

Self-Adaptive Systems

Briefing to Army AvMC

JULY 12, 2023

Gabriel Moreno
Assuring Cyber-Physical Systems



Copyright 2023 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM23-0693

Black Friday



Thank you for
we are experiencing
volume of traffic

Please visit
Need any help? Call 800-hp

Abby Rose @rose_cats321 Nov 22
@ultabeauty your website keeps crashing plz help so I can get mascara 😞


Ulta Beauty @ultabeauty Follow

Replying to @rose_cats321

Our website is experiencing high traffic at the moment. We apologize for the inconvenience and expect a fix to get it up and running soon.

4:03 PM - 22 Nov 2018

2 Likes



...ING

...next hour.
...filiated sites.

Careers

Self-Adaptive Software evaluates its own behavior and changes behavior when the evaluation indicates that it is not accomplishing what the software is intended to do, or when better functionality or performance is possible.

[DARPA BAA-98-12]

Self-Adaptation Characteristics

Self-configuration

- Ability to install, replace and configure components and to modify the structure of the system

Self-optimization

- Ability to allocate resources in the most efficient way to achieve the system's goals

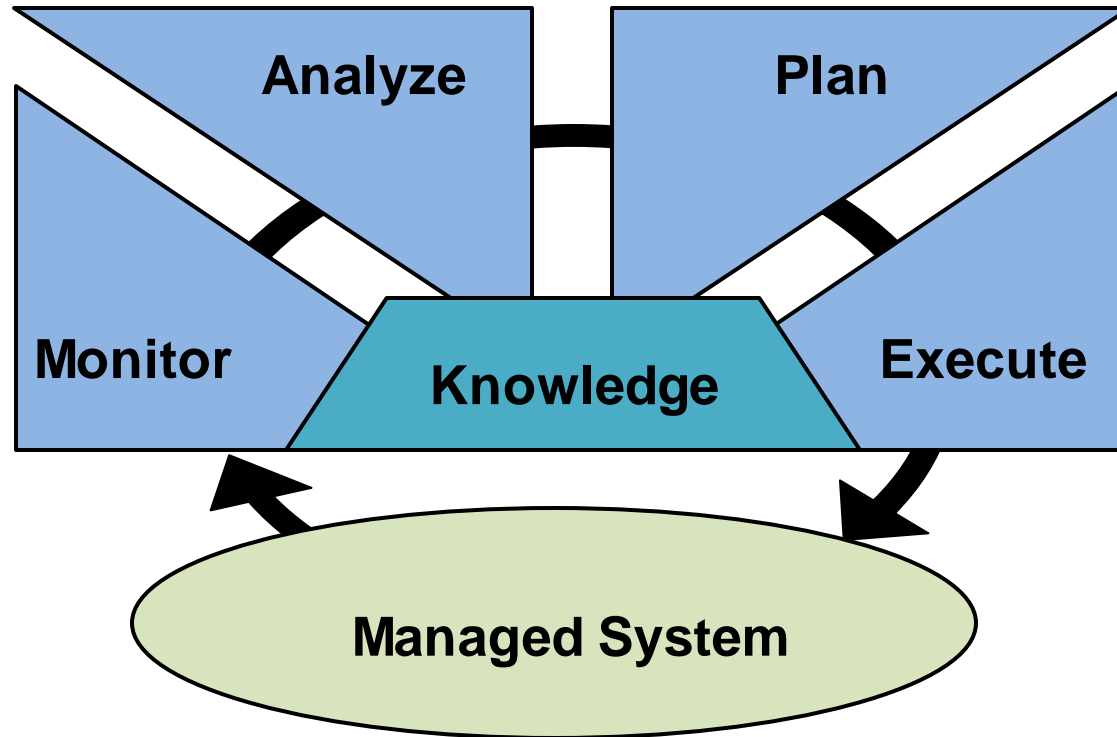
Self-healing

- Ability to detect, diagnose and recover from problems

Self-protection

- Ability to defend against cyber-attacks and to avoid or mitigate security problems anticipated by early warnings

MAPE-K Loop



J.O. Kephart, and D.M. Chess. "The vision of autonomic computing."
Computer 36, no. 1, 2003

Self-Adaptation Approaches

Event-Condition-Action (ECA) Rules

- Simple method for analysis and planning.
- Rules manually created based on heuristics.

<i>When average CPU utilization goes over 60%,</i>	← event
<i>if maximum number of servers not reached ,</i>	← condition
<i>add a server</i>	← action

- Example: Amazon EC2 auto-scaling

Pros and Cons of ECA Rules

Pros

- Each individual rule is easy to understand
- Rules can be added incrementally

Cons

- The management of a goal can be scattered over multiple rules, hindering comprehension
- Multiple rules can interact and have conflicts
- It's difficult to guarantee coverage of the whole adaptation space

H. Ghanbari, B. Simmons, M. Litoiu, & G. Iszlai. "Exploring alternative approaches to implement an elasticity policy." *Proceedings of the International Conference on Cloud Computing*, 2011

Utility-Based Adaptation Decisions

- In general, more than one adaptation option is applicable at a given time.
- The adaptation decision consists on selecting the best.

