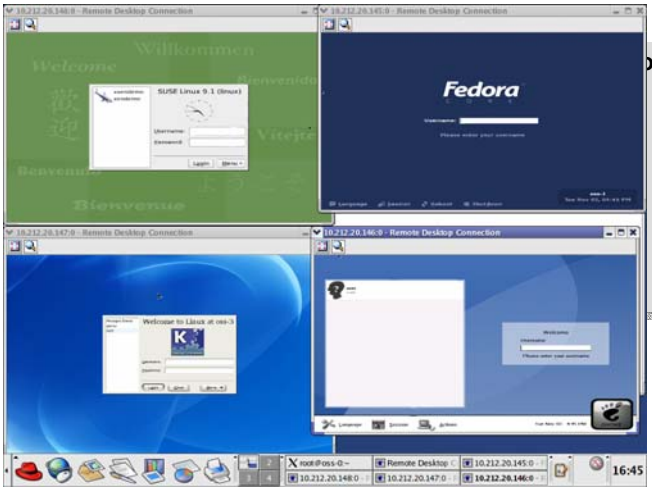
Hierarchy and Encapsulation

- structured programming
- object orientation
- layered architecture
- interfaces
- middleware
- virtual machines
- virtualization

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

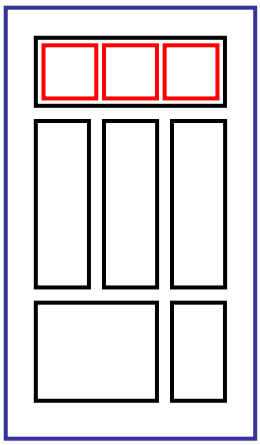
Abstraction



The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction



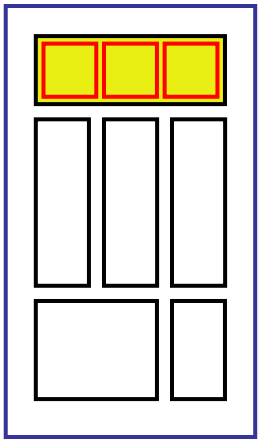
Hierarchy and Encapsulation

- structured programming
- object orientation
- layered architecture
- interfaces
- middleware
- virtual machines
- virtualization

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction



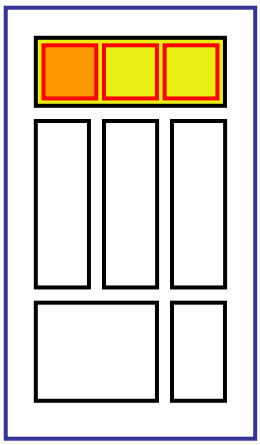
Hierarchy and Encapsulation

- structured programming
- object orientation
- layered architecture
- interfaces
- middleware
- virtual machines
- virtualization

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction



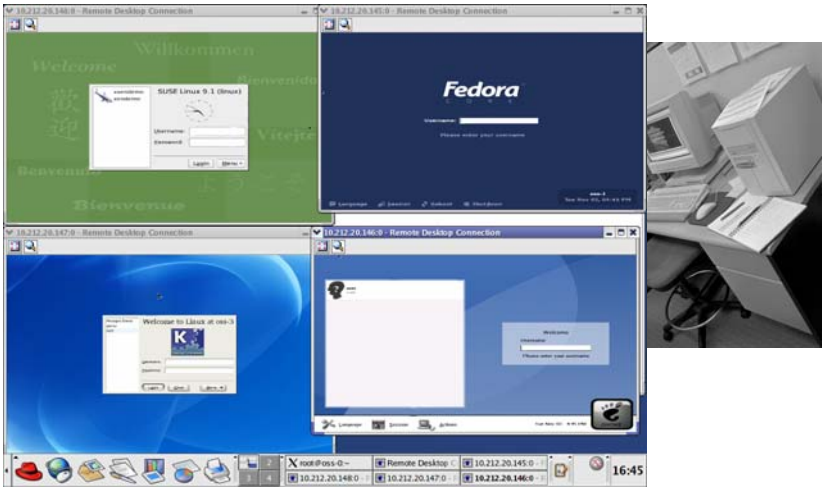
Hierarchy and Encapsulation

- structured programming
- object orientation
- layered architecture
- interfaces
- middleware
- virtual machines
- virtualization

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

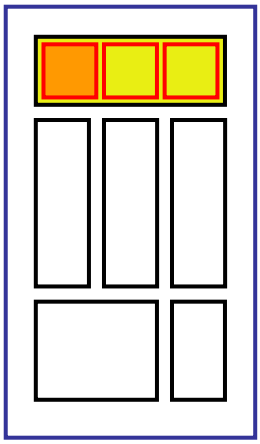
Abstraction



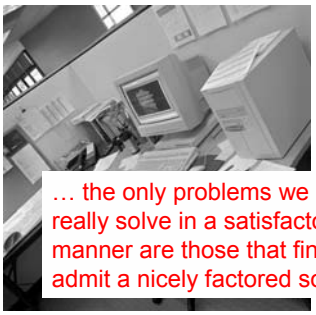
The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction



Java
TCP/IP
OCaml
Unix
CORBA
SQL
SAP
BEPL
XEN
…

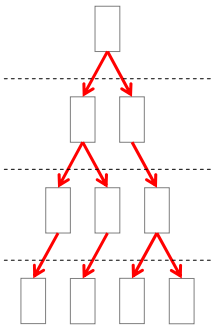


The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

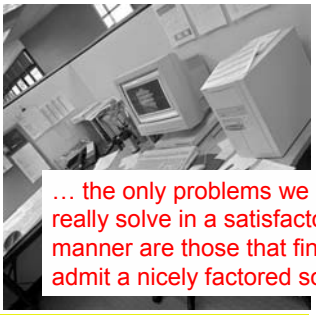
H. Veith 5/2008

Abstraction

Hierarchy / Layers



Java
TCP/IP
OCaml
Unix
CORBA
SQL
SAP
BEPL
XEN


… the only problems we can
really solve in a satisfactory
manner are those that finally
admit a nicely factored solution.

Feature Driven Development
- functional requirements
- top down / bottom up implementation
- code topology reflects features

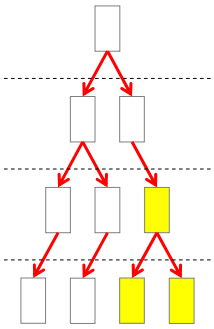
The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

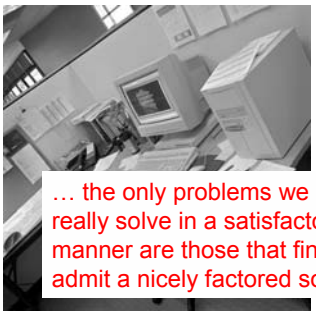
Abstraction

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

Hierarchy / Layers



Java
TCP/IP
OCaml
Unix
CORBA
SQL
SAP
BEPL
XEN
...



… the only problems we can really solve in a satisfactory manner are those that finally admit a nicely factored solution.

Feature Driven Development
- functional requirements
- top down / bottom up implementation
- code topology reflects features

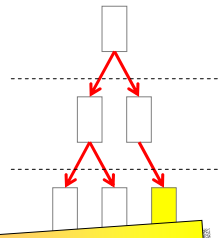
The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Abstraction

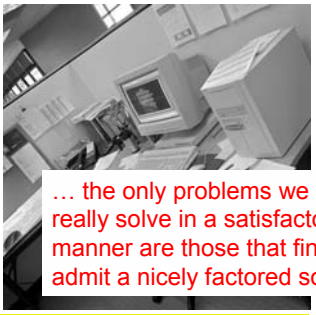
TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

Hierarchy / Layers



Divide and Conquer,
Encapsulation,
Components …

Java
TCP/IP
OCaml
Unix
CORBA
SQL
SAP
BEPL
XEN
...



… the only problems we can really solve in a satisfactory manner are those that finally admit a nicely factored solution.

Feature Driven Development
- functional requirements
- top down / bottom up implementation
- code topology reflects features

The purpose of abstraction is *not* to be vague, but to create a new semantic level in which one can be absolutely precise. [*Turing Award Lecture*]

H. Veith 5/2008

Compilers: A Microcosm of Model Based Design

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

high level program

executable

abstract model

… with mathematical precision.

concretization

Compiler
… analyzes (one) input program
… recognizes certain program errrors
… resolves language nondeterminism
… knows target hardware
… performs optimizations
… is trusted to be *correct*

Formal Methods

◄ Model Analysis / Validation

◄ Model Transformation

◄ Model Optimization

H. Veith 5/2008

Compilers: A Microcosm of Model Based Design

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

constraints

… -O1 -O2 -O3 -Os

… with mathematical precision.

solution

Compiler
… analyzes (one) input program
… recognizes certain program errrors
… resolves language nondeterminism
… knows target hardware
… performs optimizations
… is trusted to be *correct*

Formal Methods

◄ Model Analysis / Validation

◄ Model Transformation

◄ Model Optimization

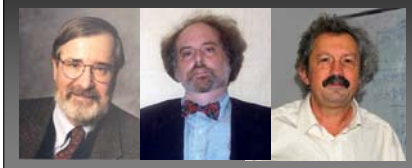
H. Veith 5/2008

Formal Methods

Computer-aided mathematically precise reasoning methodologies about systems (and) models.

SAT and SMT Solving, Model Checking, Synthesis, Testing, Theorem Proving, …

A. M. Turing Award 2007 for Model Checking



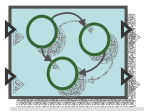
Clarke Emerson Sifakis

Formulas have always frightened me.

At the same time, I have a warm appreciation for well-designed formalisms that enable us to do things that I couldn't possibly do without them.

[*My Hopes of Computing Science*]

Model Checking



critical property

↓ model checking

program

↓ compilation

executable

code assertions
absence of deadlocks
termination
correct API use
path feasibility
memory violations
safety & liveness …

Property violations documented by program traces ! I know a bug when I see it.

