NSF CPS: Breakthrough: Compositional System Modeling with Interfaces (COSMOI)

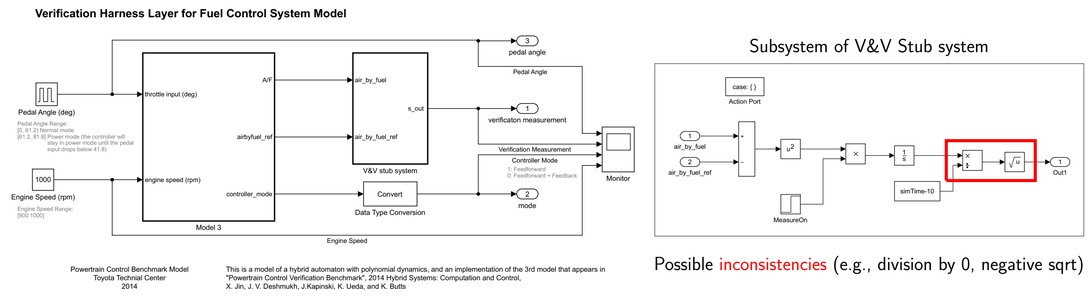
PI: Stavros Tripakis Co-PI: Edward A. Lee
University of California, Berkeley

Demo: The Refinement Calculus of Reactive Systems (RCRS)

`http://rcrs.cs.aalto.fi`

Iulia Dragomir Viorel Preoteasa Stavros Tripakis

1. Goal: Compositional Reasoning for Simulink



Subsystem of V&V Stub system

Possible inconsistencies (e.g., division by 0, negative sqrt)

Figure: Example: Simulink model of a Fuel Control System (public benchmark by Toyota)

- Compositional Static Analysis: detect inconsistencies, compute preconditions, eliminate internal variables, check substitutability (when can a block replace another?), etc., at compile-time, without flattening!

2. The RCRS Framework (Refinement Calculus of Reactive Systems)

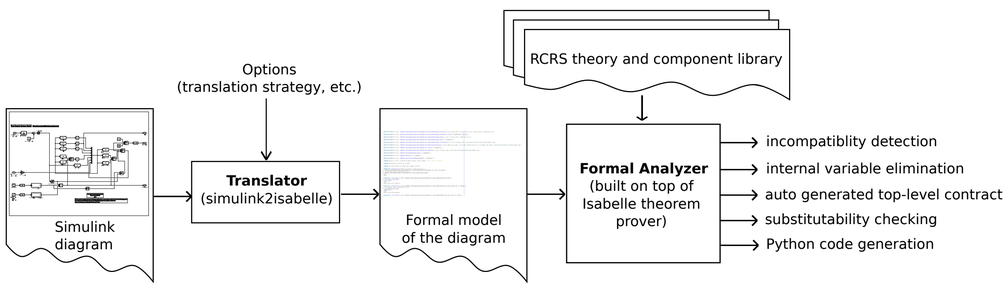
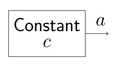


Figure: RCRS Framework Workflow

3. The Algebra of Hierarchical Block Diagrams (HBDs)

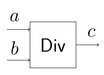
Challenge 1: How to represent graphical diagrams in a textual notation with formal semantics?

- Basic blocks: represented as atomic monotonic predicate transformers (MPTs). Some examples:
  - *stateless basic block*:



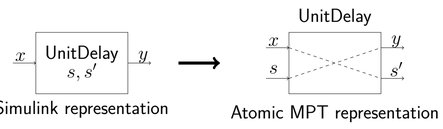
is defined as: Constant $= [() \rightsquigarrow c]$

  - *stateless basic block with precondition*:



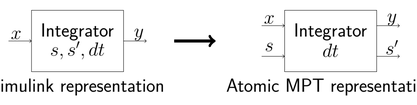
is defined as: Div $= \{a, b.\ b \neq 0\ \} \circ [a, b \rightsquigarrow \frac{a}{b}]$

  - *discrete-time stateful basic block* (s: current state, s': next state):



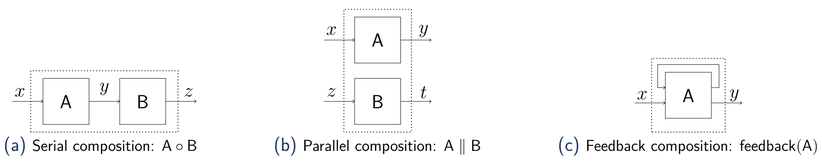
is defined as: UnitDelay $= [x, s \rightsquigarrow s, x]$