Utility-Based Decision

Utility: metric that quantifies the goodness of an adaptation alternative.

Usually a function of how the resulting configuration would perform in the current environment.

$$U = f(c, e)$$

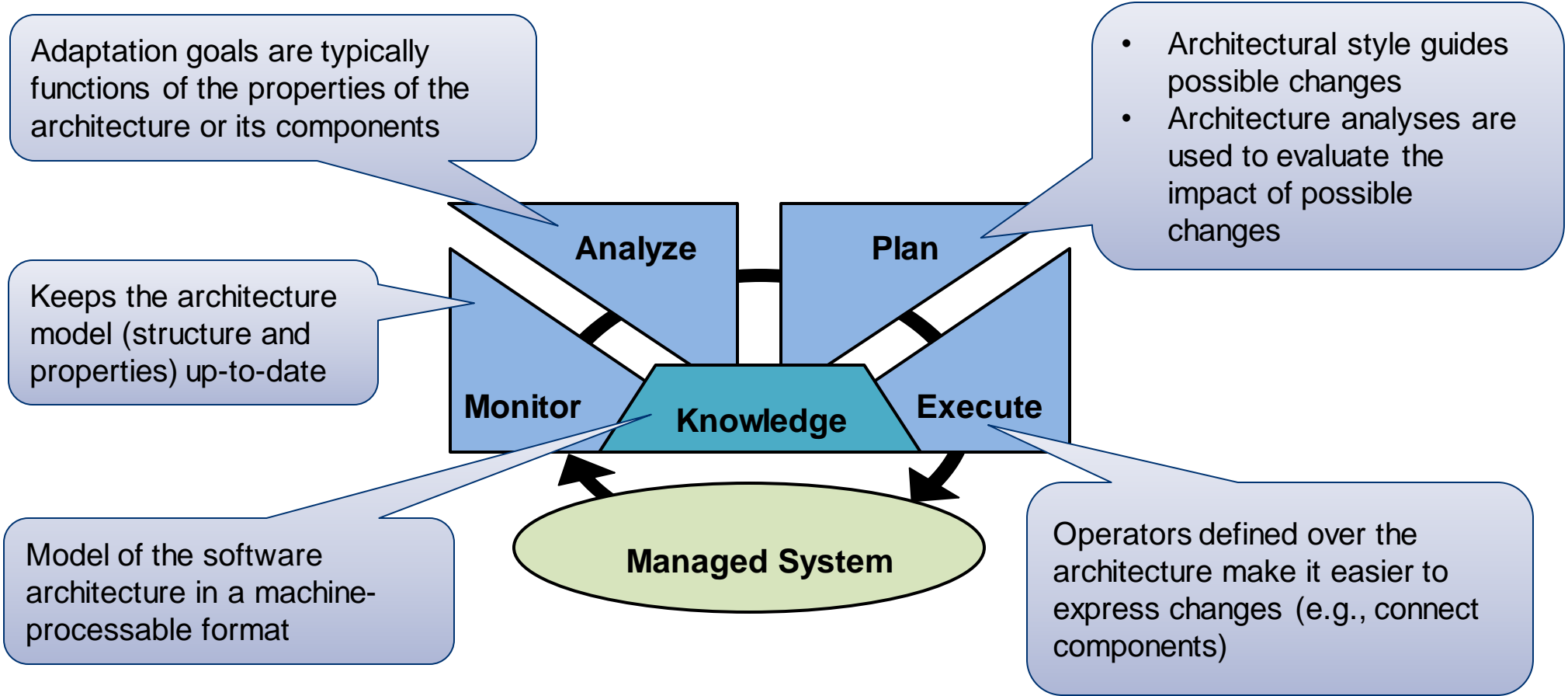
Algorithm: filter out not-applicable alternatives; quantify the utility for each applicable alternative; select the adaptation with the highest utility

Architecture-Based Self-Adaptation

- Software architecture
 - provides a global perspective of the system
 - exposes system properties
 - response time, availability, etc.
 - with the use of architectural styles, imposes restrictions on the possible changes (allows checking the integrity of the adaptation)
- Focuses on the dynamic structure of the architecture (components and connectors)
- Allows using different existing architecture analyses

