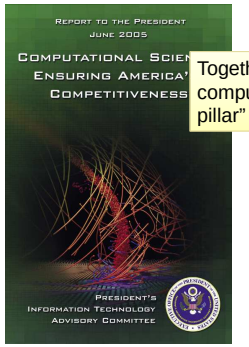
Engineered System Design and Integration—A semantic domain for modeling cyber-physical systems

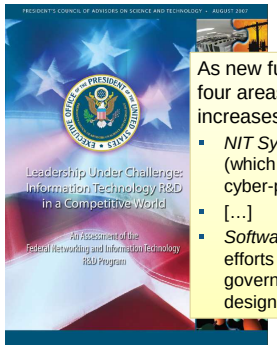
Pieter J. Mosterman

The importance of computation



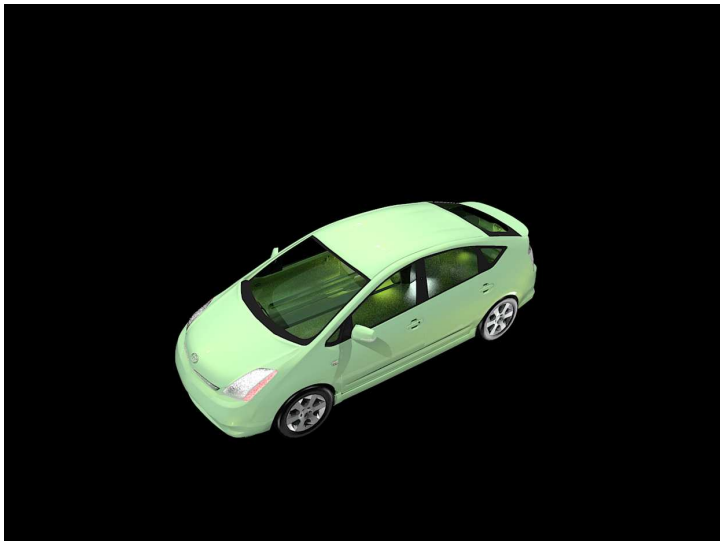
Together with theory and experimentation, computational science now constitutes the "third pillar" of scientific inquiry,

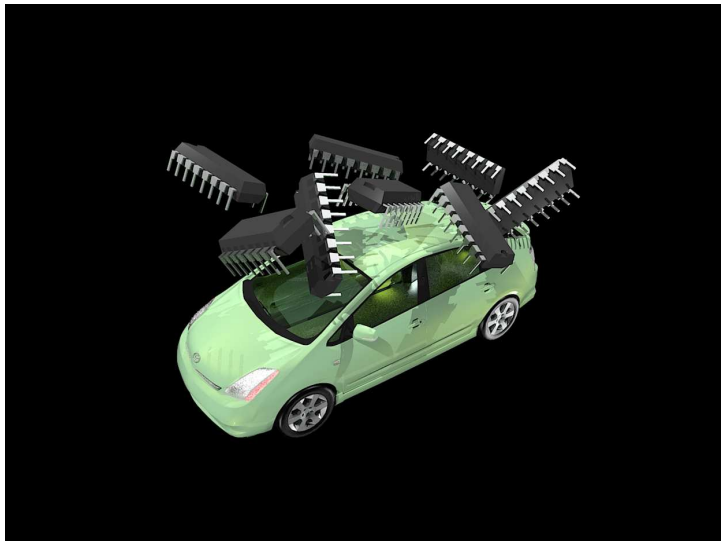
The importance of computation

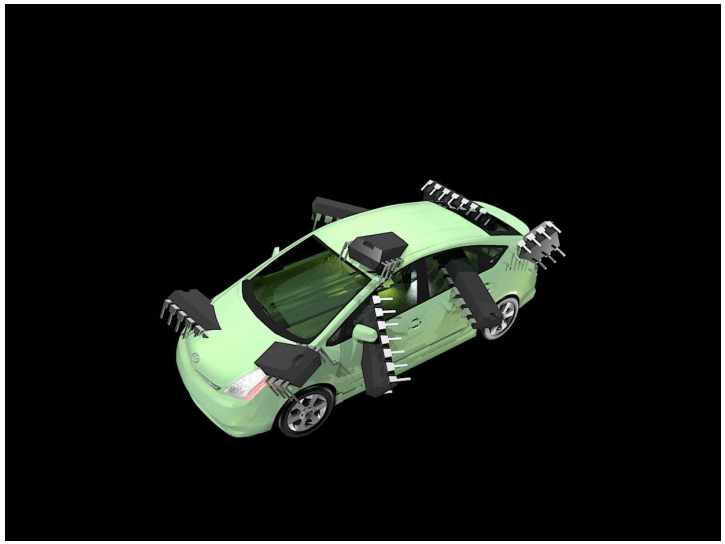


As new funding becomes available, the following four areas should receive disproportionally larger increases […]

- *NIT Systems Connected with the Physical World* (which are also called embedded, engineered, or cyber-physical systems)
- […]
- *Software:* The NITRD Subcommittee should facilitate efforts by leaders from academia, industry, and government to identify critical issues in software design and development





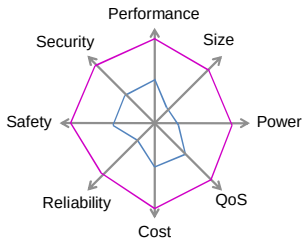


Agenda

- Engineering complex systems
- Cyber-Physical Systems
- Modeling paradigms
- Semantic domains
- Verification for synthesis
- Conclusions

8

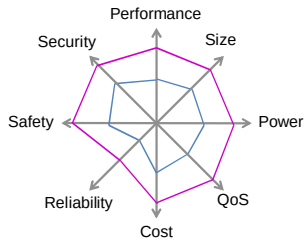
System properties



Known minimum
Known maximum

9

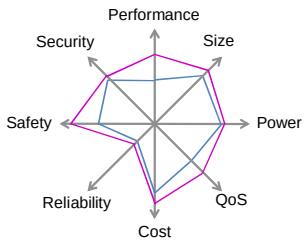
System properties



Known minimum
Known maximum

10

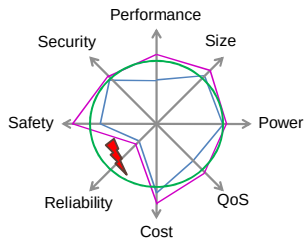
System properties



Known minimum
Known maximum

11

System properties



Known minimum
Known maximum
Required

12

Separation of concerns

Divide and conquer

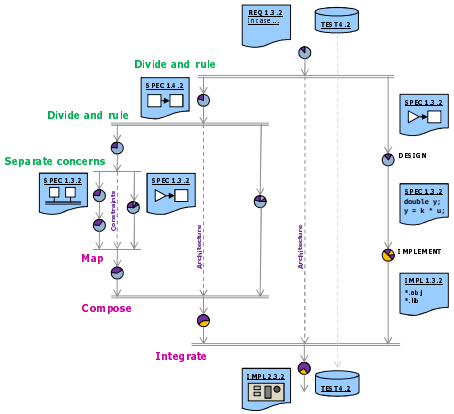
Divide and ~~conquer~~ rule

"Having divided to conquer, we must reunite to rule."
-- Michael Jackson

Michael Jackson, "Some Complexities in Computer- Based Systems and Their Implications for System Development," International Conference on Computer Systems and Software Engineering, pp. 344-351, Tel-Aviv, Israel, May 1990,

MathWorks

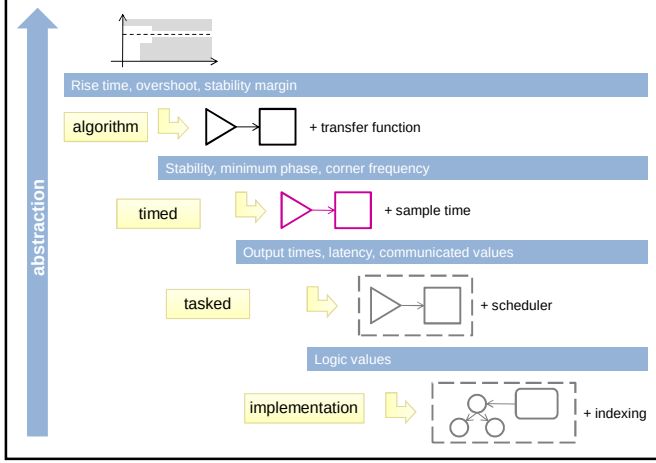
A canonical development structure



16

Separation of concerns

Continuous refinement of abstractions



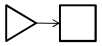
18

Abstraction

- Axiomatic
  - Good for properties

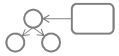
{a=a0 ∧ b=b0} swap(a,b) {a=b0 ∧ b=a0}

- Denotational
  - Good for specifications



swap: initial state ↦ new state

- Operational
  - Good for implementations



swap: tmp = a; a = b; b = tmp;