Model Checking

critical property

↓ model checking

program

↓ compilation

executable

Property violations documented by program traces ! I know a bug when I see it.

H. Veith 5/2008

Formal Methods Ctd.

compiler	*correctly typed ?*
SAT / SMT decision proc.	*test case feasible ?* *deployment possible ?*
testing	*runtime errors ? coverage ?*

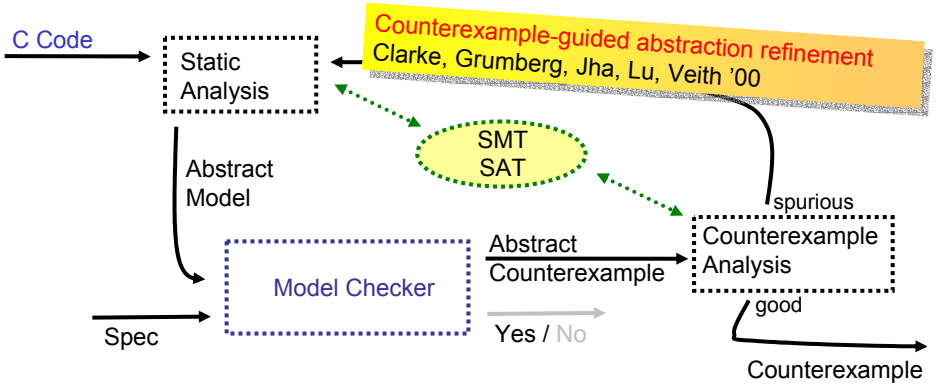
theorem proving (ISABEL, ACL2, PVS)

"*requires specialized expertise that precludes its broad use in industry*"

(Bob Kurshan, Cadence)

H. Veith 5/2008

Software Model Checking



- 2000s: development of industrial strength C model checkers
- "… rivals theorem proving for many verification tasks" (Rushby)
- →Microsoft product for Windows device driver verification

H. Veith 5/2008

Software Engineering

(i) separation of correctness and efficiency
(ii) abstraction and hierarchy

H. Veith 5/2008

Systems Engineering

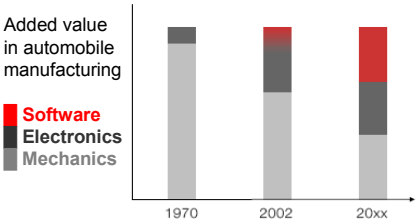
TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

Mass Market

70 ECUs
Inside

Multidisciplinary Effort

- ► Mechanical Engineering
- ► Electrical Engineering
- ► Computer Science



H. Veith 5/2008

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

Weight ?
Repair logistics ?
Component Reuse ?
Intellectual Propert
Product Lines ?
Variant Management
Physical Environment ?
Obsoliscence ?
Dependability ?

Mechanical engineering process ?
ECU distribution ?
Deployment strategy ?
Certification ?
Power consumption ?
Execution Time ?
Memory use ?
Safety ?
Cost per car ?

HW or SW ?
FPGA or ASIC ?
Multicore ?

Computer-aided Car Compilation ?

H. Veith 5/2008

System Design Challenge

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

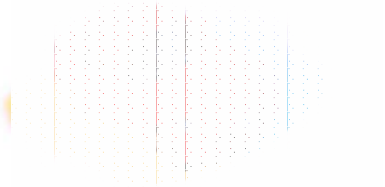
Constraints

- Software: algorithms, protocols, architectures, compilers, …
- Hardware: speed, memory, p
- Environment: performance, robustness, …

informatics + physics = emerging theory
Embedded Systems Design Challenge

integration / solution / compilation

SYSTEM BLUEPRINT



H. Veith 5/2008

System Design Challenge

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

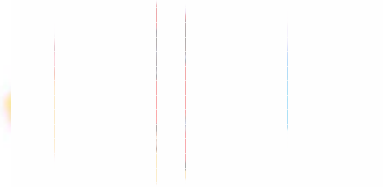
Constraints

- Software: algorithms, protocols, architectures, compilers, …
- Hardware: speed, memory, p
- Environment: performance, robustness, …

informatics + physics = emerging theory
Embedded Systems Design Challenge

integration / solution / compilation

SYSTEM BLUEPRINT



H. Veith 5/2008

System Design Challenge

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

Constraints

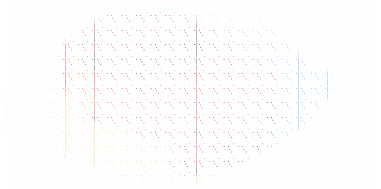
- Software: algorithms, protocols, architectures, compilers, …
- Hardware: speed, memory, p
- Environment: performance, robustness, …
- Cost: development, HW, safe warranty, testing, …
- Time: time to market, produc

informatics + physics = emerging theory
Embedded Systems Design Challenge

business + law + culture
automotive, avionic, …

integration / solution / compilation

SYSTEM BLUEPRINT



H. Veith 5/2008

System Design Challenge

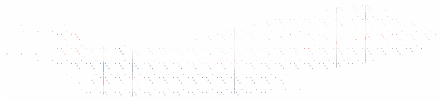
TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