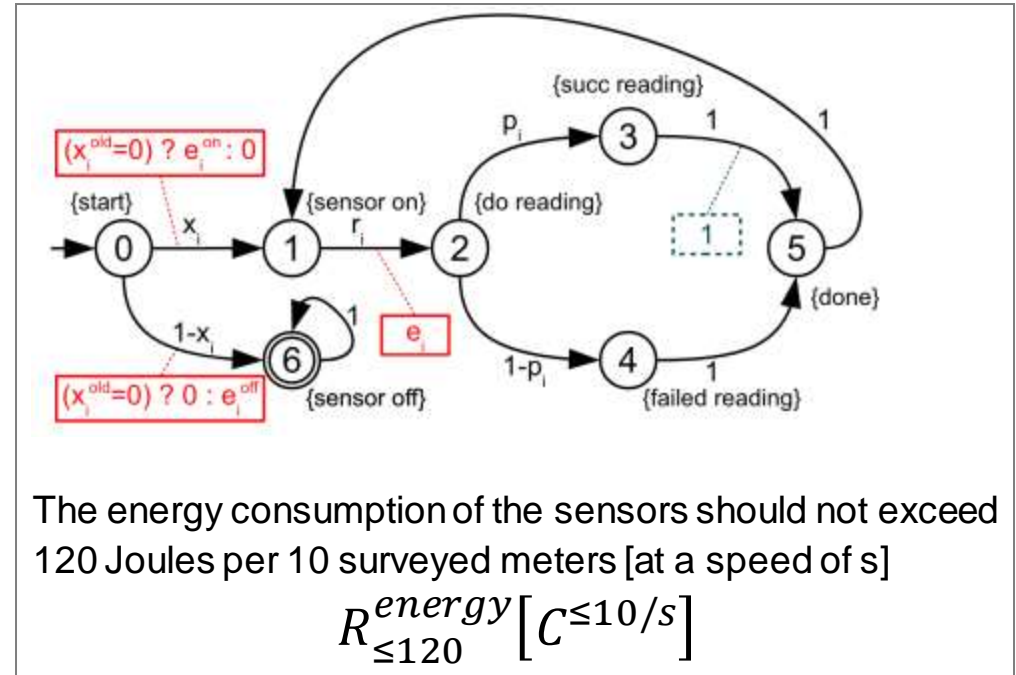
D. Garlan, B. Schmerl, & S.-W. Cheng. "Software architecture-based self-adaptation."
In *Autonomic Computing and Networking*, 2009

Architecture-Based Self-Adaptation



Runtime Quantitative Verification

- The system is mathematically modeled using probabilistic formalisms
 - Discrete-Time Markov Chain (DTMC), Continuous Time Markov Chain (CTMC)
- System requirements are formally expressed
 - Probabilistic Computation Tree Logic (PCTL)
 - Continuous Stochastic Logic (CSL)

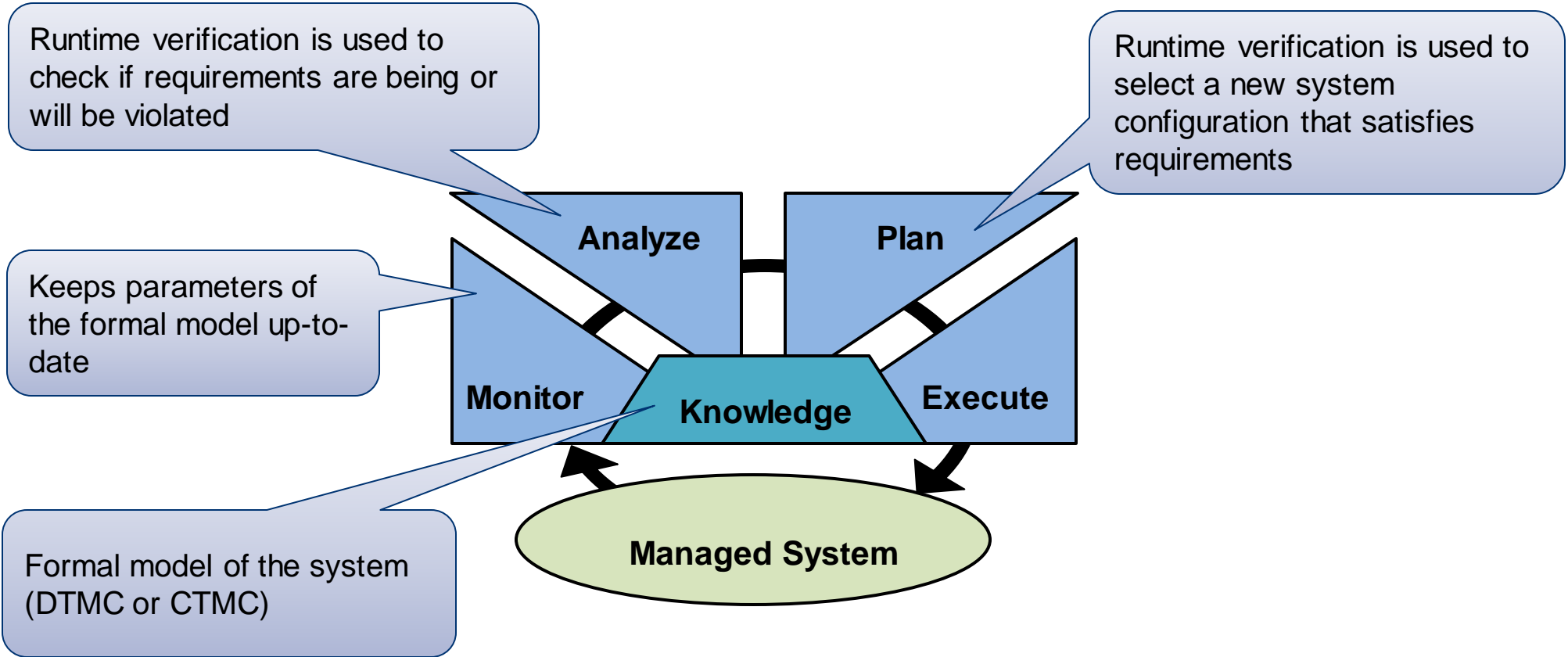


source: [2]

[1] Calinescu, R., C. Ghezzi, M. Kwiatkowska, and R. Mirandola. "Self-Adaptive Software Needs Quantitative Verification at Runtime." *Communications of the ACM*, 2012.