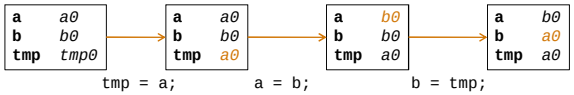
19

Semantics of a swap operation

```
tmp = a;
a = b;
b = tmp;
```

- Operational semantics
  - Details of execution
  - Describe a series of state changes → imperative

swap: tmp = a; a = b; b = tmp;



Taken from Frédéric Boulanger and Cécile Hardebolle, "Execution of models with heterogeneous semantics"

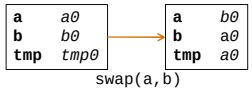
20

Semantics of a swap operation

```
tmp = a;
a = b;
b = tmp;
```

- Denotational semantics
  - Results of execution
  - Describe the path from initial to final state → declarative

    `swap: initial state ↦ new state`



Taken from Frédéric Boulanger and Cécile Hardebolle, "Execution of models with heterogeneous semantics"

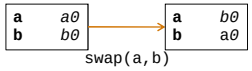
21

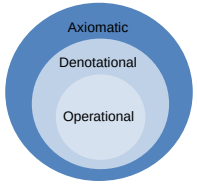
Semantics of a swap operation

```
tmp = a;
a = b;
b = tmp;
```

- Axiomatic semantics
  - Properties of execution state
  - Describe the change of properties of the state → declarative

    `{a=a0 ∧ b=b0} swap(a,b) {a=b0 ∧ b=a0}`

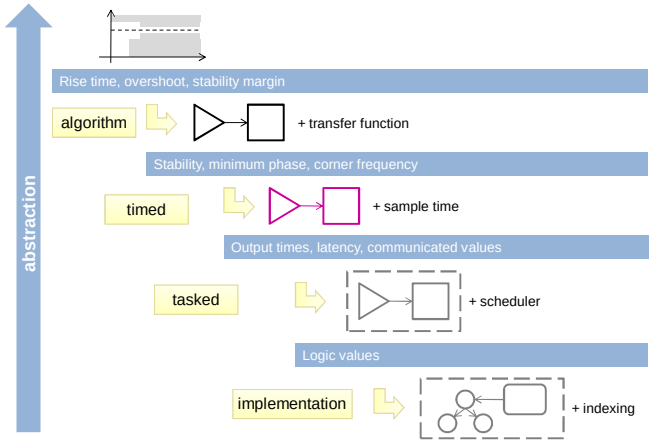




Taken from Frédéric Boulanger and Cécile Hardebolle, "Execution of models with heterogeneous semantics"

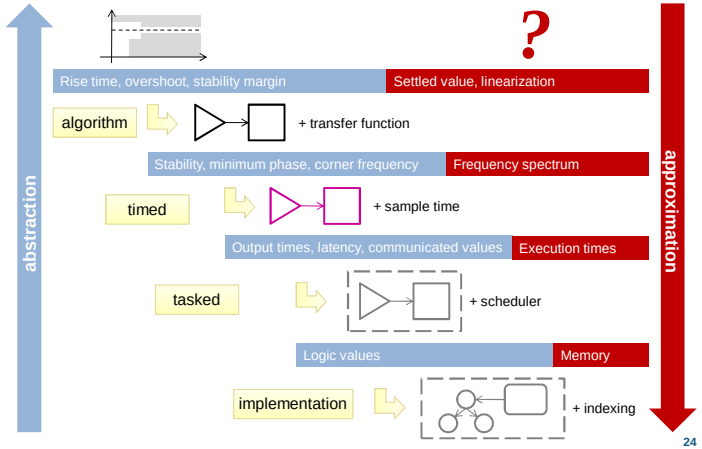
22

Separation of concerns



23

Separation of concerns

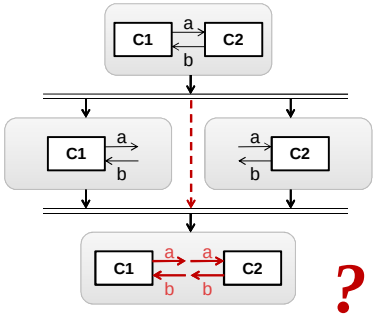


24

Divide and rule

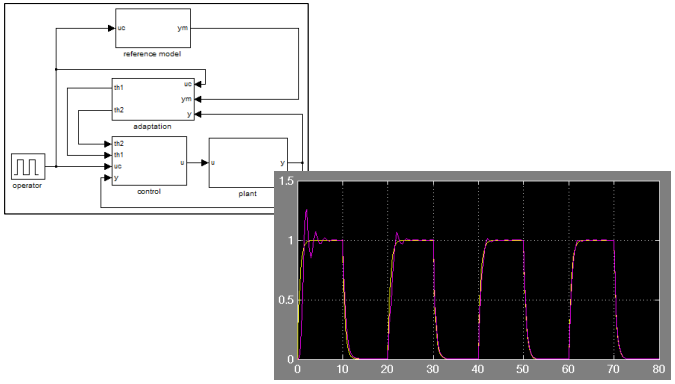
Composition for system integration?

"Whoever ate your sandwich does not like bread crusts."
-- Sherlock Hemlock



26

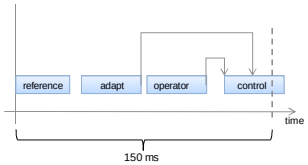
A model-reference adaptive controller



Zhi Han and Pieter J. Mosterman, "**Detecting Data Store Access Conflict in Simulink by Solving Boolean Satisfiability Problems**," in *2010 American Control Conference* (ACC'10), pp. 5702 - 5707, Baltimore, MD, June 30-July 2, 2010

27

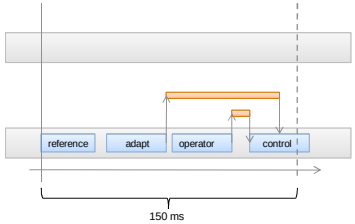
Examine real-time behavior



Does not fit in a single process; multi-tasking!

28

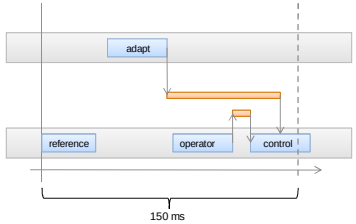
Multi-tasking



Shared memory

29

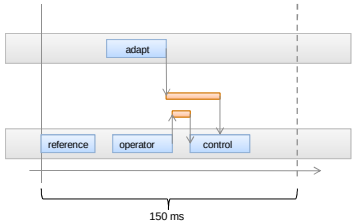
Separation of tasks



Run the adapt task in a separate process

30

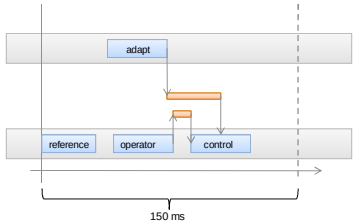
Execution fits in our time budget!



Exploit concurrency in execution resources

31

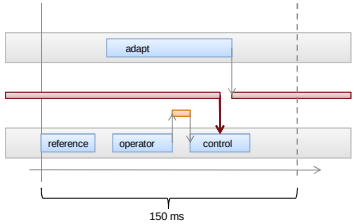
Execution fits in our time budget!



But what if my adapt task runs longer on another processor?