  - *continuous-time stateful basic block* (fixed time-step integration with time-step parameter dt):



is defined as: Integrator$(dt) = [x, s \rightsquigarrow s, s + x \cdot dt]$

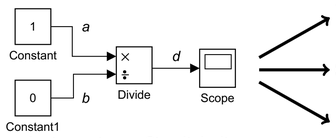
- Block diagrams: represented as composed MPTs. **Only 3 composition primitives**:



(a) Serial composition: A ∘ B (b) Parallel composition: A ∥ B (c) Feedback composition: feedback(A)

Figure: The 3 Composition Operators in the HBD Algebra

Challenge 2: One graphical diagram, many possible translations:



feedback(feedback(feedback(Constant ∥ Constant1 ∥ Div ∥ Scope)))

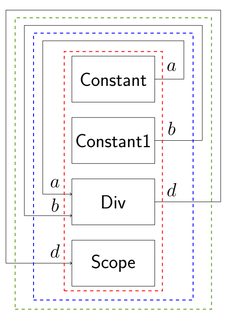
(Constant ∥ Constant1) ∘ Div ∘ Scope

Constant1 ∘ ((Constant ∥ Id) ∘ Div) ∘ Scope

Figure: An inconsistent Simulink diagram

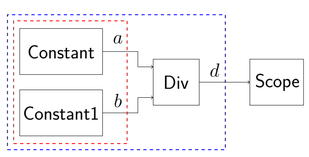
4. Three Strategies for Translating HBDs to Algebraic Terms

- Feedback-parallel translation strategy (`-fp` option)



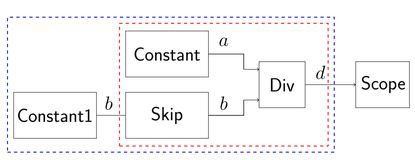
feedback(feedback(feedback(Constant ∥ Constant1 ∥ Div ∥ Scope)))

- Incremental translation strategy (`-ic` option):



((Constant ∥ Constant1) ∘ Div) ∘ Scope

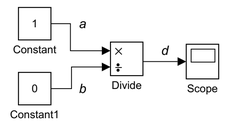
- Feedbackless translation strategy (`-nfb` option):



(Constant1 ∘ ((Constant ∥ Skip) ∘ Div)) ∘ Scope

- All three strategies are implemented in the `simulink2isabelle` translator, and achieve different tradeoffs.

5. Formal Analyzer: Expansion and Simplification



↓ translation

`simulink` "DivIncomp $=$ (Constant(1) ∥ Constant1(0)) ∘ Div ∘ Scope"

↓ expansion

DivIncomp $= \{(\lambda(a,b).b \neq 0) \circ ((\lambda(x,y).(1,0)) \circ (\lambda u.((),())))\} \circ [(\lambda(a,b).1 * a/y) \circ ((\lambda(x,y).(1,0)) \circ (\lambda u.((),())))]$

↓ simplification

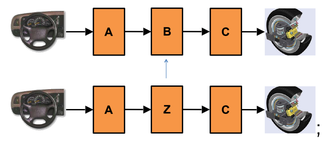
DivIncomp $= \bot$ (meaning that this model is inconsistent)

Challenge 3: Simplification generally involves non-trivial symbolic formula manipulations.

- We implemented fully automatic simplification algorithms on top of the Isabelle proof assistant.
- These generate an atomic MPT ("contract") for the top-level system.

6. RCRS: a Contract-Based Framework with Refinement

- "Horizontal" contracts: MPTs are pairs of pre/post-conditions, e.g., $\{a, b.\ b \neq 0\ \} \circ [a, b \rightsquigarrow a/b]$.
- Used to: (1) check compatibility; (2) compute contract of parent system from contracts of subsystems.
- Refinement ("vertical contract"): allows to replace a component with another while preserving all properties.
  - If $S' \preceq S$ (S' refines S) and S satisfies P, then S' satisfies P
  - If $S' \preceq S$ and $T' \preceq T$, then $S' \otimes T' \preceq S \otimes T$ where $\otimes \in \{\circ, \|, \text{feedback}\}$



;

Figure: Substitutability by refinement: component Z can replace component B if Z refines B