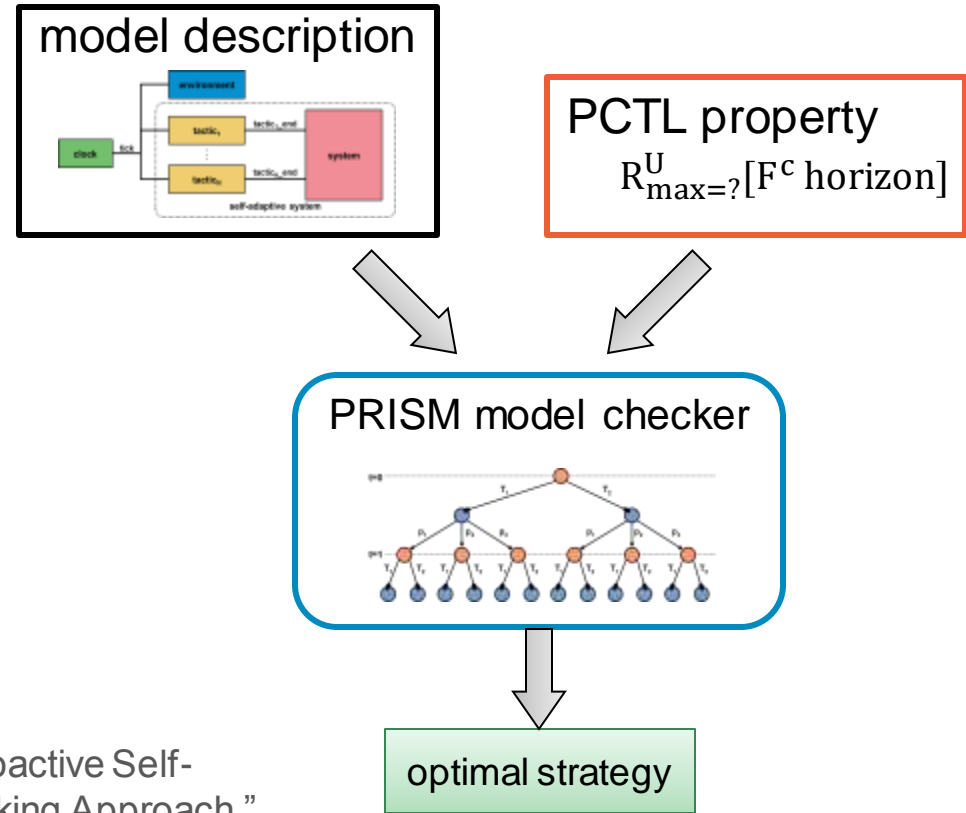
[2] Gerasimou, S., R. Calinescu, and A. Banks. "Efficient Runtime Quantitative Verification Using Caching, Lookahead, and Nearly-Optimal Reconfiguration." *International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2014.