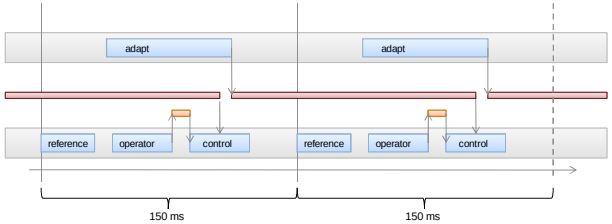
32

Variable is read before it is written



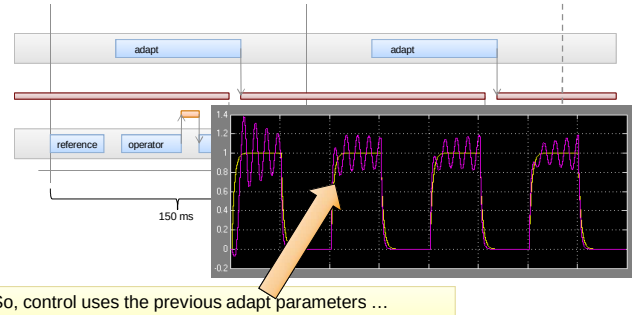
Where does that value for control come from … ?

Incorrect timing?



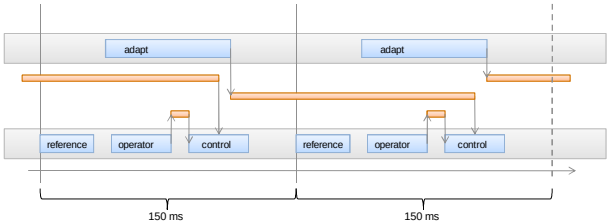
So, control uses the previous adapt parameters …
… how much impact could it possibly have?

Incorrect timing?



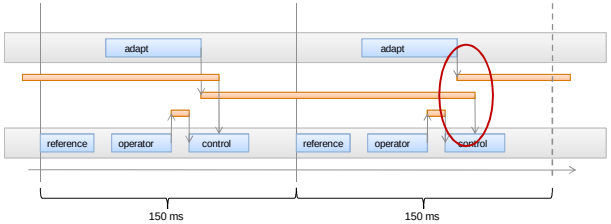
So, control uses the previous adapt parameters …
… how much impact could it possibly have?

Double buffering



The same variable has two memory locations
In one period, read from one memory location and write to the other

Consistent value independent of architecture

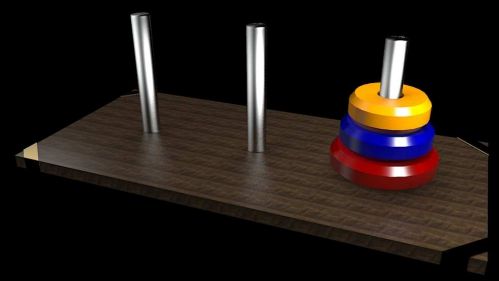


Use previous value, even if the adapt computation may be shorter
- at times
- on different architectures

Determinism: no more surprises! Really, though … ?!

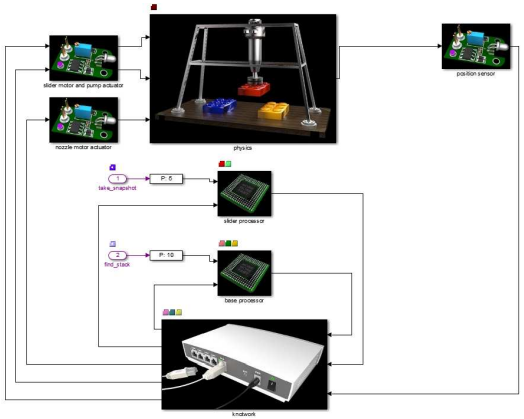
37

Towers of Hanoi …



… as a SCADA system

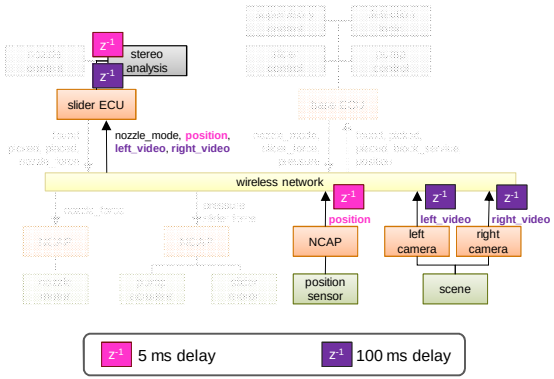
A SCADA system to sort blocks



http://www.mathworks.com/matlabcentral/fileexchange/38515-smart-manufacturing-robotics-cyber-physical-system

