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

Pieter J. Mosterman

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
System properties

- Performance
- Security
- Size
- Power
- Safety
- Reliability
- Cost
- QoS

Known minimum
Known maximum

System properties

- Performance
- Security
- Size
- Power
- Safety
- Reliability
- Cost
- QoS

Known minimum
Known maximum

System properties

- Performance
- Security
- Size
- Power
- Safety
- Reliability
- Cost
- QoS

Known minimum
Known maximum

System properties

- Performance
- Security
- Size
- Power
- Safety
- Reliability
- Cost
- QoS

Known minimum
Known maximum

Required
Separation of concerns

Divide and conquer

Divide and conquer

“Having divided to conquer, we must reunite to rule.”

--- Michael Jackson

A canonical development structure
Separation of concerns

Abstraction

- Axiomatic
  - Good for properties
    \[(a=b) \land (b=c) \Rightarrow (a=b) \land (b=c)\]

- Denotational
  - Good for specifications
    \[
    \text{swap: initial state } \rightarrow \text{ new state}
    \]

- Operational
  - Good for implementations
    \[
    \text{swap: tmp = a; a = b; b = tmp;}
    \]

Continuous refinement of abstractions

Semantics of a swap operation

- Operational semantics
  - Details of execution
    - Describe a series of state changes \(\rightarrow\) imperative
      \[
      \text{swap: tmp = a; a = b; b = tmp;}
      \]

  \[
  \begin{array}{c}
  \text{b} \quad \text{b}
  \\
  \text{a} \quad \text{a}
  \\
  \text{tmp} \quad \text{tmp}
  \\
  \text{tmp} \quad \text{tmp}
  \\
  \end{array}
  \]

  - \(a = b\)
  - \(b = \text{tmp}\)
  - \((a = b) \land (b = \text{tmp})\)
Semantics of a swap operation

\[
\begin{align*}
\text{tmp} &= a; \\
\text{a} &= b; \\
\text{b} &= \text{tmp};
\end{align*}
\]

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

\[
\text{swap: initial state } \Rightarrow \text{ new state}
\]

\[
\begin{array}{c|c|c}
\text{a} & \text{b} & \text{tmp} \\
\hline
\text{a0} & \text{b0} & \text{tmp0}
\end{array}
\]

\[
\begin{array}{c|c|c}
\text{a} & \text{b} & \text{tmp} \\
\hline
\text{b0} & \text{a0} & \text{a0}
\end{array}
\]

\[
\text{swap(a,b)}
\]

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

\[
\{a=a0 \land b=b0\} \Rightarrow \{a=b0 \land b=a0\}
\]

\[
\begin{array}{c|c|c}
\text{a} & \text{b} & \text{tmp} \\
\hline
\text{a0} & \text{b0} & \text{b0}
\end{array}
\]

\[
\begin{array}{c|c|c}
\text{a} & \text{b} & \text{tmp} \\
\hline
\text{b0} & \text{a0} & \text{a0}
\end{array}
\]

\[
\text{swap(a,b)}
\]

Separation of concerns

- **Rise time, overshoot, stability margin**
  - Transfer function
  - Sample time
  - Scheduler
  - Sample time

- **Algorithm**
  - Stability, minimum phase, corner frequency
  - Output times, latency, communicated values
  - Logic values
  - Implementation
  - Indexing

- **Settled value, linearization**
  - Stability, minimum phase, corner frequency
  - Frequency spectrum
  - Output times, latency, communicated values
  - Execution times
  - Memory
  - Indexing
A model-reference adaptive controller

Composition for system integration?
“Whoever ate your sandwich does not like bread crusts.”
-- Sherlock Hemlock

Examine real-time behavior

Does not fit in a single process: multi-tasking!
Multi-tasking

Separation of tasks

Execution fits in our time budget!

Execution fits in our time budget!

- Run the adapt task in a separate process
- But what if my adapt task runs longer on another processor?
- Exploit concurrency in execution resources
- Shared memory
- Shared memory
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
Towers of Hanoi ...

