Federal Register Notice: 89 FR 78915, Federal Register :: Networking and Information Technology Research and Development Request for Information on Cyber-Physical Systems Resilience Research, September 26, 2024.

Request for Information on the National Cyber-Physical Systems Resilience Plan

Vinayak S. Prabhu

Indrajit Ray

Indrakshi Ray

Formal Specifications for Resilient CPS Design

Vinayak S. Prabhu
Colorado State University
[redacted]

Indrajit Ray
Colorado State University
[redacted]

Indrakshi Ray
Colorado State University
[redacted]

October 26, 2024

We argue for formal methods based frameworks to facilitate the development of resilient Cyber-Physical Systems (CPS). One of the core elements of formal methods is the rigorous specification of system requirements. Such requirements are specified in frameworks such as timed and untimed temporal logics, automata based frameworks, or process calculii frameworks. In the context of CPS, such frameworks have been augmented to incorporate signals which are functions specifying the evolution of signal values over time. Typically the correctness of systems is guaranteed under certain environment hypotheses — the system is only guaranteed to be correct provided the environment is not *too* antagonistic. Both the environment hypotheses and the system guarantees are given in the frameworks mentioned above in a reactive setting where the system is continually interacting with the environment.

For resilient CPS design [HSK19, Fir19], we need to reason about degraded modes of operation, and we argue the specification frameworks need to be explicit concerning the degraded modes. For example, a component guarantee could be that under environment hypotheses H_1, it can guarantee behavior G_1, and under a degraded mode H_2 where the environment is more hostile (for instance, under increased network latency due to the system being under attack), it can guarantee behavior G_2.

For a modular CPS topological setting where the CPS is a composition of several modules and one/many of the modules may be under attack leading to degraded modes of operation, the research problems include what frameworks need to be developed to specify the hypotheses and the guarantees to facilitate rigorous reasoning about the impact chain of degradation of one component on the other components in the system. If the system were monolithic, one could have the hypotheses in one framework, for instance as parameters such as network delay, the CPU speed of the underlying execution platform, the actuator delay etc, and the system guarantees could be in standard formal methods based frameworks. This would allow one to specify system behaviors under various forms of attacks affecting network delays, CPU throttling due to temperature attacks, actuator attacks etc. However, such a coarse formalism is not amenable towards analysis of how degradation of one component affects other components.

The hypotheses, as well as the guarantees ideally need to be connected to quality of service (QoS) measures of the system. A purely Boolean logical framework (where a property is either satisfied or not) might not be appropriate in many settings; a more desirable quantitative framework would indicate how well a property is satisfied, with the "how well" number being an interpretable QoS measure.

The explicit specification of such hypotheses, guarantees, and a formal framework for analyzing the network effects of component degradation would highlight which are the most attack vulnerable components that need to be fortified the most, or even that the CPS component topology needs to be changed. For example, due to the reactive nature of CPS where components are continually interacting with one another, the initial degradation of component X's operation may get amplified over time due to component X negatively affecting component Y, and the resulting degradation in component Y further degrading X over time, in a feedback loop.

The above example also illustrates the need for reasoning about dynamically changing the topology of software functionality amongst the CPS modules using computational models such as Bigraphs [Mil09].

A design of resilient systems involves 3 Rs: *Recognition*, *Resistance*, and *Recovery*. The property specification frameworks need to be expressible enough to be dynamically expressive in the following manners:

1. Express a monitoring framework that recognizes when a module is under attack.

2. Express the desired property that the effect of quantified short term disruptions lasts for only a quantified bounded time interval, and quantify these effects.

3. Express the transient properties that modules must satisfy in the period when the system recovers or adjusts (eg., when the environment hypothesis switches back from a degraded mode H_2 back to H_1).

4. Express the stabilization properties which specify the system adjusting to various hypotheses changes.

The system architecture also needs to concretize the mechanism for *restoration* – for instance the mechanism for restoring the original system state by restoring/replacing any particular modules that were compromised. There needs to be a rigorous specification and checking of the transient behavior of the system when the compromised modules are being restored/replaced.

A more resilient system provides more functionality than a non-resilient one, and thus the software implementing such a resilient system would necessarily be larger. However, more complex software is more vulnerable to attacks, which in turn makes the system *less* resilient. Thus the desired functionality increase of resilient systems must be implemented with care. For instance, the various modes of component operations under various levels of environment degradation could mathematically correspond to restricted models of computation, which are easier to implement in software, and also easier to formally verify as compared to general software. Such model restrictions, even though they might be inferior to general computational models for performance guarantees, might be preferable as they reduce the attack surface itself. Thus, the model restriction – model performance guarantee trade-off for CPS is also an important consideration for resilient CPS design.

Lastly, a resilient CPS system design needs to account for *human factors*: does the system alert human operators at several levels with sufficient redundancies when modules are suspected of being compromised; are the alerts descriptive and understandable; and is the system robust when multiple human operators attempt to intervene in contradictory manners.

References

[Fir19] Donald Firesmith. System resilience: What exactly is it? Carnegie Mellon University, Software Engineering Institute's Insights (blog), Nov 2019. Accessed: 2024-Oct-26.

[HSK19] Md Ariful Haque, Sachin Shetty, and Bheshaj Krishnappa. Cyber-physical system resilience. In *Complexity Challenges in Cyber Physical Systems*, chapter 12, pages 301–337. John Wiley & Sons, Ltd, 2019.

[Mil09] Robin Milner. *The Space and Motion of Communicating Agents*. Cambridge University Press, 2009.