CPS: Breakthrough: Scalable Component-based Model Revision of Cyber-Physical Systems with Separation of Concerns

MICHIGAN STATE UNIVERSITY

Department of Computer Science & Engineering

PI: Sandeep S. Kulkarni

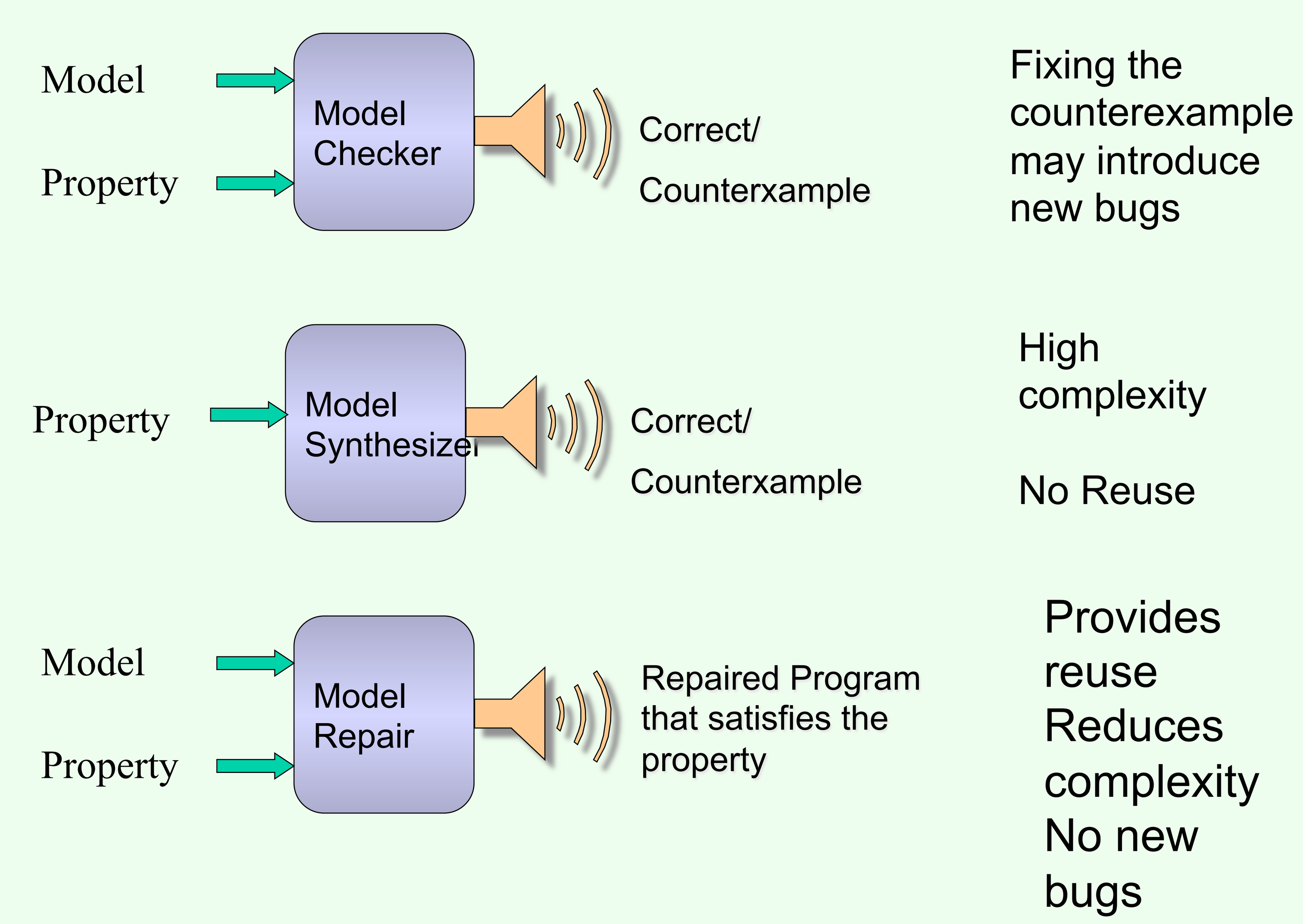
Abstract

This project develops algorithms for revising a given model for cyber-physical system while ensuring that the revised model is correct-by-construction and is realizable in the constraints imposed by the cyber-physical system. It specializes these algorithms in the context of fault-tolerance (with the theory of separation of concerns) and in the context of timed models (with the role of fairness).

This project identifies constraints imposed by the inability to revise some/all physical components and ensure that they are satisfied during revision. It specializes model revision algorithms in two contexts: fault-tolerance and role of fairness during revision. Regarding fault-tolerance, it develops the theory of separation of concerns for cyber-physical systems. This work bridges the gap between fault-tolerance components, control theory and model revision. Regarding fairness, it develops efficient algorithms for revision by using abstraction to model continuous behaviors with discrete behaviors that utilize fairness.

