Enhancing Reliability for Cyber-physical Systems

Computer Science
- Significant computational resources, such as processing capability and local storage.
- High-level Programming Languages that enable rapid development of mobile CPS: Java, Objective C, C#, NesC.

Communications
- Multiple communication mechanisms, such as WiFi, 3G, EDGE, Bluetooth for interconnecting devices.

System Theory
- Multiple sensory input and output devices such as touch screens, cameras, GPS chips, speakers, microphones, light sensors, proximity sensors.
- End-user maintenance and upkeep.

In July of 2011 two high-speed trains collided in China due to a failure in the software of the signal mechanism.

In 2010 Mars Rover Spirit lost communication forcing NASA to give up the mission with Spirit in May 2011.

How do we automatically develop systems that never crash; always produce the correct results; and is always efficient?
Given a system, how do we automatically find all exceptions, logic errors, inconsistency, inefficiency, etc.?

State-of-the-art Methods
Model checking works by exhaustively searching through all possible system behaviors. It has been successful on finding errors on critical systems (e.g., Intel i7).

http://www.patroot.com

Grand Challenges...

Software Has Errors
- Errors in software cause financial losses of several millions and risk Human Life

Model Checking
- Specification S (Deadlock-freeness, Assertions, Temporal Properties)

Model/Specification refinement

Model Checking $M \models S$

Error Trace

Ok

Non Satisfied

Satisfied