Fraunhofer USA

Center for Experimental Software Engineering

CPS: FRONTIER: COLLABORATIVE RESEARCH: COMPOSITIONAL, APPROXIMATE, AND QUANTITATIVE REASONING FOR MEDICAL CYBER-PHYSICAL SYSTEMS (CYBERCARDIA)

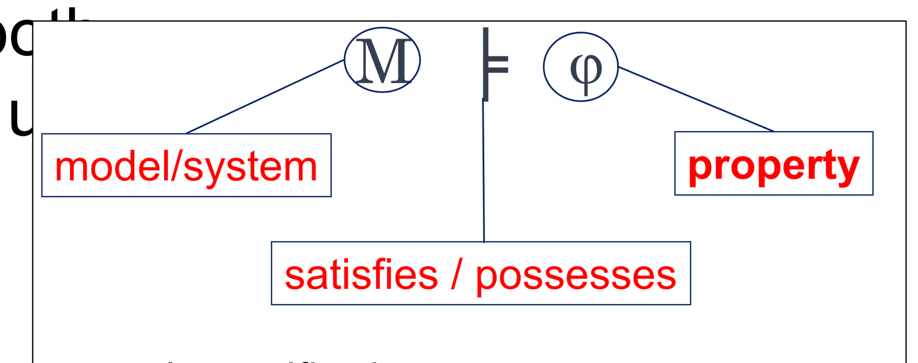
NSF AWARD#: 1446583

CyberCardia

AUTOMATED SPECIFICATION EXTRACTION

PROBLEM

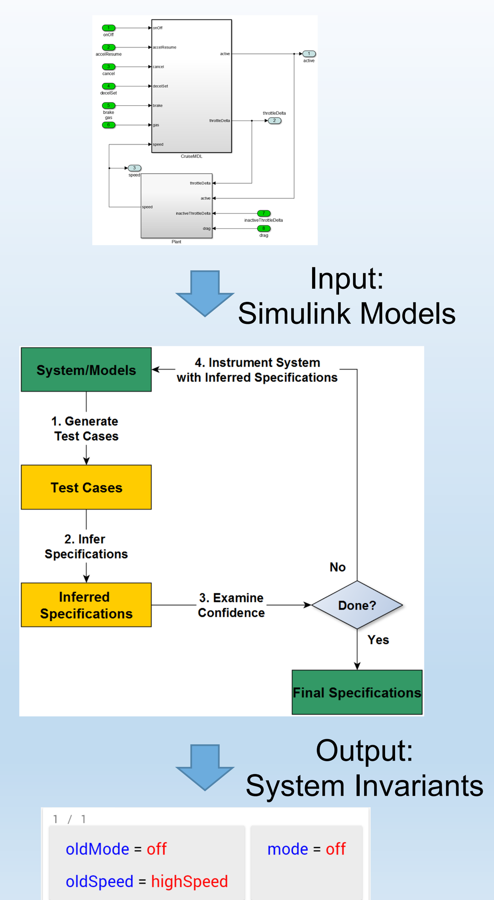
- Deficient or inaccurate specifications can impede bo system development, as well as effective and safe u of the system once it is deployed
- System is usually most up to date artifact
- Can we learn specifications from system?



Example Specification:
*"If the **speed is too low** the **cruise control shall be off**"*

APPROACH

Automated Specification Extraction



We developed framework and associated tooling for the extraction of *system invariants* for Simulink models *from test data* that has been automatically generated from automated executions of the models.

Our framework follows a *test -> infer -> instrument -> retest* paradigm.

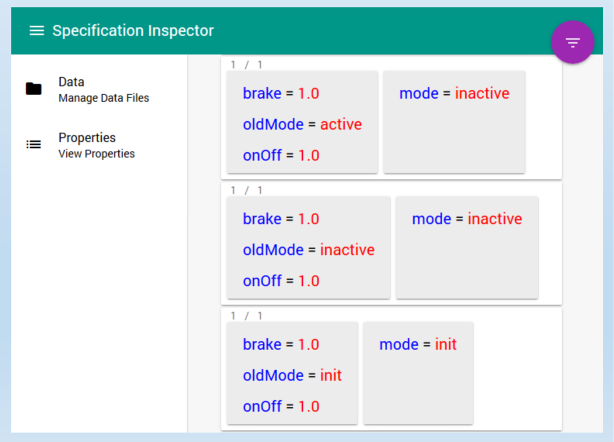
- The *test* data is generated by an automated coverage testing tool for Simulink models called Reactis*. Reactis tries to execute large parts of the systems behavior by targeting coverage metrics such as Branch/Condition/MCDC coverage
- Association rule** mining is used to *infer* invariants from the collected test data
- In order to validate the invariants, the system models are *instrumented* with monitoring logic that checks whether proposed invariants are being maintained
- The *test* generation process is then applied to the instrumented system to generate a new set of tests that actively try to find counterexamples to the invariants, thereby either disproving them, or strengthening the confidence in them. Additional test cases can also find new invariants that were missed in a previous iteration.
- This iterative process is performed several times, until no new invariants are found, and no old ones have been invalidated.
- The resulting invariants of are then proposed as actual behaviors of the system.