One broad impact of this project is to advance in the fundamental science and technology of cyber-physical systems by developing systematic methods that ensure system correctness during maintenance where the system is revised due to changing requirements and/or environment. The algorithms from this project will provide techniques for providing assurance in automotive and aeronautical systems. In the context where fault-tolerance properties are added, the proposed activities also have the potential to identify missing specifications early and thereby reduce the cost of designing corresponding systems. The proposed activities facilitate in educating graduate students about different tasks involved in providing assurance via component based models and via model revision.

Model checking vs Model Synthesis vs Model revision



Issues in Model Revision of cyber-physical systems

- Physical components may be hard to repair
  - Repair must be done without changing physical components
  - Some actions performed by the physical components and their effect cannot be eliminated.
- Cyber-cyber, Cyber-physical, Physical-Cyber and Physical-Physical interactions present different issues during repair
- Adding fault-tolerance requires modification to several components at once
- Timed analysis may be necessary.

Our Approach

- Identify the impact of C-C, C-P, P-C, P-P interactions and their impact on repair
- Identify complexity barrier(s)
- Develop heuristics that enable efficient repair algorithms that generate efficient programs
- Utilizing untimed fair executions instead of timed executions to reduce the cost of repair.

Selected Results

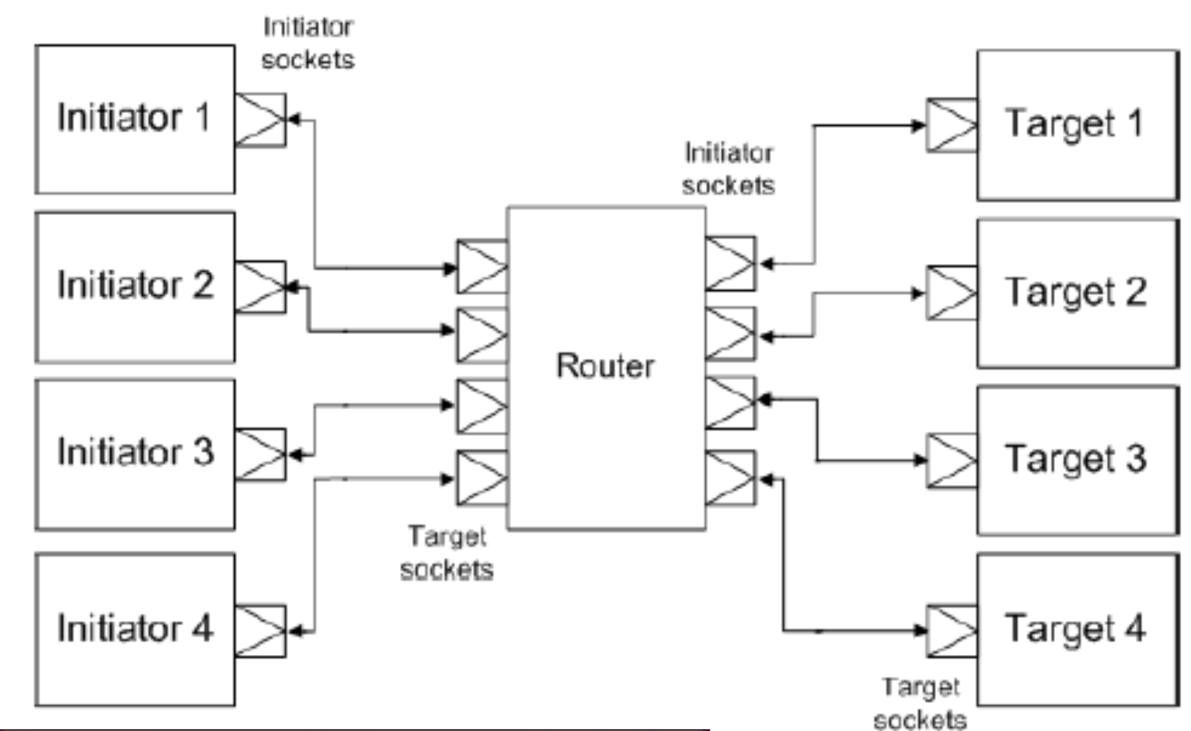
- Development of repair algorithms that incorporate C-C, C-P, P-C, P-P integration
  - NP-completeness of the general problem.
  - Development of Component-based method that enables revision without identifying global state space
  - Heuristics that focus on components that interact already
  - Case studies in communication protocols, railway signal protocols, protocols for vehicle interaction at an intersection