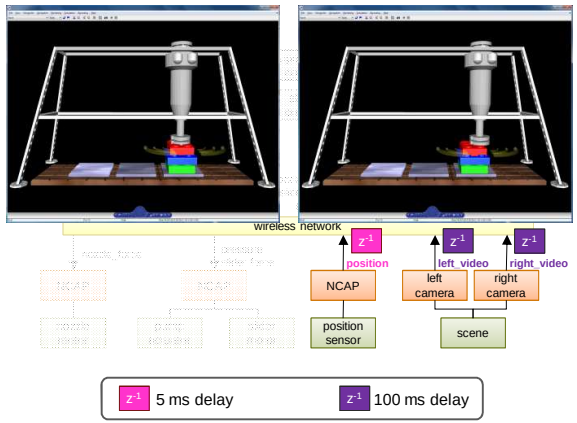
40

A multirate distributed architecture

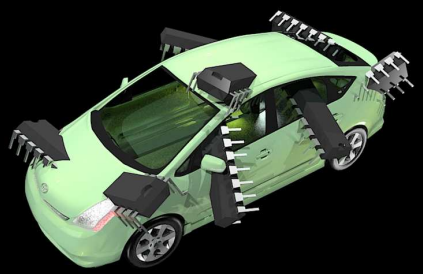


41

A multirate distributed architecture

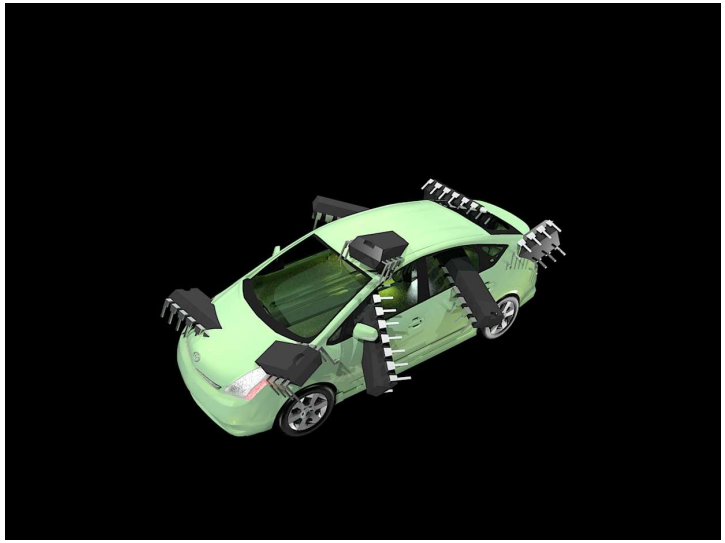


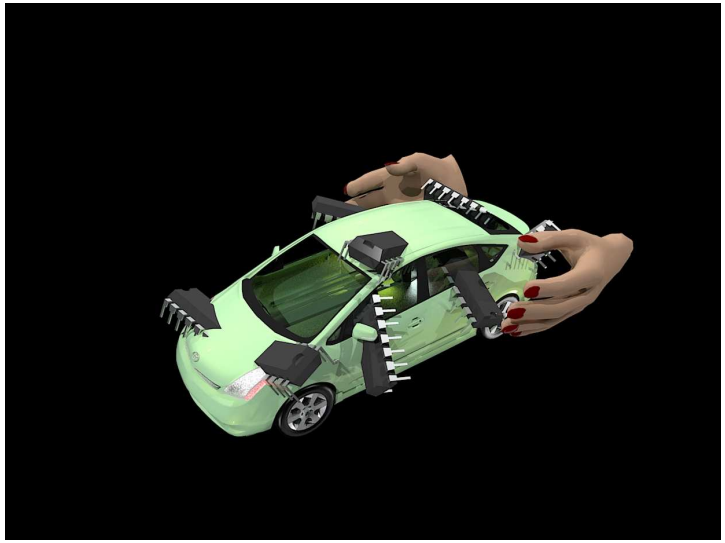
42

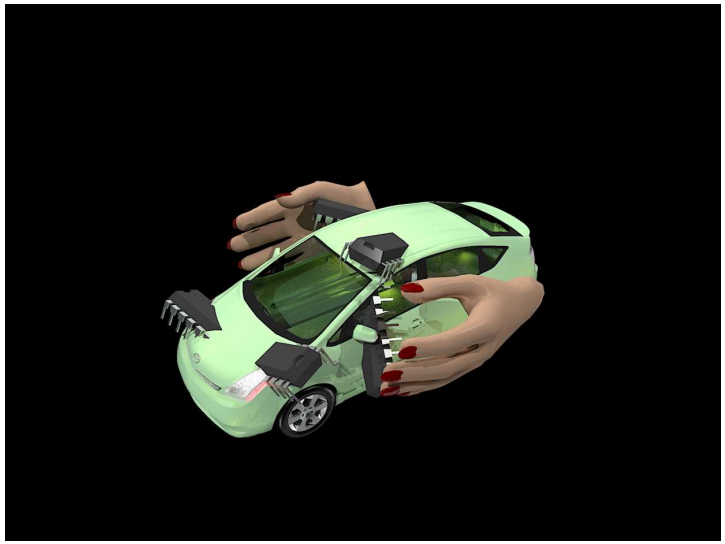


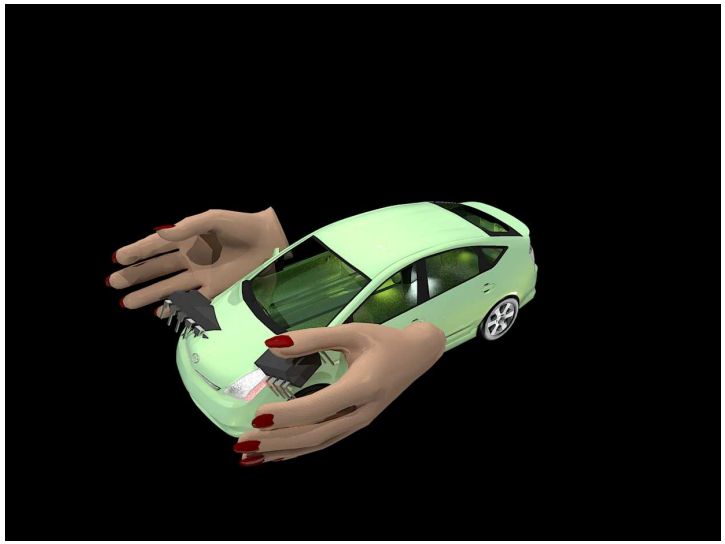
System Integration

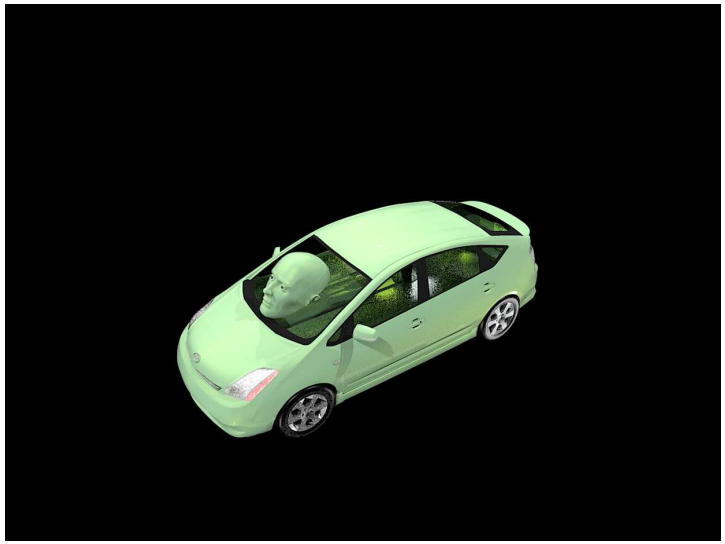
Timing
Concurrency
Interfaces
Shared resources
...









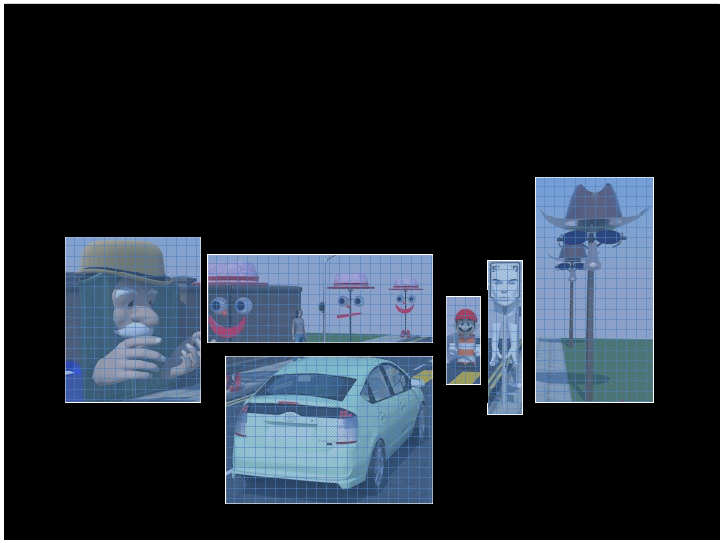


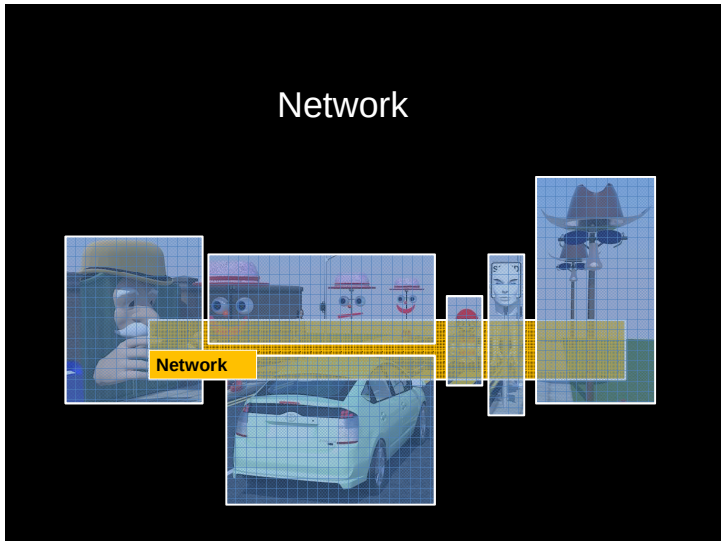
Agenda

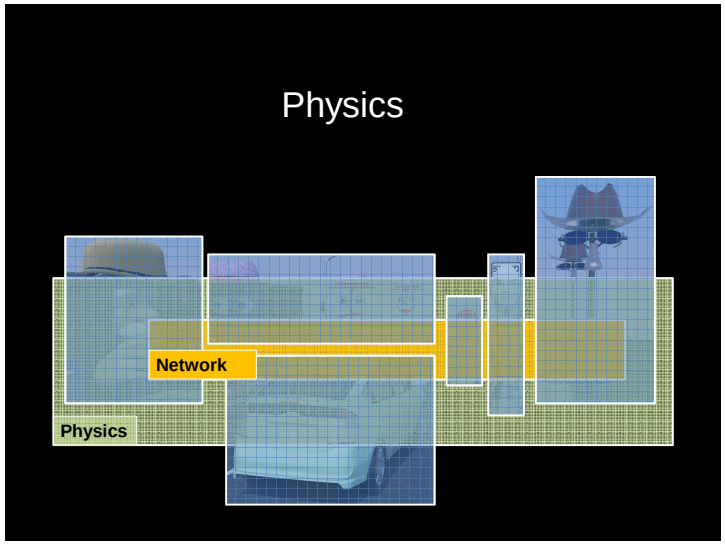
- Engineering complex systems
- Cyber-Physical Systems
- Modeling paradigms
- Semantic domains
- Verification for synthesis
- Conclusions