Constraints

- Software: algorithms, protocols, architectures, compilers, …
- Hardware: speed, memory, power, failure rates, weight, size…
- Environment: performance, robustness, …
- Cost: development, HW, safety, certification, warranty, testing, …
- Time: time to market, product lines, variants, component reuse …

integration / solution / compilation

SYSTEM BLUEPRINT **ARINC / AADL**



H. Veith 5/2008

System Design Challenge

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

Constraints

- Software: algorithms, protocols, architectures, compilers, …
- Hardware: speed, memory, power, failure rates, weight, size…
- Environment: performance, robustness, …
- Cost: development, HW, safety, certification, warranty, testing, …
- Time: time to market, product lines, variants, component reuse …

integration / solution / compilation

SYSTEM BLUEPRINT

H. Veith 5/2008

System Design Challenge

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

Constraints

- Software: algorithms, protocols, architectures, compilers, …
- Hardware: speed, memory, power, failure rates, weight, size…
- Environment: performance, robustness, …
- Cost: development, HW, safety, certification, warranty, testing, …
- Time: time to market, product lines, variants, component reuse …

integration / solution / compilation

SYSTEM BLUEPRINT

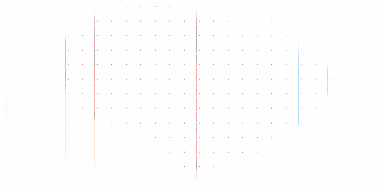
H. Veith 5/2008

System Design Challenge

Constraints

integration / solution / compilation

SYSTEM BLUEPRINT

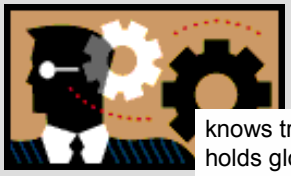


H. Veith 5/2008

System Design Challenge

Constraints

heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

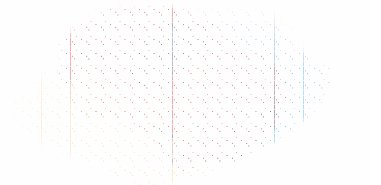


Guru Principle

knows trade-offs
holds global view
maintains IP

integration / solution / compilation

SYSTEM BLUEPRINT



H. Veith 5/2008

System Design Challenge

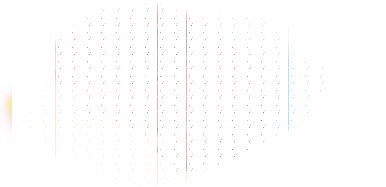
Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

Multiple models & disciplines
Transition between models
Model validation and integration

Multiple orthogonal views
Exceeding complexity

integration / solution / compilat

SYSTEM BLUEPRINT



H. Veith 5/2008

System Design Challenge

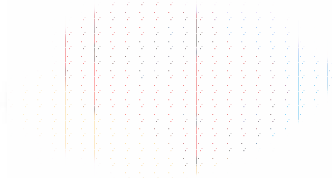
Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

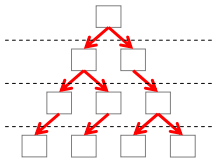
Multiple models & disciplines
Transition between models
Model validation and integration

Multiple orthogonal views
Exceeding complexity

integration / solution / compilat

SYSTEM BLUEPRINT





H. Veith 5/2008

System Design Challenge

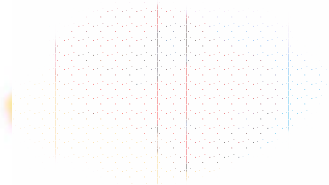
Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

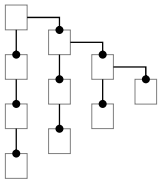
Multiple models & disciplines
Transition between models
Model validation and integration

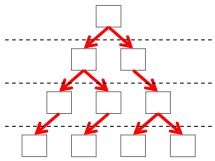
Multiple orthogonal views
Exceeding complexity

integration / solution / compilat

SYSTEM BLUEPRINT







H. Veith 5/2008

System Design Challenge

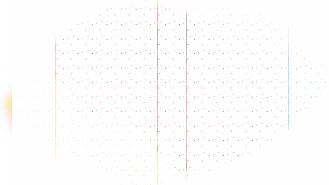
Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

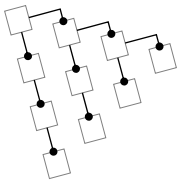
Multiple models & disciplines
Transition between models
Model validation and integration