7. Case Study: a Fuel Control System (FCS)

- Benchmark provided by Toyota. Publicly available at: `http://cps-vo.org/group/ARCH/benchmarks`
- Simulink model:
  - 3-level hierarchy
  - 104 blocks: 97 atomic blocks and 7 subsystems
  - 101 links of which 7 feedbacks

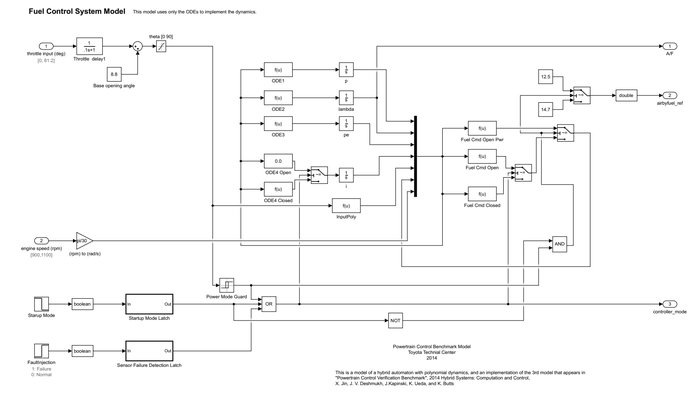


Figure: Model 3: the largest subsystem of FCS

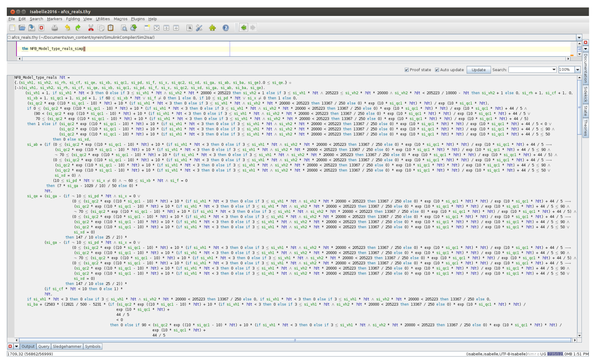


Figure: Screen shot of the auto generated top-level contract for FCS

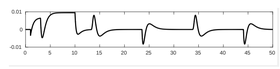


Figure: Simulink simulation of the FCS model

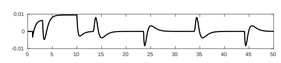


Figure: Simulation of RCRS auto generated code

		FPT	IT		NFBT
			HBD	IO-HBD	
Translation	Translation time (secs)	0.367	0.196	0.194	0.404
	Total algebraic term length (chars)	8969	49241	48569	31856
	Number of terms	2	2	50	64
Simplification and compatibility check	Expansion and simplification time	9139.37	2089.64	1704.206	17.141
	Simplified term printing time	1.152	1.642	1.317	1.894
	Simplified term length	47481	47487	47487	47482

Table: Experimental results

- The FCS Simulink model is proven compatible $\forall dt > 0$, i.e., the model's simplified precondition is satisfiable $\forall dt > 0$ (proved in the Isabelle theorem prover).
- Translation validation: simulation plots obtained from the FCS model using Simulink vs. the RCRS tool are nearly identical, $|\text{error}| \leq 6.1487 \cdot 10^{-5}$.

8. Main Publications

- I. Dragomir, V. Preoteasa, S.Tripakis. *The Refinement Calculus of Reactive Systems Toolset*. Submitted 2017
- V. Preoteasa, I. Dragomir, S.Tripakis. *The Refinement Calculus of Reactive Systems*. Arxiv 2017
- V. Preoteasa, I. Dragomir, S.Tripakis. *Type Inference of Simulink Hierarchical Block Diagrams in Isabelle*. FORTE 2017
- S.Tripakis. *Compositionality in the Science of System Design*. Proc. IEEE 2016
- V. Proteasa, I. Dragomir, S.Tripakis. *A Nondeterministic and Abstract Algorithm for Translating Hierarchical Block Diagrams*. Arxiv 2016
- V. Preoteasa, S.Tripakis. *Towards Compositional Feedback in Non-Deterministic and Non-Input-Receptive Systems*. LICS 2016
- I. Dragomir, V. Proteasa, S.Tripakis. *Compositional Semantics and Analysis of Hierarchical Block Diagrams*. SPIN 2016
- V. Preoteasa, S.Tripakis. *Refinement Calculus of Reactive Systems*. EMSOFT 2014
- S. Tripakis, B. Lickly, T. A. Henzinger, E. A. Lee. *A Theory of Synchronous Relational Interfaces*. ACM TOPLAS 2011