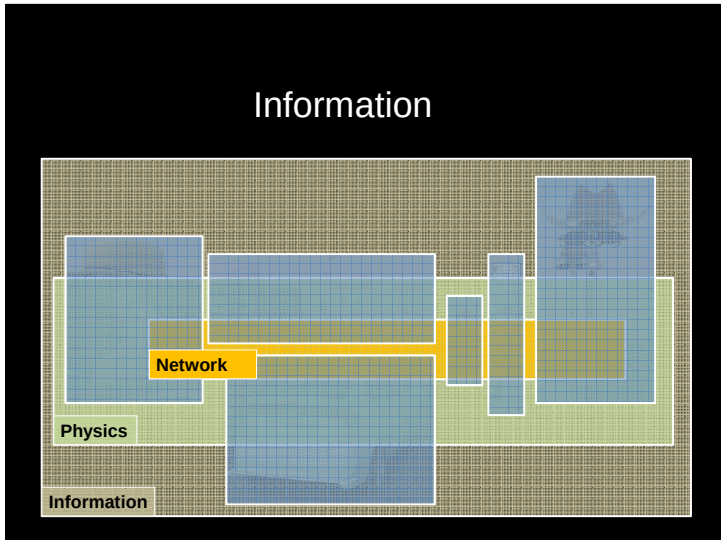
50

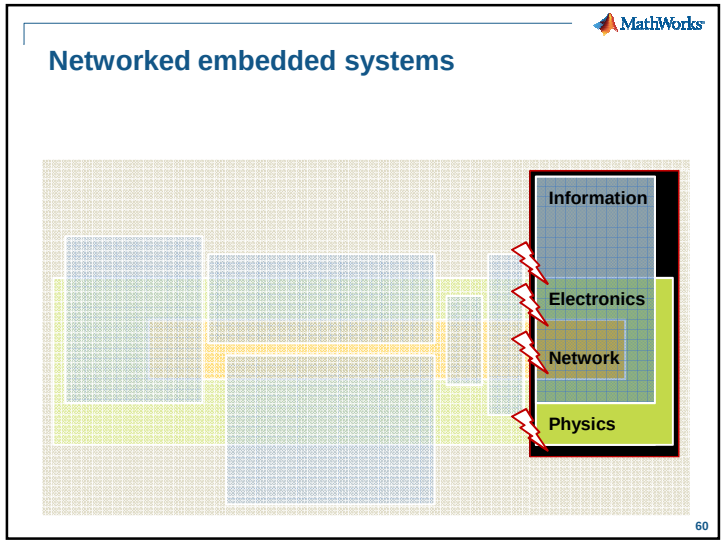

SPEED
LIMIT
35



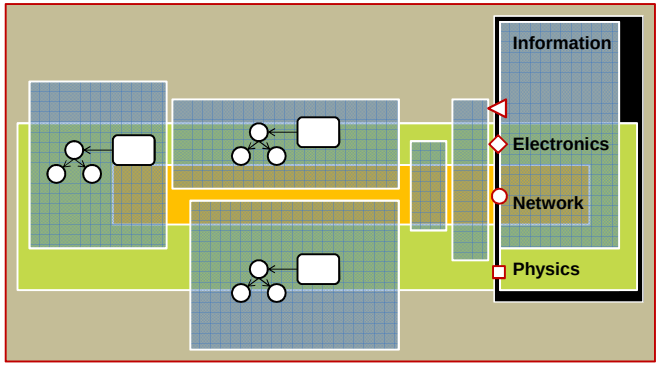

Network
Network


Physics
Network
Physics


Information
Network
Physics
Information

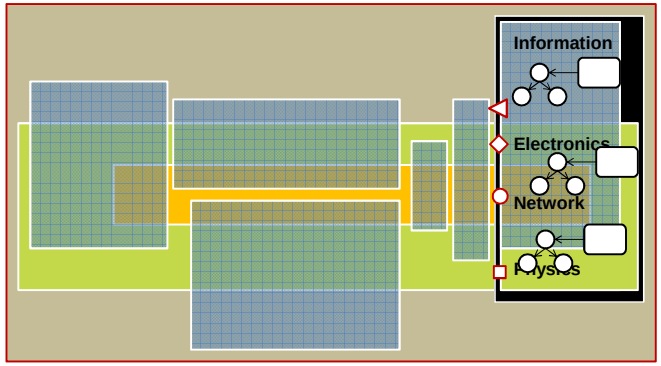

MathWorks
Networked embedded systems
Information
Electronics
Network
Physics
60

Cyber-physical systems



61

Cyber-physical systems



62

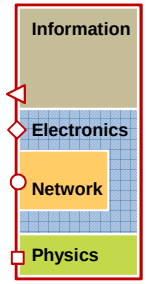
Cyber-physical systems

Shared feature functionality
Feature interaction

Post deployment integration
Emerging behavior

Design of heterogeneous systems

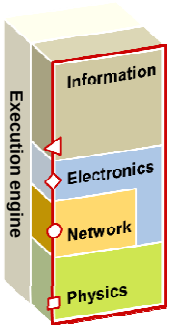
- Executable models
  - Quick feedback on design options
  - Automate design tasks
  - Automate synthesis tasks
  - …
- Computational semantics



64

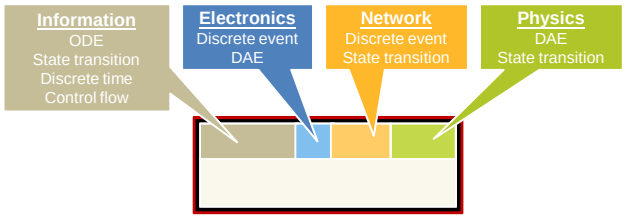
Design of heterogeneous systems

- Executable models
  - Quick feedback on design options
  - Automate design tasks
  - Automate synthesis tasks
  - …
- Computational semantics
- Execution engine
  - Combines many formalisms



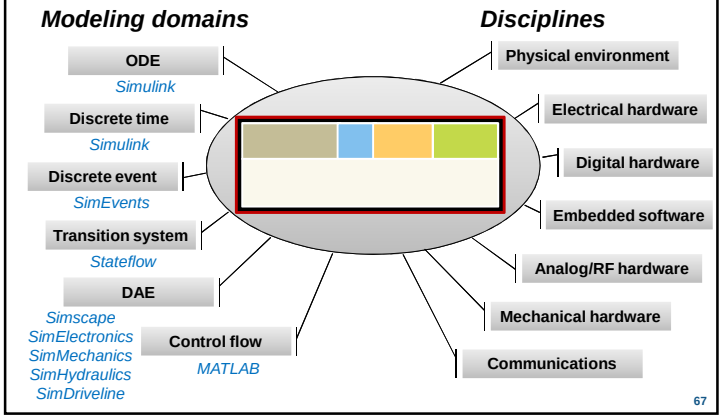
65

Heterogeneity in computational solutions



66

Heterogeneity in computational solutions



67

Agenda

- Engineering complex systems
- Cyber-Physical Systems
- Modeling paradigms
- Semantic domains
- Verification for synthesis
- Conclusions

68


Towers of Hanoi …

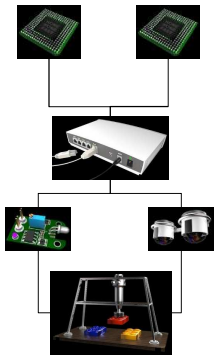

… as a Cyber-Physical System?

A Cyber-Physical System!

Justyna Zander and Pieter J. Mosterman, **"Technical Engine for Democratizing Modeling, Simulation, and Prediction,"** in Proceedings of the *Winter Simulation Conference*, Berlin, Germany, December, 2012

Pieter J. Mosterman, Justyna Zander, and Zhi Han, **"The Towers of Hanoi as a Cyber-Physical System Education Case Study,"** in proceedings of the *First Workshop on CPS Education*, Philadelphia, PA, April 8, 2013

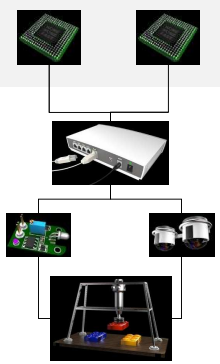
Modeling paradigms



- Signal processing
- Control
  - Supervisory and sequence control
  - Feedback control
- Network, communication
- Physics, plant

74

Modeling the signal processing



- Algorithmic
- Assignments
  - Destructive state access
- Untimed
- Data centric

75

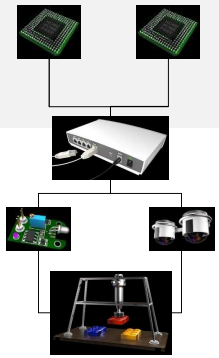
M ssing

```
function row = fcn(image_left, image_right)
%#codegen

persistent hmean;
if isempty(hmean)
    hmean = video.Mean();
end

% number of successive image comparisons
nimages = 100;

% initialize the minimum mean and the corresponding row at which this mean
% is found for the number of successive image comparisons
min = 1e5;
row = 0;

% compute left image submatrix to successively compare
uint8_video_left = uint8(image_left);
left = uint8_video_left(75:224, 1:120, :);

% compute uint8 version of right image
uint8_video_right = uint8(image_right);

parfor k=1:nimages
    % compute successive right image submatrices for comparison
    right = uint8_video_right(125+k:125+149+k, 1:120, :);

    % compare left and right image submatrices
    cmp = bitxor(left,right);

    % compute the mean over all pixels of the comparison results
    pixel_mean = step(hmean,double(cmp));

    % in case of the final row only accept a substantially less
    % (at least 2) value of the mean
    if (pixel_mean < min && k < 100) || (pixel_mean < min - 2)
        min = pixel_mean;
        row = k;
    end
end

% to be consistent with negative offset in graphical version
row = - row;
```