Runtime Quantitative Verification



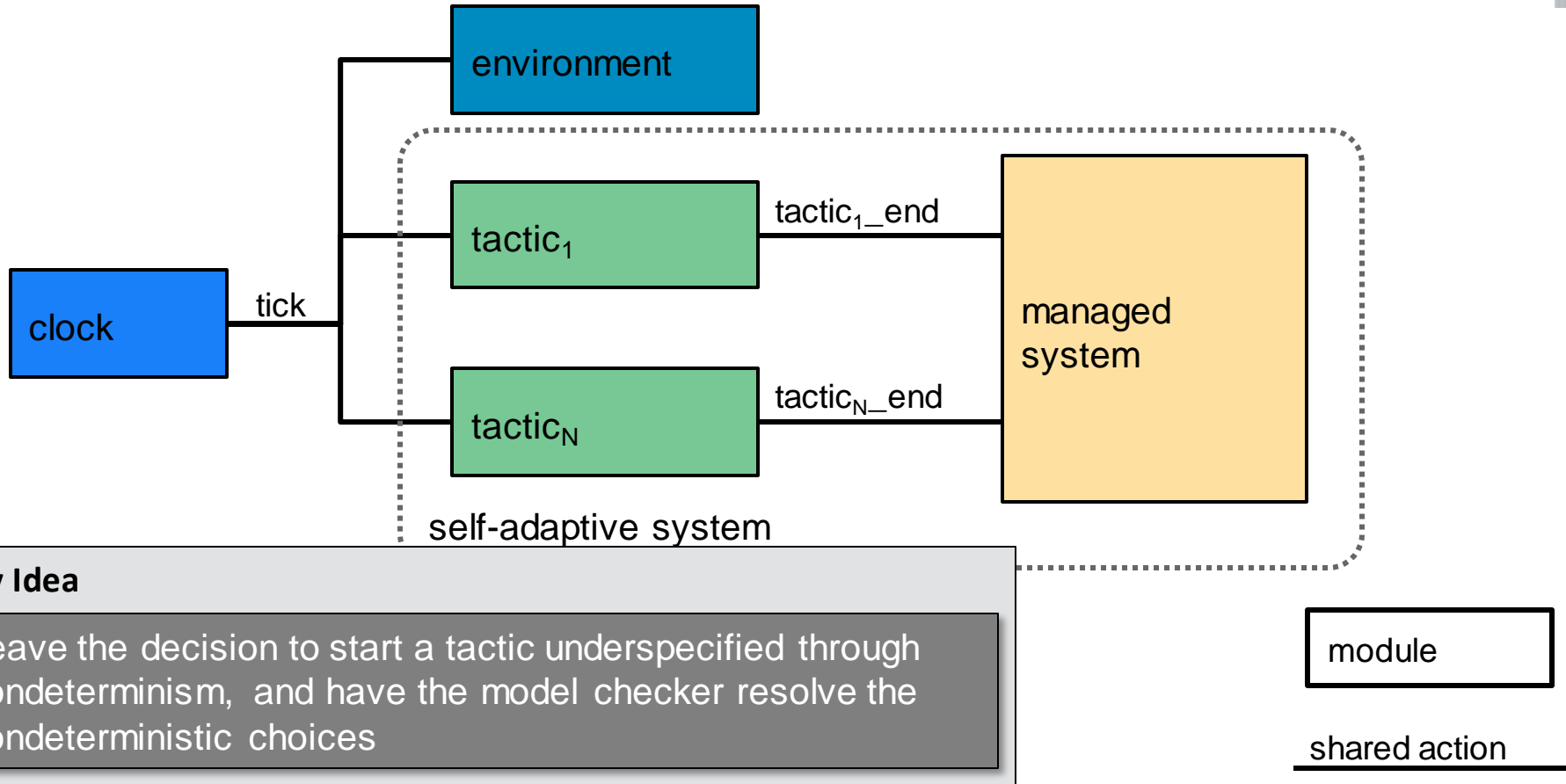
Proactive Latency-Aware Adaptation with Probabilistic Model Checking (PLA-PMC)

- The system *and* the adaptation tactics mathematically modeled using a Markov Decision Process (MDP)
- Utility function formally expressed as a reward property PCTL
- Probabilistic model checker used to synthesize optimal adaptation strategy



Moreno, G.A., J. Cámara, D. Garlan, and B. Schmerl. "Proactive Self-Adaptation under Uncertainty: A Probabilistic Model Checking Approach." *Joint Meeting on Foundations of Software Engineering ESEC/FSE*, 2015.