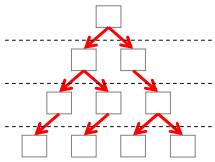
Multiple orthogonal views
Exceeding complexity

integration / solution / compilat

SYSTEM BLUEPRINT







H. Veith 5/2008

System Design Challenge

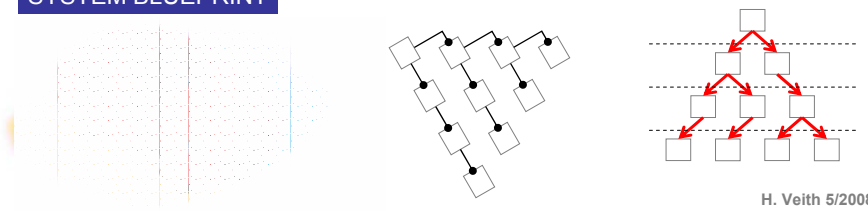
Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

Multiple models & disciplines
Transition between models
Model validation and integration

Multiple orthogonal views
Exceeding complexity

integration / solution / compilat

SYSTEM BLUEPRINT



H. Veith 5/2008

System Design Challenge

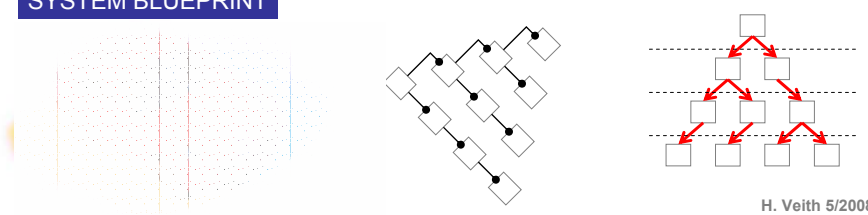
Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

Multiple models & disciplines
Transition between models
Model validation and integration

Multiple orthogonal views
Exceeding complexity

integration / solution / compilat

SYSTEM BLUEPRINT



H. Veith 5/2008

System Design Challenge

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

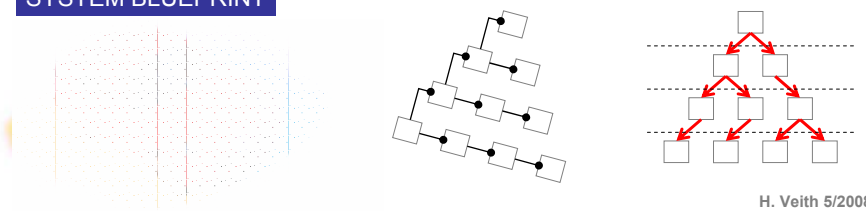
Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

Multiple models & disciplines
Transition between models
Model validation and integration

Multiple orthogonal views
Exceeding complexity

integration / solution / compilat

SYSTEM BLUEPRINT



H. Veith 5/2008

System Design Challenge

TECHNISCHE UNIVERSITÄT DARMSTADT
for(syte)

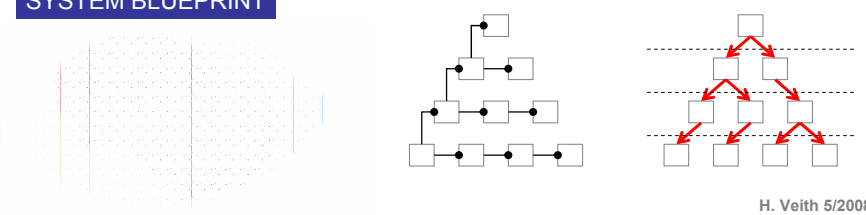
Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

Multiple models & disciplines
Transition between models
Model validation and integration

Multiple orthogonal views
Exceeding complexity

integration / solution / compilat

SYSTEM BLUEPRINT



H. Veith 5/2008

System Design Challenge

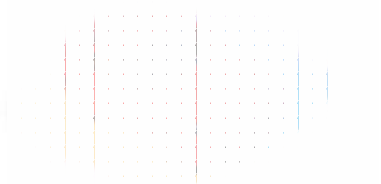
Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

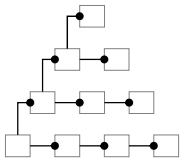
Multiple models & disciplines
Transition between models
Model validation and integration

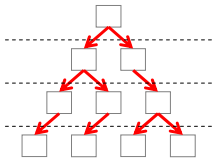
Multiple orthogonal views
Exceeding complexity
Formal methods needed

integration / solution / compilat

SYSTEM BLUEPRINT







H. Veith 5/2008

System Design Challenge

Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

Multiple models & disciplines
Transition between models
Model validation and integration

Multiple orthogonal views
Exceeding complexity
Formal methods needed

H. Veith 5/2008

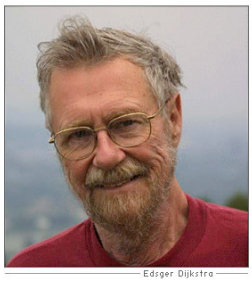
System Design Challenge

Constraints
heterogeneous
global / horizontal
analytical + discrete
transcend design layers
multiple optimization criteria

Multiple models & disciplines
Transition between models
Model validation and integration

Multiple orthogonal views
Exceeding complexity
Formal methods needed
Powerful architectures needed

Time-Triggered Architecture

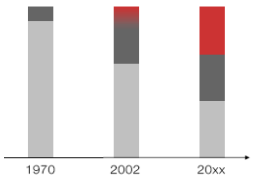

Edsger Dijkstra

"The clever programmer is fully aware of the strictly limited size of his own skull; therefore, he approaches the programming task in full humility, and among other things he avoids clever tricks like the plague."