- Algorithmic
- Assignments
  - Destructive state access
- Untimed
- Data centric

76

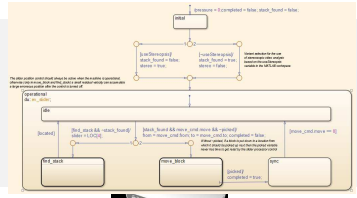
Modeling the supervisory and sequence control

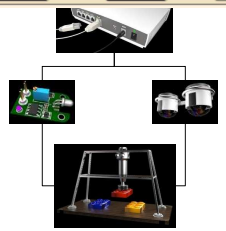


- Discrete state based
- Discrete events cause transitions between states
- Conditions to guard the transition
- Untimed
- Control centric

77

Modeling the supervisory and sequence control

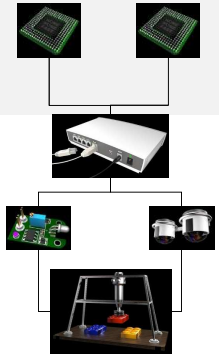




- Discrete state based
- Discrete events cause transitions between states
- Conditions to guard the transition
- Untimed
- Control centric

78

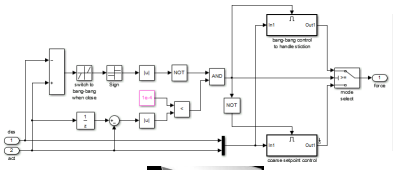
Modeling the feedback control

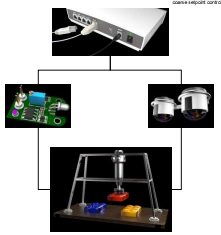


- Sampled discrete time
- Fixed sample time
- Periodic
- Data centric

79

Modeling the feedback control

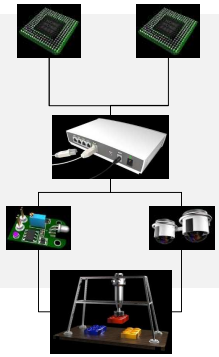




- Sampled discrete time
- Fixed sample time
- Periodic
- Data centric

80

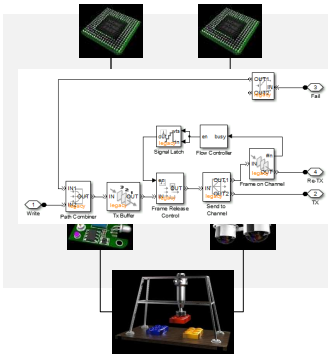
Modeling network traffic



- Entity flow through a graph
- Attributes
  - Source
  - Destination
  - Service time
  - Priority
  - …
- Discrete events
- Preemption
- Data centric
- Aperiodic
- Often stochastic

81

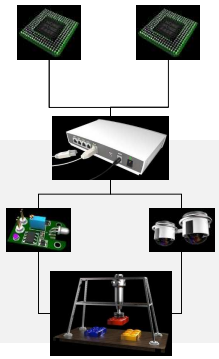
Modeling network traffic



- Entity flow through a graph
- Attributes
  - Source
  - Destination
  - Service time
  - Priority
  - …
- Discrete events
- Preemption
- Data centric
- Aperiodic
- Often stochastic

82

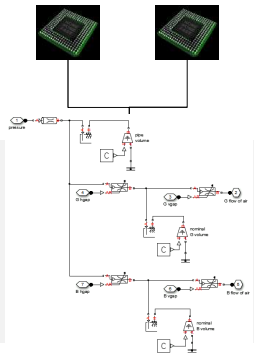
Modeling the plant physics



- Domain-specific modeling—Simscape
  - Electrical
  - Pneumatic
  - Thermal
  - …
- Differential equation based
- Noncausal, energy-based, modeling

83

Modeling the plant physics

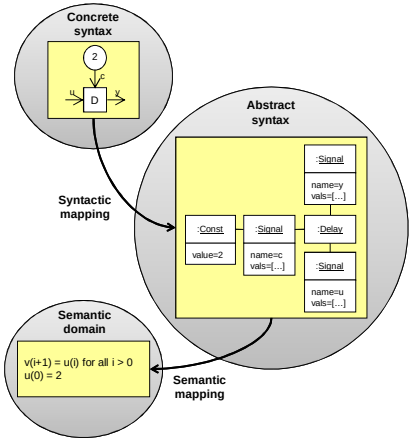


- Domain-specific modeling—Simscape
  - Electrical
  - Pneumatic
  - Thermal
  - …
- Differential equation based
- Noncausal, energy-based, modeling

84

The elements of a model

- Syntax
  - Concrete syntax
  - Abstract syntax
  - Syntactic mapping
- Semantics
  - Semantic domain
  - Semantic mapping



Developed at the Computer Automated Multiparadigm Modeling (CAMPaM) workshop, McGill Bellairs Research Institute

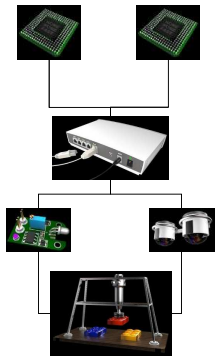
85

Agenda

- Engineering complex systems
- Cyber-Physical Systems
- Modeling paradigms
- Semantic domains
- Verification for synthesis
- Conclusions

86

A cyber-physical architecture



- Signal processing
  - Data intensive algorithms
- Control
  - Frequent periodic events
- Network, communication
  - Frequent aperiodic events
- Physics, plant
  - Continuous with sporadic events

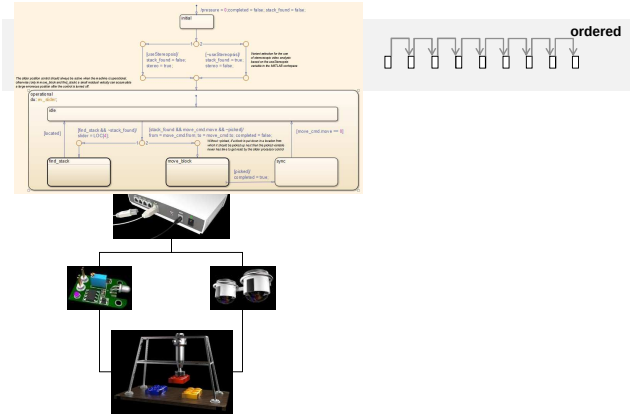
87

All part of the networked embedded system

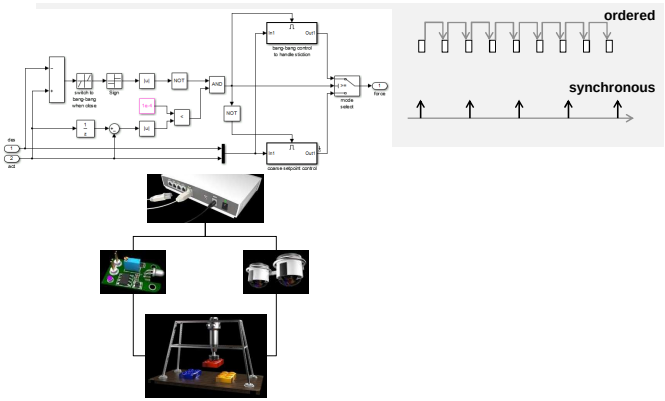
```
function row = fcn(image_left, image_right)
%#codegen

persistent hmean;
if isempty(hmean)
    hmean = video.Mean();
end

% number of successive image comparisons
nimages = 100;

% initialize the minimum mean and the corresponding row at which this mean
% is found for the number of successive image comparisons
min = 1e5;
row = 0;

% compute left image submatrix to successively compare
uint8_video_left = uint8(image_left);
left = uint8_video_left(75:224, 1:120, :);

% compute uint8 version of right image
uint8_video_right = uint8(image_right);

parfor k=1:nimages
    % compute successive right image submatrices for comparison
    right = uint8_video_right(125+k:125+149+k, 1:120, :);

    % compare left and right image submatrices
    cmp = bitxor(left,right);

    % compute the mean over all pixels of the comparison results
    pixel_mean = step(hmean,double(cmp));

    % in case of the final row only accept a substantially less
    % (at least 2) value of the mean
    if (pixel_mean < min && k < 100) || (pixel_mean < min - 2)
        min = pixel_mean;
        row = k;
    end
end

% to be consistent with negative offset in graphical version
row = - row;
```