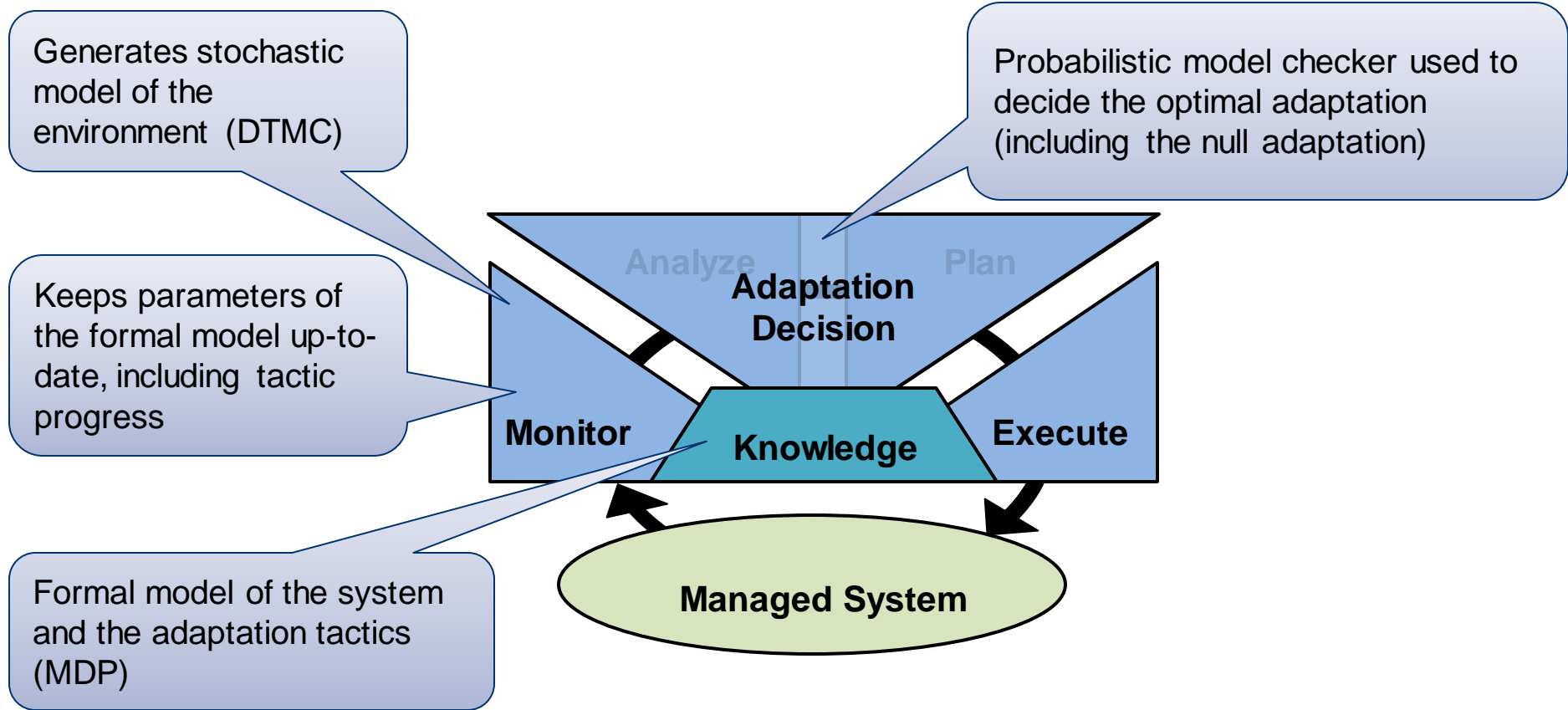
PLA-PMC Model Overview



Key Idea

Leave the decision to start a tactic underspecified through nondeterminism, and have the model checker resolve the nondeterministic choices

PLA-PMC Adaptation Loop



Smart Cyber-Physical Systems

Smart Cyber-Physical Systems

“Cyber-physical systems (CPS) are engineered systems that are built from, and depend upon, the seamless integration of computation and physical components.” [NSF]

Smart Cyber-Physical Systems (sCPS)

Modern CPS need to be smart to be able to operate in dynamic and uncertain environments, be scalable, tolerate threats and perform as good as possible given a situation.

These needs can be fulfilled with self-adaptation techniques.

Challenges in Smart Cyber-Physical Systems

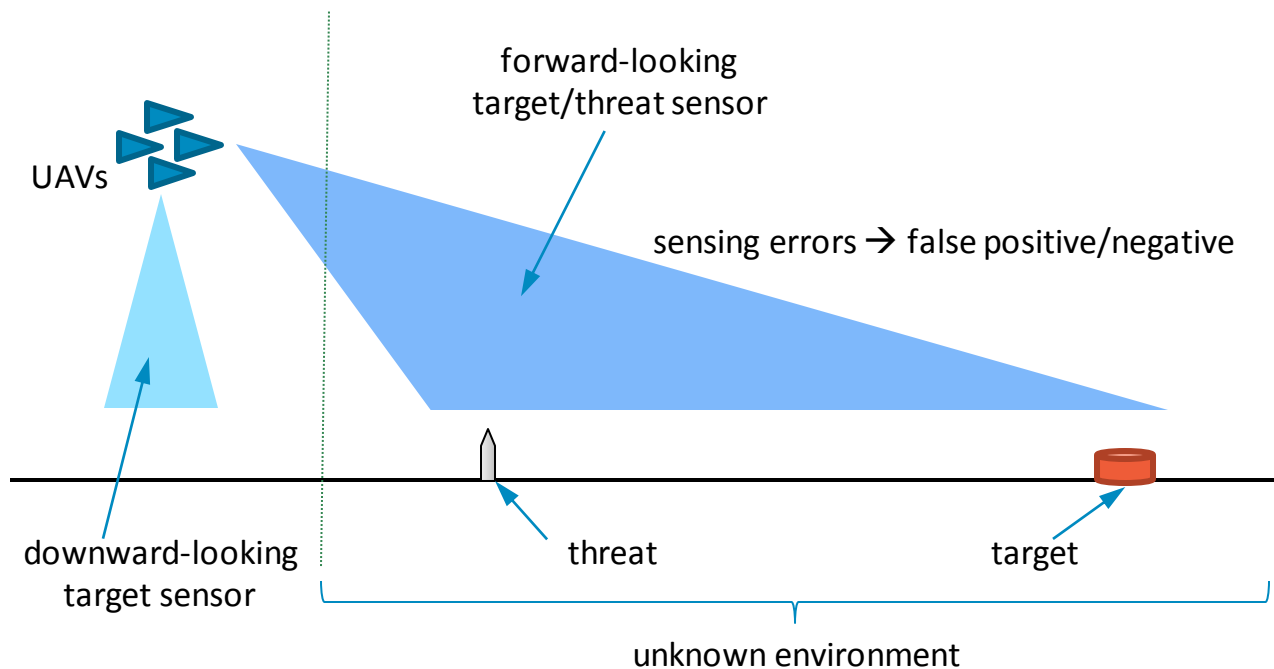
Self-adaptation for sCPS faces particular challenges:

- environment monitoring subject to sensing errors
 - e.g., noisy sensors, limited precision
- adaptation actions that take time
 - e.g., due to physical movement, sensors that recalibrate on power-up
- dire consequences for not adapting in a timely manner
 - e.g., physical damage to system or third party
- incomparable objectives that cannot be conflated into a single utility metric
 - e.g., avoiding an accident vs. providing good service

DARTSim

Simulated autonomous team of unmanned aerial vehicles (UAVs)

Reconnaissance mission in hostile and unknown environment: *fly predefined route to detect targets on the ground while avoiding threats*



*DART = Distributed Adaptive Real-Time

Moreno, G., C. Kinneer, A. Pandey, and D. Garlan. "DARTSim: An Exemplar for Evaluation and Comparison of Self-Adaptation Approaches for Smart Cyber-Physical Systems." In *International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. 2019.

Self-Adaptation in DARTSim

Adaptation Tactics

- climb/descend to change flight level (slow tactic)
- change formation loose/tight (fast tactic)

Adaptation Goal

- detect as many targets as possible
- survive the mission with probability $\geq 90\%$

Realization of sCPS Self-Adaptation Challenges

- Dynamically discovered and uncertain environment
 - Targets and threats localization/number unknown
- Forward-looking sensor is subject to errors
 - The sensor reduces uncertainty but does not eliminate it
- Adaptation actions that require large physical movement take time
 - Limits on climb and descend rate result in adaptation latency
- Not adapting in a timely manner may be catastrophic
 - Loss of UAVs to threat
- Mission survival and target detection maximization not trivial to combine
 - Not possible to assign value to targets to balance with survival (due to unknown number of targets)
 - Simply prioritizing survival results in very conservative behavior (not detecting enough targets)

A Different Form of Utility

probability of having survived up to time t

reward gained at time t

maximize $\sum_{t=1}^H \left(\prod_{i=1}^t s(c_i, e_i) \right) g(c_t, e_t)$

subject to $\prod_{t=1}^H s(c_t, e_t) \geq P$

probability of surviving

survivability requirement

Moreno, G., J. Cámara, D. Garlan, and B. Schmerl. "Flexible and Efficient Decision-Making for Proactive Latency-Aware Self-Adaptation." *ACM Transactions on Autonomous and Adaptive Systems*. 2018.

Adaptive Avionics

Self-Adaptive Avionics Platform

Project out of Germany (University of Stuttgart).

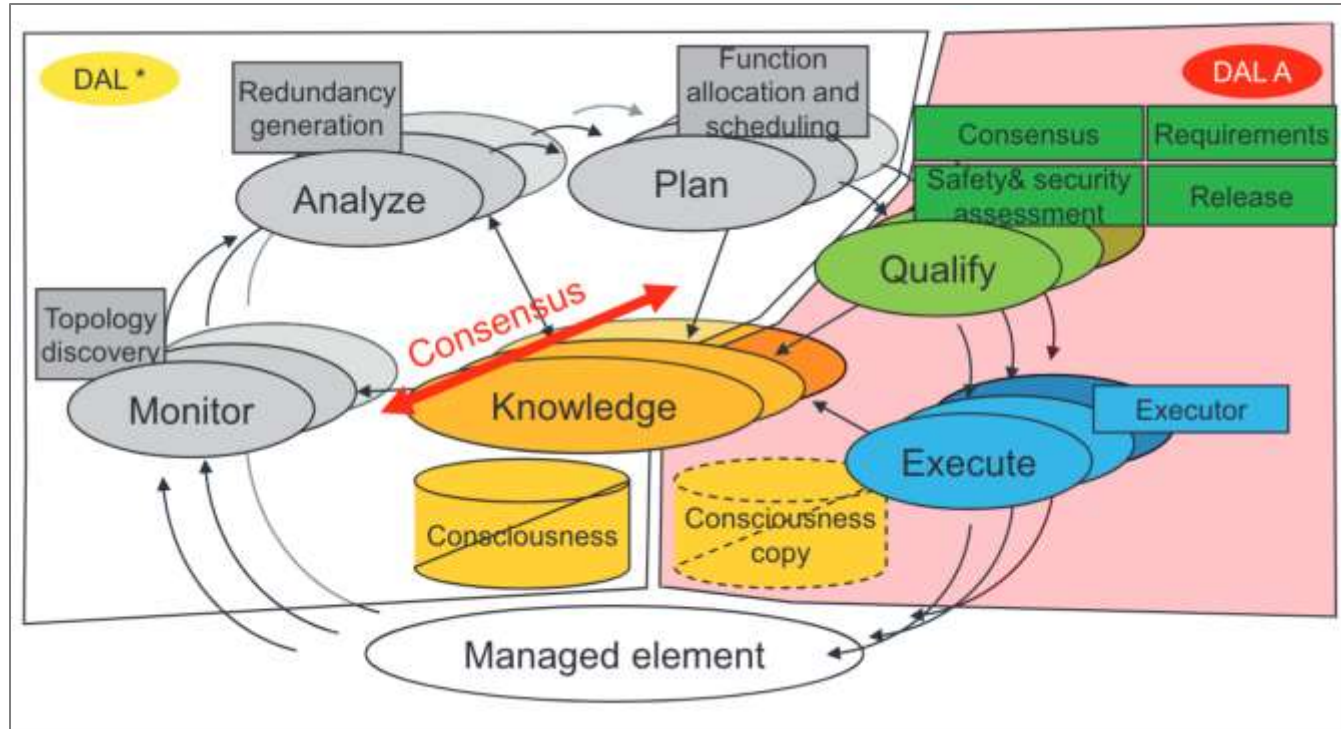
The goal is to reduce the configuration effort, increase fault-tolerance and reduce resource usage in Integrated Modular Avionics.