H. Veith 5/2008

Case Study
Automotive Industry

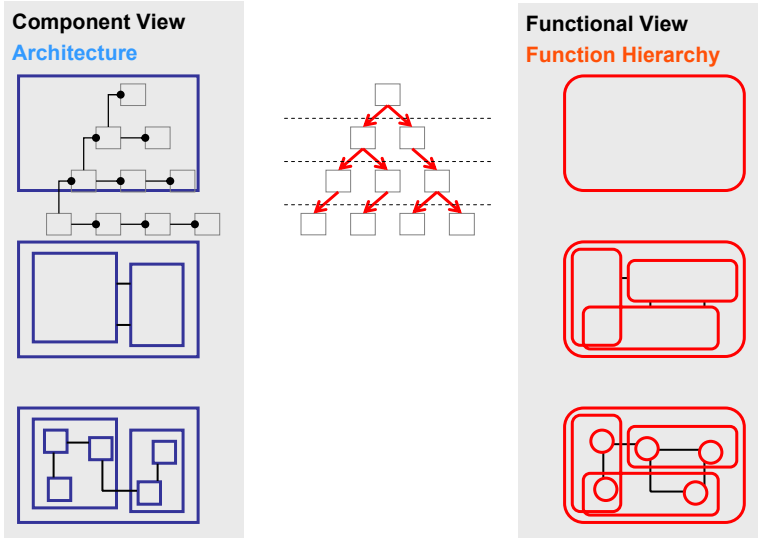
- Mechanical engineering culture
- Software recognized as strategic IP
- No software engineering process
- Jungle of models and ontologies
- Excessive warranty costs necessitate change



Project Goals
- Domain exploration and assessment
- Seamless modeling framework
- Prototype automotive tool chain
- Accompanying software engineering process
- Technology proof of concept

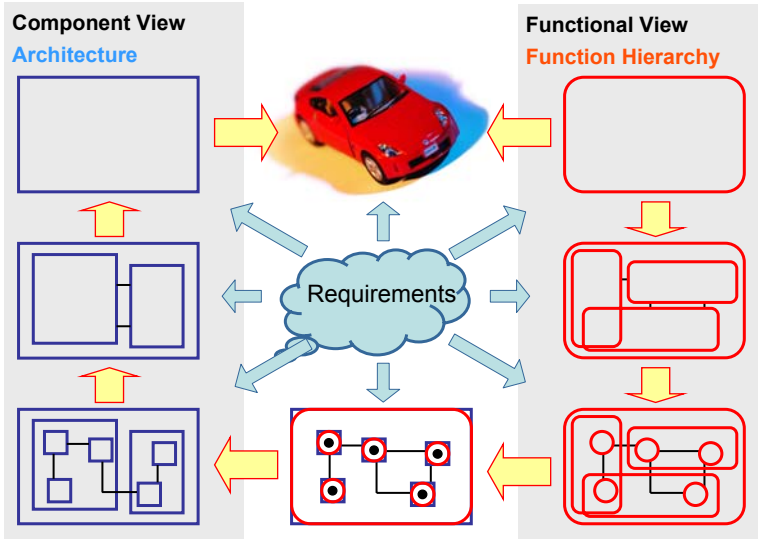
H. Veith 5/2008

Seamless Modeling



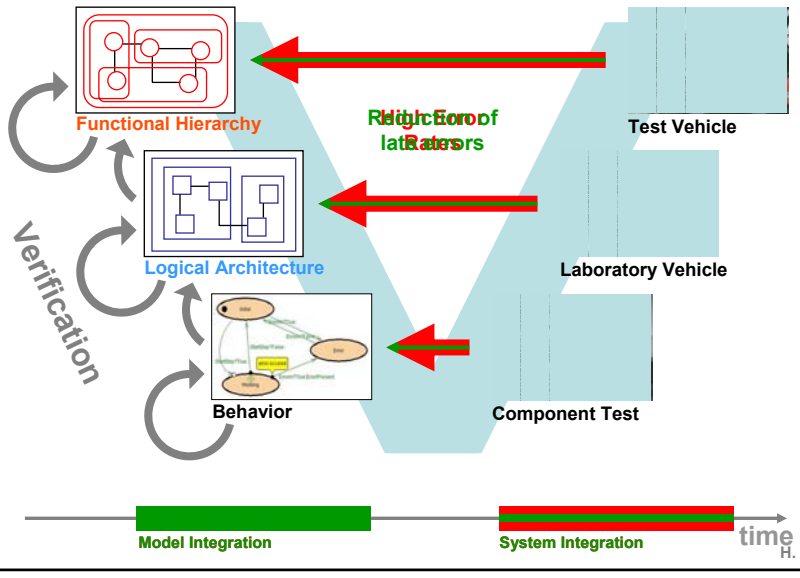
H. Veith 5/2008

Seamless Modeling



H. Veith 5/2008

Model Integration

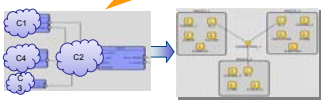


H. Veith 5/2008

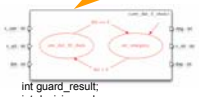
Automotive Tool Chain

COLA model:

Model-level verification, testing, simulation

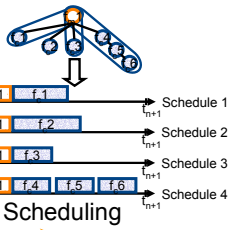


Clustered COLA model

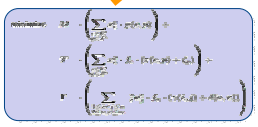


```
int guard_result;
int decision_val;
switch(unit_state->atm_state) {
        case 0:
emergency_guard200747(s_user_i
        n_0, s_act_in_1, dist_in_2,
        &guard_result);
    if(guard_result) {
        unit_state->atm_state =
1;
        ...
```

Generated C code for each cluster



Scheduling



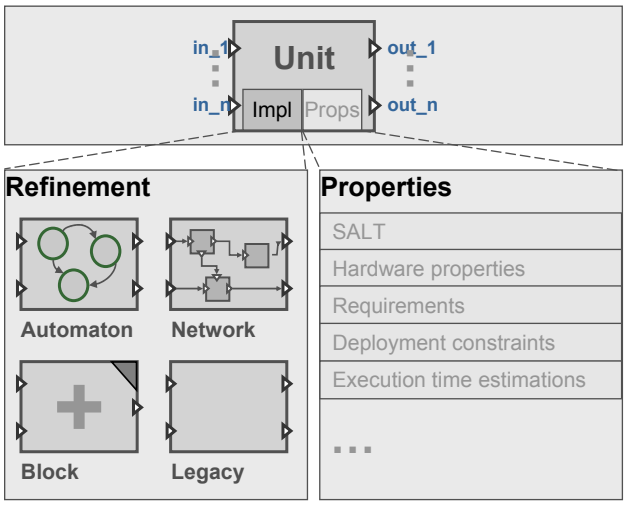
Allocation

SYSTEMC

Execution time estimation

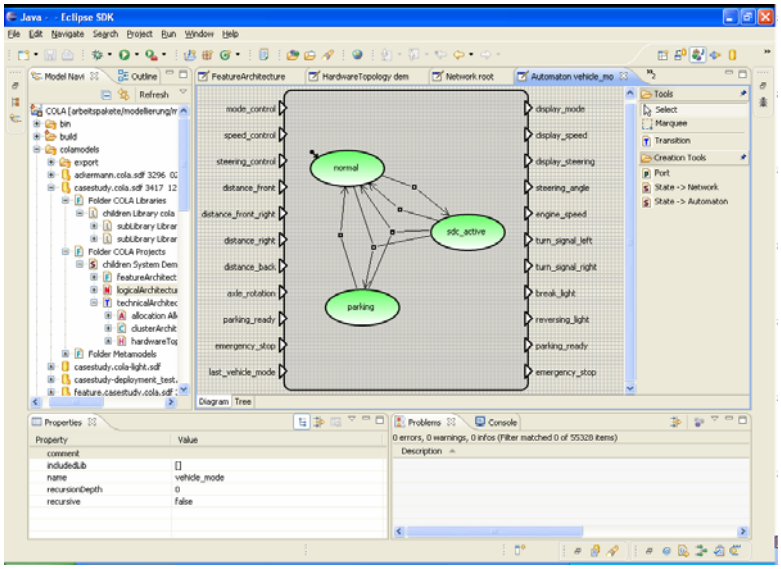
H. Veith 5/2008

COLA Component Language



H. Veith 5/2008

COLA Component Language

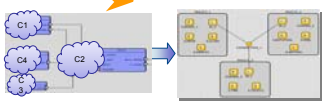


H. Veith 5/2008

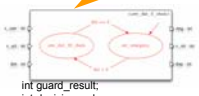
Automotive Tool Chain

COLA model:

Model-level verification, testing, simulation



Clustered COLA model

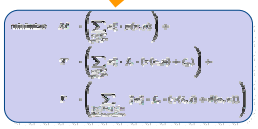


```
int guard_result;
int decision_val;
switch(unit_state->atm_state) {
        case 0:
emergency_guard200747(s_user_i
        n_0, s_act_in_1, dist_in_2,
        &guard_result);
    if(guard_result) {
        unit_state->atm_state =
1;
        ...
```