88

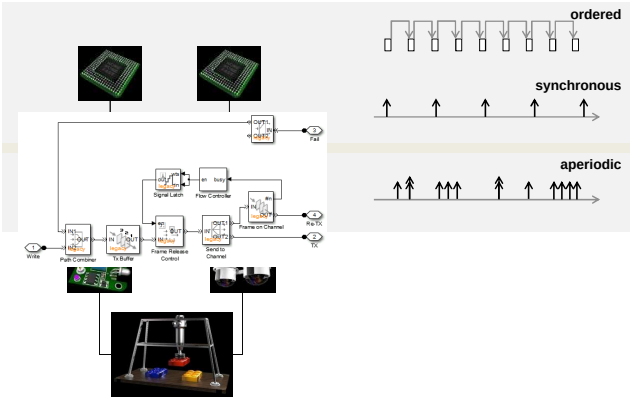
All part of the networked embedded system



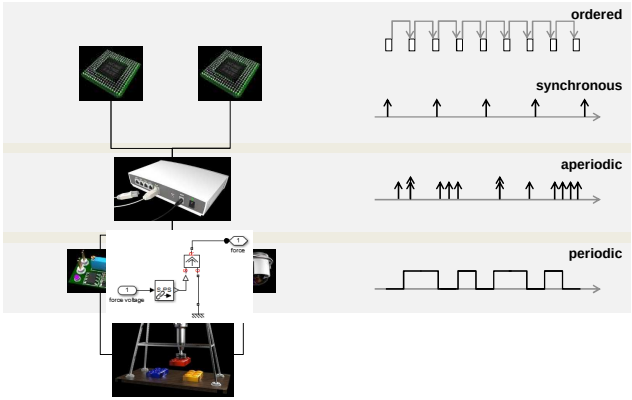
All part of the networked embedded system



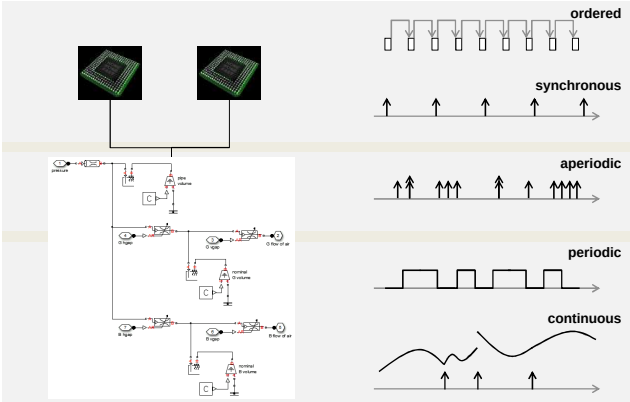
All part of the networked embedded system



All part of the networked embedded system



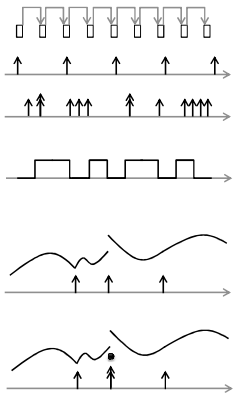
All part of the networked embedded system



93

The semantic domain of a dynamic system

- Points, [ ]
  - On **N**
  - On **R** x **N**
- Intervals, [⟩ (⟨ ⟩, ⟨])
  - On **R**
- Hybrid point/interval
  - On **R**
  - On **R** x **N**



MATLAB, Stateflow

Discrete time Simulink

SimEvents

Simulink

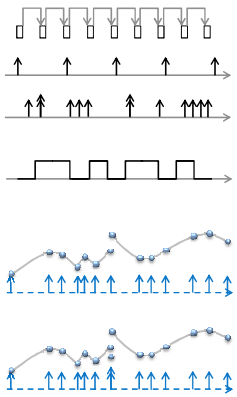
Simulink, Simscape

Simulink, Simscape

94

The semantic domain of a dynamic system

- Points, [ ]
  - On **N**
  - On **R** x **N**
- Intervals, [⟩ (⟨ ⟩, ⟨])
  - On **R**
- Hybrid point/interval
  - On **R**
  - On **R** x **N**



MATLAB, Stateflow

Discrete time Simulink

SimEvents

Simulink

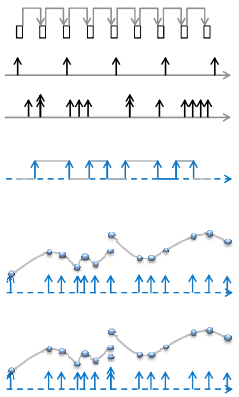
Simulink, Simscape

Simulink

95

The semantic domain of a dynamic system

- Points, [ ]
  - On **N**
  - On **R** x **N**
- Intervals, [⟩ (⟨ ⟩, ⟨])
  - On **R**
- Hybrid point/interval
  - On **R**
  - On **R** x **N**



MATLAB, Stateflow

Discrete time Simulink

SimEvents

Simulink

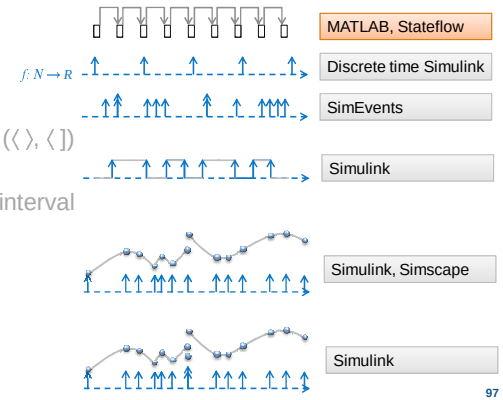
Simulink, Simscape

Simulink

96

The semantic domain of a dynamic system

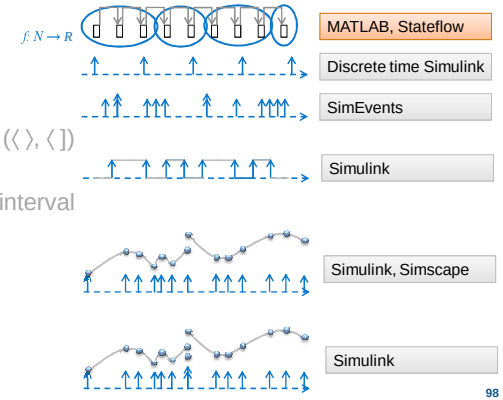
- Points, [ ]
  - On **N**
  - On **R** x **N**
- Intervals, [⟩ (⟨ ⟩, ⟨])
  - On **R**
- Hybrid point/interval
  - On **R**
  - On **R** x **N**



97

The semantic domain of a dynamic system

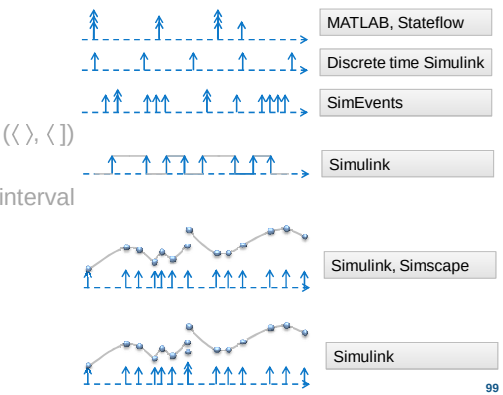
- Points, [ ]
  - On **N**
  - On **R** x **N**
- Intervals, [⟩ (⟨ ⟩, ⟨])
  - On **R**
- Hybrid point/interval
  - On **R**
  - On **R** x **N**



98

The semantic domain of a dynamic system

- Points, [ ]
  - On **N**
  - On **R** x **N**
- Intervals, [⟩ (⟨ ⟩, ⟨])
  - On **R**
- Hybrid point/interval
  - On **R**
  - On **R** x **N**