... as a SCADA system

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

A SCADA system to sort blocks

A multirate distributed architecture

System Integration
Timing
Concurrency
Interfaces
Shared resources
...
Computation as main feature differentiator

System integration
Agenda

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

Design of heterogeneous systems

- Executable models
  - Quick feedback on design options
  - Automate design tasks
  - Automate synthesis tasks
  - ...
- Computational semantics
Design of heterogeneous systems

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

Heterogeneity in computational solutions

Modeling domains

- ODE
- Simulink
- Discrete time
- Simulink
- Discrete event
- SimEvents
- Transition system
- Stateflow
- DAE
- Simscape
- SimElectronics
- SimMechanics
- SimHydraulics
- SimDriveline

Disciplines

- Physical environment
- Electrical hardware
- Digital hardware
- Embedded software
- Analog/RF hardware
- Mechanical hardware
- Communications

Agenda

- Engineering complex systems
- Cyber-Physical Systems
- Modeling paradigms
- Semantic domains
- Verification for synthesis
- Conclusions
Towers of Hanoi …

… as a Cyber-Physical System?
Modeling the signal processing

- Algorithmic
- Assignments
- Destructive state access
- Untimed
- Data centric
Modeling the supervisory and sequence control

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

Modeling the feedback control

• Sampled discrete time
• Fixed sample time
• Periodic
• Data centric
Modeling network traffic

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

Modeling the plant physics

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

Modeling network traffic

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

Modeling the plant physics

- Domain-specific modeling—Simscape
  - Electrical
  - Pneumatic
  - Thermal
  - ...
- Differential equation based
- Noncausal, energy-based, modeling
The elements of a model

- Syntax
  - Concrete syntax
  - Abstract syntax
  - Syntactic mapping

- Semantics
  - Semantic domain
  - Semantic mapping

A cyber-physical architecture

- Signal processing
  - Data intensive algorithms

- Control
  - Frequent periodic events

- Network, communication
  - Frequent aperiodic events

- Physics, plant
  - Continuous with sporadic events

Agenda

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

All part of the networked embedded system
All part of the networked embedded system

ordered

synchronous

aperiodic

periodic
All part of the networked embedded system

The semantic domain of a dynamic system

- Points, [ ]
  - On \( \mathbb{N} \)
  - On \( \mathbb{R} \times \mathbb{N} \)

- Intervals, \([ \langle \rangle, \langle \rangle \])
  - On \( \mathbb{R} \)

- Hybrid point/interval
  - On \( \mathbb{R} \)
  - On \( \mathbb{R} \times \mathbb{N} \)
The semantic domain of a dynamic system

- Points, [ ]
  - On \( \mathbb{N} \)
  - On \( \mathbb{R} \times \mathbb{N} \)

- Intervals, [ ), ( )]
  - On \( \mathbb{R} \)
  - Hybrid point/interval
    - On \( \mathbb{R} \)
    - On \( \mathbb{R} \times \mathbb{N} \)

Simulink, Simscape

The semantic domain of a dynamic system

- Points, [ ]
  - On \( \mathbb{N} \)
  - On \( \mathbb{R} \times \mathbb{N} \)

- Intervals, [ ), ( )]
  - On \( \mathbb{R} \)
  - Hybrid point/interval
    - On \( \mathbb{R} \)
    - On \( \mathbb{R} \times \mathbb{N} \)

Simulink, Simscape

The semantic domain of a dynamic system

- Points, [ ]
  - On \( \mathbb{N} \)
  - On \( \mathbb{R} \times \mathbb{N} \)

- Intervals, [ ), ( )]
  - On \( \mathbb{R} \)
  - Hybrid point/interval
    - On \( \mathbb{R} \)
    - On \( \mathbb{R} \times \mathbb{N} \)

Simulink, Simscape

A general operational semantic domain

- Points, [ ]
  - On \( \mathbb{R} \times \mathbb{N} \)

- Without losing the analysis ability and efficiency
  - Integer precision
  - Clock calculus
  - Scheduling
    - Static when possible
    - Dynamic when necessary
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
  \[ F_{\text{control}} = ma \Leftrightarrow a = \frac{1}{m} F_{\text{control}} \]

A surface mount device

- Newton’s Law
  \[ F_{\text{control}} = ma \Leftrightarrow a = \frac{1}{m} F_{\text{control}} \]
- Viscous friction
  \[ F_{\text{control}} = RV \]
**A surface mount device**

- Newton's Law
  \[ F_{\text{control}} = ma \implies a = \frac{1}{m} F_{\text{control}} \]
- Viscous friction
  \[ F_r = Rv \]
- Hooke's Law
  \[ F_c = C(x - x_0) \implies F_c = Cx \]

**A surface mount device**

- Contact behavior
  - Discontinuous changes
  \[ F_{\text{board}}(t) = \begin{cases} \frac{K(t)}{C} & \text{if } x(t) < 0 \\ 0 & \text{otherwise} \end{cases} \]

**Explicitly modeled execution engine**

- Controller behavior
  - Sampled data
  \[ F_{\text{control}}(k) = u(k) \text{ with } t = kT_s \]

Completely modeled solver and rate transition with the discontinuous world … … all with two basic 'sequential' blocks
Control synthesis using model checking

A counterexample

Agenda

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

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