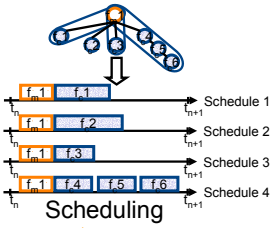
Generated C code for each cluster

SYSTEMC

Execution time estimation



Allocation



Scheduling

H. Veith 5/2008

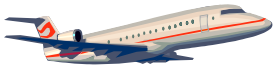
Relating
Efficiency and Computation

Time = Power = Hardware = Money

- Architecture
- Understanding execution time

H. Veith 5/2008

Case Study
Avionic Software

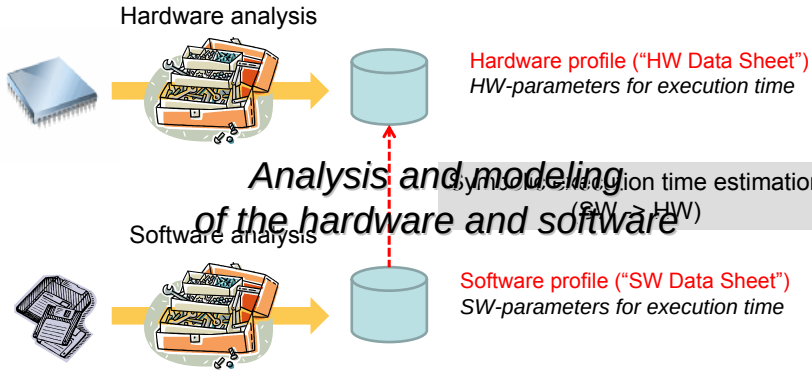


Deployment decisions trial and error

Project Goals
- predictability and efficiency
- simplified automated deployment
- resource aware composition
- reusability of function units (IMA - AADL)
- execution time estimation

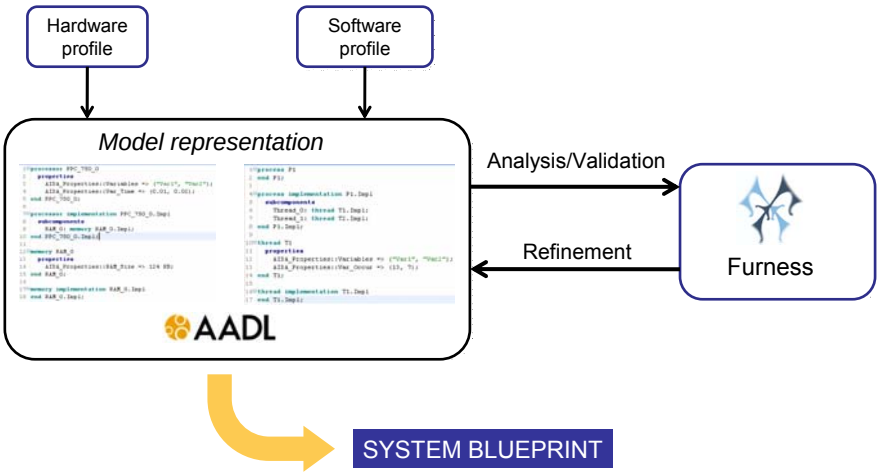
H. Veith 5/2008

Avionic Software Analysis



H. Veith 5/2008

Avionic Software Analysis



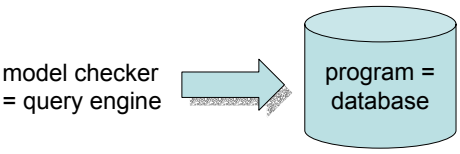
H. Veith 5/2008

Foundational Research

Formal Timing Analysis Suite for Real Time Programs

Advanced white box test case generation
powered by model checking for time measurement

(jointly with Real Time Systems Group, TU Vienna)

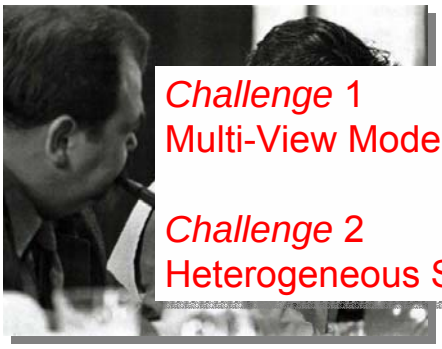


Automatic Extraction of Loop Bounds from C programs

… software engineering requires (cruelly) hard science for its support.

H. Veith 5/2008

Dijkstra Revisited
Challenges



Challenge 1
Multi-View Modeling Framework

Challenge 2
Heterogeneous System Composition

(i) ~~separation of correctness and efficiency~~
(ii) abstraction and hierarchy

H. Veith 5/2008

Challenge 1
Multi-View Modeling Framework

Challenge 2
Heterogeneous System Composition

Challenge 3: Scale !!!

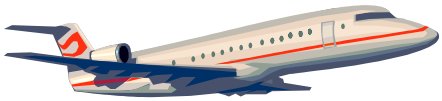
Challenge 4: Tools !!!

Can the entertainment unit affect the engine controller ?
What is the power consumption of class X ?
Can we implement this function in an ASIC ?
Can we deploy the entertainment system on this car ?
How many car variants needed for test driving ?
Can you run integration tests for this architecture ?
Does your geographical database meet hard deadlines ?

H. Veith 5/2008

TECHNISCHE UNIVERSITÄT DARMSTADT

for(syte)



Thank you for your attention !

H. Veith 5/2008