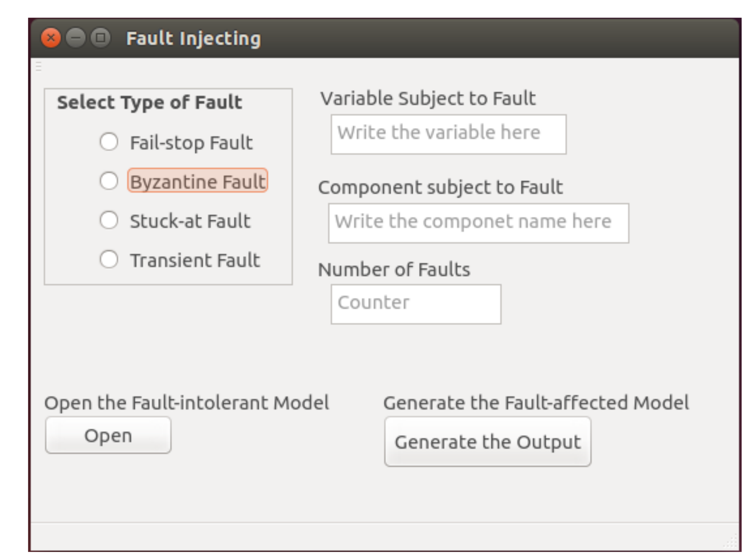
# of train--> Num of signaks	2	3	4	5
5	50			
8	52	53		
10	107	65	62	
12	119	582	54	
15	765	11649	105	138

- Algorithm for Addition of Stabilization, Failsafe and Masking Fault-tolerance with an environment
  - Allows non-terminating environments
  - Allows environments that whose collaboration is essential for providing tolerance
  - Applicable in CPS where physical components can be viewed as environment so their actions are not affected during repair.
  - Case studies in automotive design, traffic control

Selected Results (Continued)

- Application in repair of SystemC TLM programs for fault-tolerance and a tool UFIT to provide automation.
  - SystemC TLM is a de-facto standard in several industries
  - Effect of faults in these programs is not considered in the literature
  - We develop a tool for addition of impact of faults
    - Incorporates permanent faults, transient faults, message faults and timing faults
  - We develop algorithms for model slicing.





Cause	Affected Locations	SPEC 1	SPEC 2	SPEC 3	SPEC 4	SPEC 5	Total Time (ms)
Fault-free model	–	s	s	s	s	s	13.5
Message loss	Initiator to Router	v	v	v	s	v	12.2
	Router to Target	v	v	v	s	v	12.2
	Target to Router	v	v	v	s	v	13.1
	Router to Initiator	v	v	v	s	v	13.0
Fail-stop	Initiator	v	v	v	z	v	13.1
	Router	v	v	v	s	v	13.2
	Target	v	v	v	s	v	14.1
Byzantine	Initiator	s	z	z	s	s	14.0
	Router	s	z	z	s	s	14.3
	Target	s	z	z	s	s	14.4
Stuck-at	Initiator	s	z	z	s	s	12.0
	Router	s	z	s	s	s	12.2
	Target	s	z	s	s	s	12.4

Participants and collaborators

- Yiyan Lin (Graduate Student Michigan State)
- Reza Hajishey (Graduate Student, Michigan State)
- Ling Zhu (Graduate Student, Michigan State)
- Ali Ebnenasir (Faculty, Michigan Tech)
- Borzoo Bonakdarpour (Faculty, McMaster University)

Publications

- Mohammad Roohitavaf, Sandeep S. Kulkarni: Stabilization and Fault-Tolerance in Presence of Unchangeable Environment Actions. ICDCN 2016
- Reza Hajisheykhi, Ali Ebnenasir, Sandeep S. Kulkarni: UFIT: A Tool for Modeling Faults in UPPAAL Timed Automata. NFM 2015: 429-435
- Fathiyeh Faghih, Borzoo Bonakdarpour, Sébastien Tixeuil, Sandeep S. Kulkarni. Specification-based Synthesis of Distributed Self-Stabilizing Protocols, under submission
- Reza Hajisheykhi, Ali Ebnenasir and Sandeep Kulkarni (2014). *Analysis of Permanent Faults in Transaction Level SystemC Models*. Thirteenth International Workshop on Assurance in Distributed Systems and Networks. Madrid, Spain
- Yiyan Lin and Sandeep S. Kulkarni (2014). *Automatic repair for multi-threaded programs with Deadlock/Livelock using maximum satisfiability.*. International Symposium on Software Testing and Analysis, ISSTA '14,. San Jose, CA
- Reza Hajisheykhi, Ali Ebnenasir and Sandeep S. Kulkarni: (2014). *Evaluating the Effect of Faults in SystemC TLM Models Using UPPAAL*. Software Engineering and Formal Methods, 2014. Grenoble, France.