Plug & Fly Avionics (PAFA) conceptual platform brings the self-* characteristics

- self-configuration, self-optimizing, self-healing.

Annighoefer, B., J. Reinhart, M. Brunner, and B. Schulz. "The Concept of an Autonomic Avionics Platform and the Resulting Software Engineering Challenges." In *International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2021.

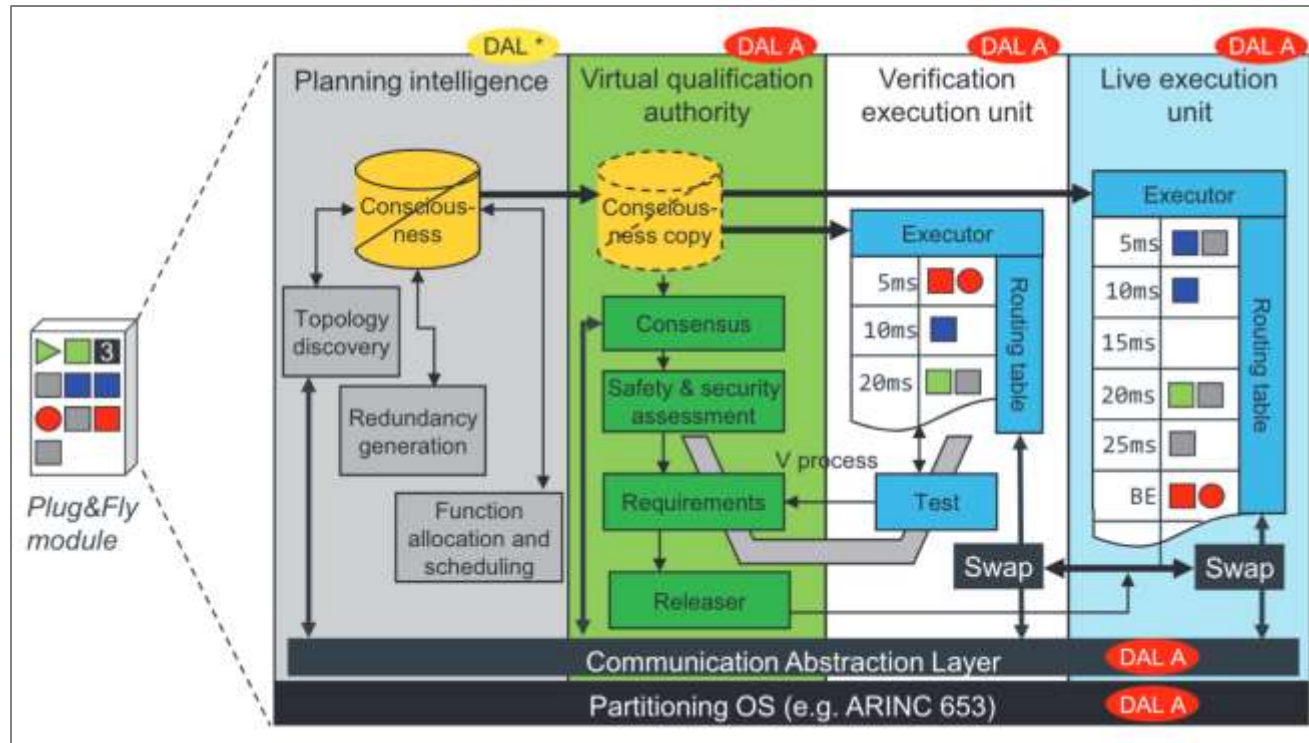
PAFA MAP-QE-K Loop



source: [1]

[1] Annighoefer, B. et al. "The Concept of an Autonomic Avionics Platform and the Resulting Software Engineering Challenges." In *International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2021.

Partition-Based Architecture of PAFA



source: [1]

[1] Annighoefer, B. et al. "The Concept of an Autonomic Avionics Platform and the Resulting Software Engineering Challenges." In *International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2021.

Concluding Remarks

Self-adaptation techniques endow systems with useful autonomous properties

- self-configuration, self-optimization, self-healing, self-*

Although originated in information systems, most current research efforts target cyber-physical systems.

Several decision-making approaches are available to tackle different adaptation decision problems.

Model-based automated analyses are key to realizing advanced forms of self-adaptation.