* Reactis, http://www.reactive-systems.com/
** SPMF, http://www.philippe-fournier-viger.com/spmf/

Specification Inspector

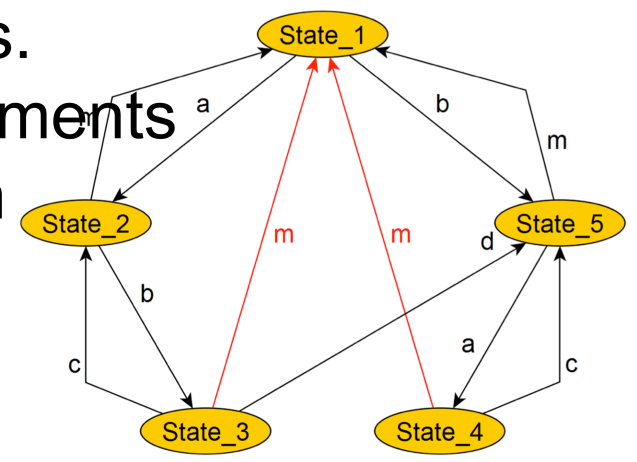
The Inspector helps domain experts to analyze the extracted specifications:

- Options for searching and filtering of the specifications
- Comparison of specifications from different versions of the system to identify deviations



RESULTS & ONGOING WORK

- Applied it to several automotive/medical control system models.
- Extracted specifications lined up very well with existing requirements
- Currently working on larger study with pacemaker models from UPenn CyberCardia team
- Expanding tool for systems implemented in C



TESTING OF AUTONOMOUS SYSTEMS

PROBLEM

- Increasing interest in using autonomous systems in safety critical applications
- Machine learning and non-deterministic control algorithms make it hard or impossible to verify the safety of these systems
- The state space is infinite so testing everything is impossible

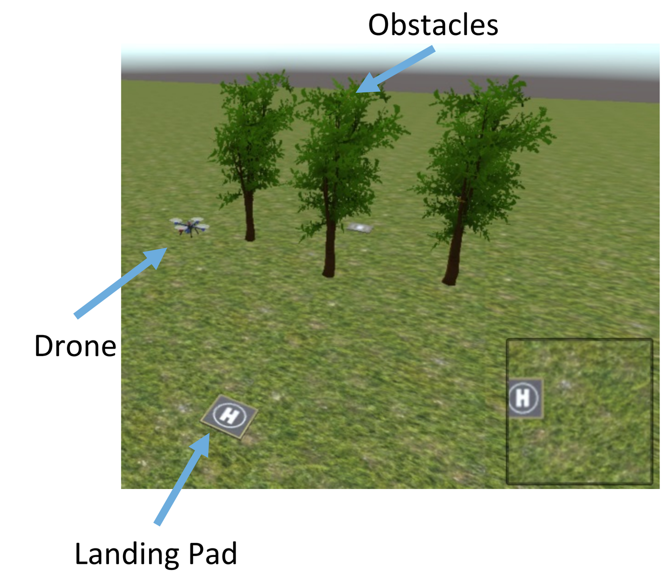
APPROACH

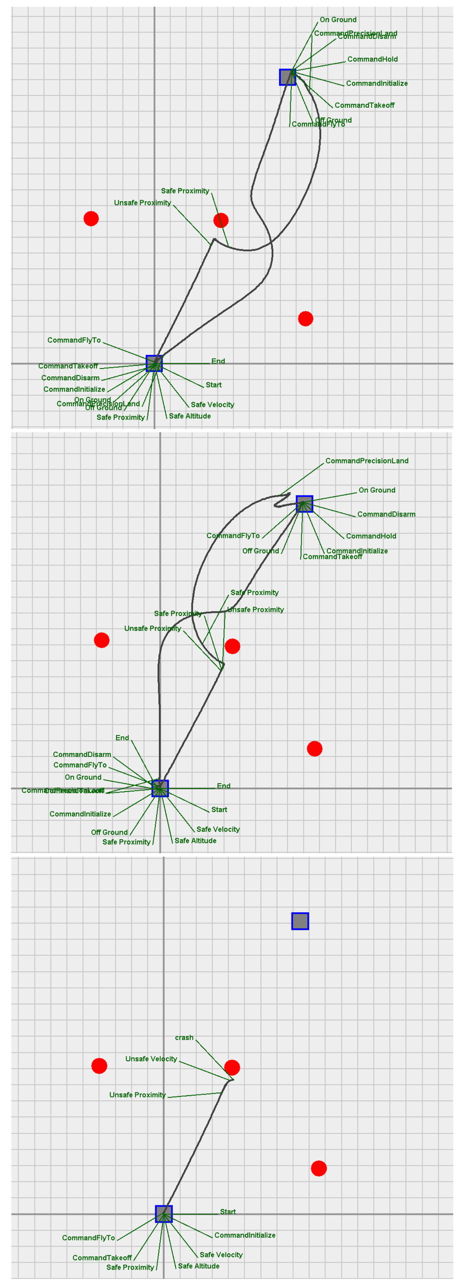
Combine ideas from metamorphic and model-based testing

- Define a model of the operating environment and generate a large number of simulated test cases
- Record the state of the system and identify points where behavior or "decisions" change
- Expect small changes in the scenario to lead to small changes in behavior
- Observing reliable and stable behavior increases trust in the system

Drone test bed

- Simulated quad-copter with lidar and cameras for navigation
- Autonomously navigate from A to B and avoid obstacles on the way
- Testing variations of a simple mission reveals edge cases and unstable behavior





FUTURE WORK

- Define language/format for autonomy requirements
- Smarter test generation. Generate more variations for scenarios where behavior is unstable

Contact: Mikael Lindvall, mikli@cese.fraunhofer.org