99

A general operational semantic domain

- Points, [ ]
  - On **R** x **N**
- Without losing the analysis ability and efficiency
  - Integer precision
  - Clock calculus
  - Scheduling
    - Static when possible
    - Dynamic when necessary

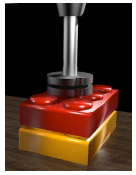
100

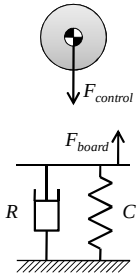
Agenda

- Engineering complex systems
- Cyber-Physical Systems
- Modeling paradigms
- Semantic domains
- Verification for synthesis
- Conclusions

A surface mount device

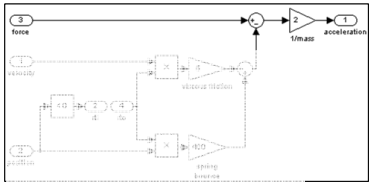
- Newton and Hooke's Laws
  - Differential equations
- Contact behavior
  - Discontinuous changes
- Control behavior
  - Sampled data (periodic)

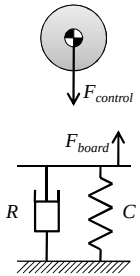




A surface mount device

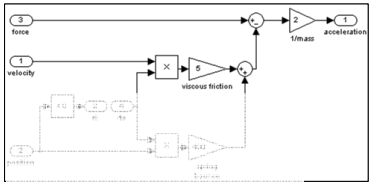
- Newton's Law
  $F_{control} = ma \Leftrightarrow a = \frac{1}{m} F_{control}$

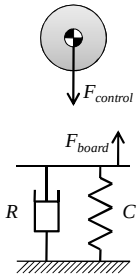




A surface mount device

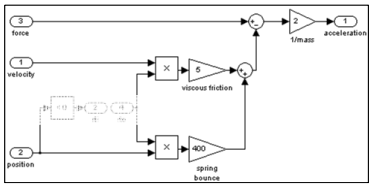
- Newton's Law
  $F_{control} = ma \Leftrightarrow a = \frac{1}{m} F_{control}$
- Viscous friction
  $F_R = Rv$

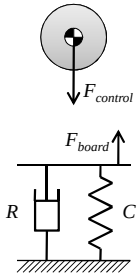




A surface mount device

- Newton's Law
  $F_{control} = ma \Leftrightarrow a = \frac{1}{m} F_{control}$
- Viscous friction
  $F_R = Rv$
- Hooke's Law
  $F_C = C(x - x_0) \Leftrightarrow F_c = Cx$



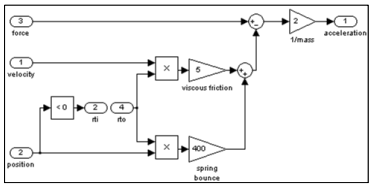


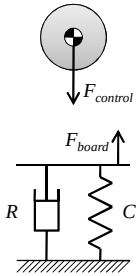
105

A surface mount device

- Contact behavior
  - Discontinuous changes

$$F_{board}(t) = \begin{cases} -\left(Rv(t) + \frac{x(t)}{C}\right) & if\ x(t) < 0 \\ 0 & otherwise \end{cases}$$



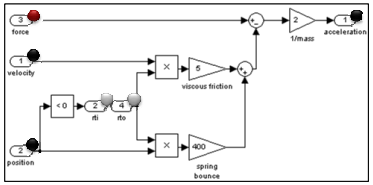


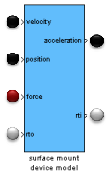
106

A surface mount device

- Controller behavior
  - Sampled data

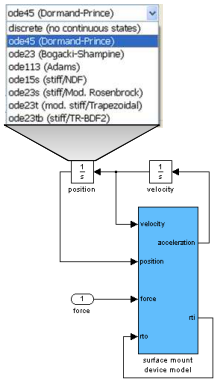
$$F_{control}(k) = u(k) \quad with \quad t = kT_s$$



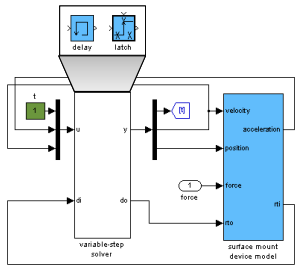


107

Explicitly modeled execution engine

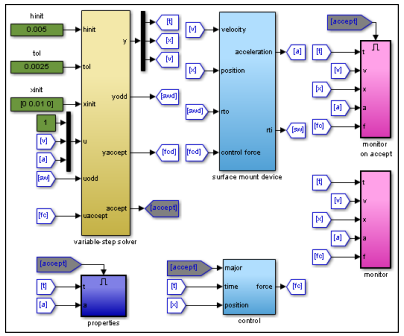


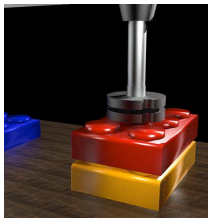
Completely modeled solver and rate transition with the discontinuous world …
… all with two basic 'sequential' blocks



108

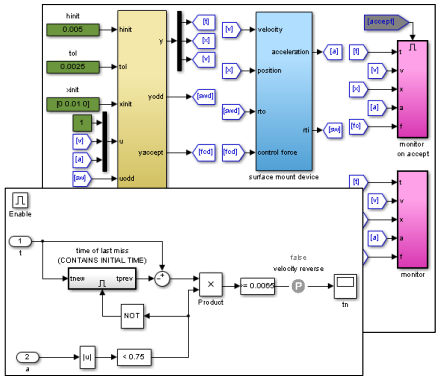
Control synthesis using model checking





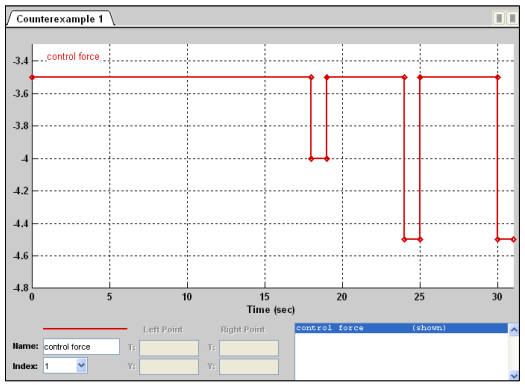
http://www.mathworks.com/matlabcentral/fileexchange/authors/4449

Control synthesis using model checking



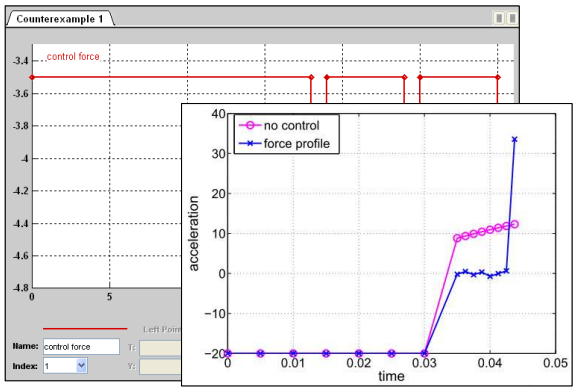
http://www.mathworks.com/matlabcentral/fileexchange/authors/4449

A counterexample



Pieter J. Mosterman, Justyna Zander, Grégoire Hamon, and Ben Denckla, "**A Computational Model of Time for Stiff Hybrid Systems Applied to Control Synthesis**," in *Control Engineering Practice*, , vol. 20, nr. 1, pp. 2-13, January 2012

A counterexample



Pieter J. Mosterman, Justyna Zander, Grégoire Hamon, and Ben Denckla, "**A Computational Model of Time for Stiff Hybrid Systems Applied to Control Synthesis**," in *Control Engineering Practice*, , vol. 20, nr. 1, pp. 2-13, January 2012

Agenda

- Engineering complex systems
- Cyber-Physical Systems
- Modeling paradigms
- Semantic domains
- Verification for synthesis
- Conclusions

113

Conclusions

- Engineered system design—principles and challenges
  - Separation of concerns
  - Divide and rule
- Cyber-physical system
  - Post deployment integration of shared functionality
  - Models are critical in design
- Many different disciplines, problems, and technologies
  - Multiparadigm modeling!
  - Variety of semantic domains
    - A unifying semantic domain on functions over streams
    - Reasoning across modeling paradigms

114

MathWorks®

Accelerating the pace of engineering and science

115