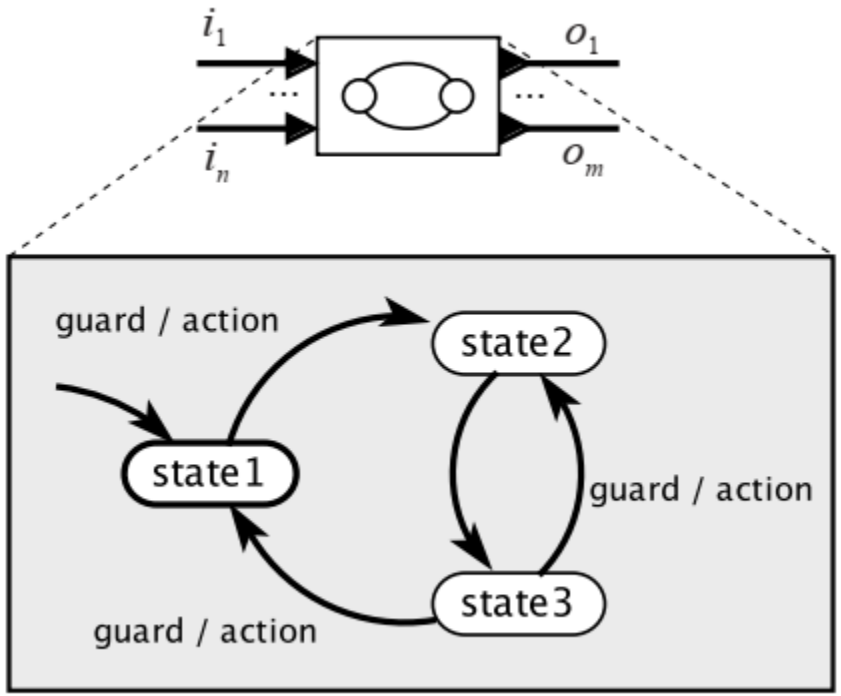
Safety Assurance CPS

Automata

- Cyber-physical systems integrate physical dynamics and computational systems, so they commonly combine both discrete and continuous dynamics.
- In Cyber-Physical Systems (CPS), automata are mathematical models used to describe, simulate, and analyze the behavior of systems that involve both digital (cyber) and physical components.
- These models are particularly useful for systems where software (cyber) interacts with physical processes (physical) in real-time, like autonomous vehicles, robotics, and smart grids.
- Automata help capture how these systems transition between different states and respond to events, inputs, or environmental changes.

Finite State Machine



Automata as a state machine model:

States: different conditions or modes of the system. For instance, an autonomous vehicle might have states like "idle," "moving forward," or "braking.“

Transitions: define how the system moves from one state to another based on specific events or conditions. For example, a transition from "moving forward" to "braking" might be triggered if an obstacle is detected.

Exercise

- Make a state diagram for Elevator system
- Make a state diagram for Robotic Arm

Types of Automata

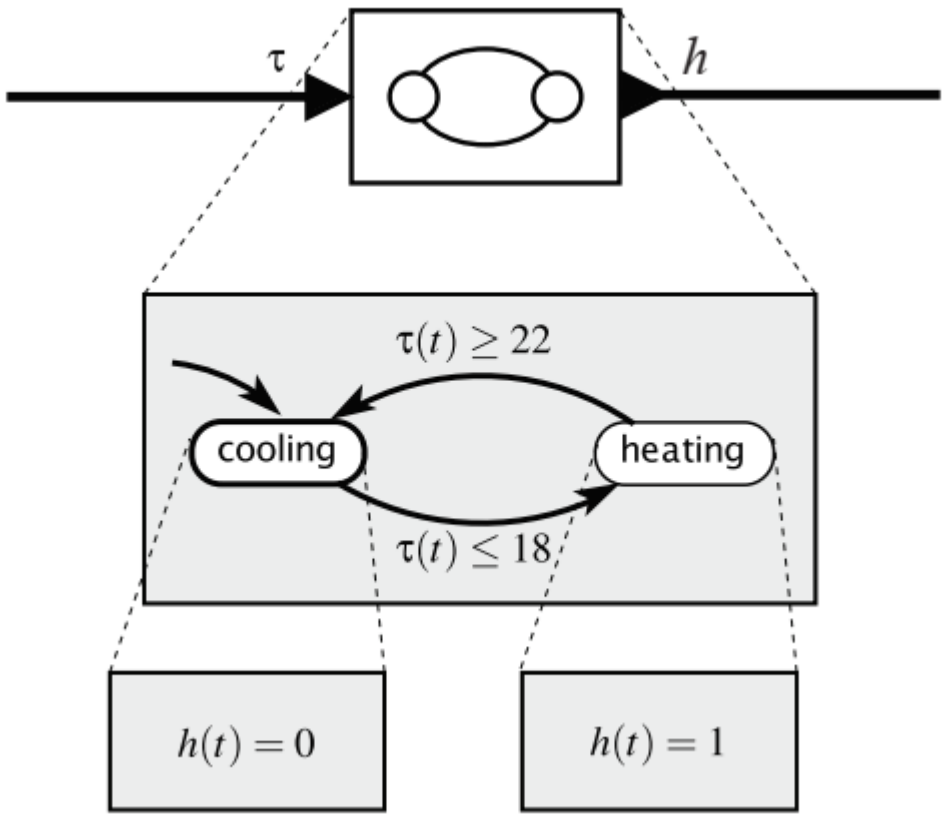
- Finite State Automata (FSA): Used for purely discrete models where the system has a limited number of states and transitions are triggered by specific events.
- Timed Automata: An extension of finite automata that incorporates timing constraints. This is crucial in CPS where real-time performance is important, such as ensuring an action is performed within a strict deadline.
- Hybrid Automata: These are essential in CPS because they model both discrete events (e.g., mode switching) and continuous changes (e.g., acceleration or temperature change). Hybrid automata allow a system to have continuous dynamics in each state, described by differential equations, to reflect the physical components of the system.

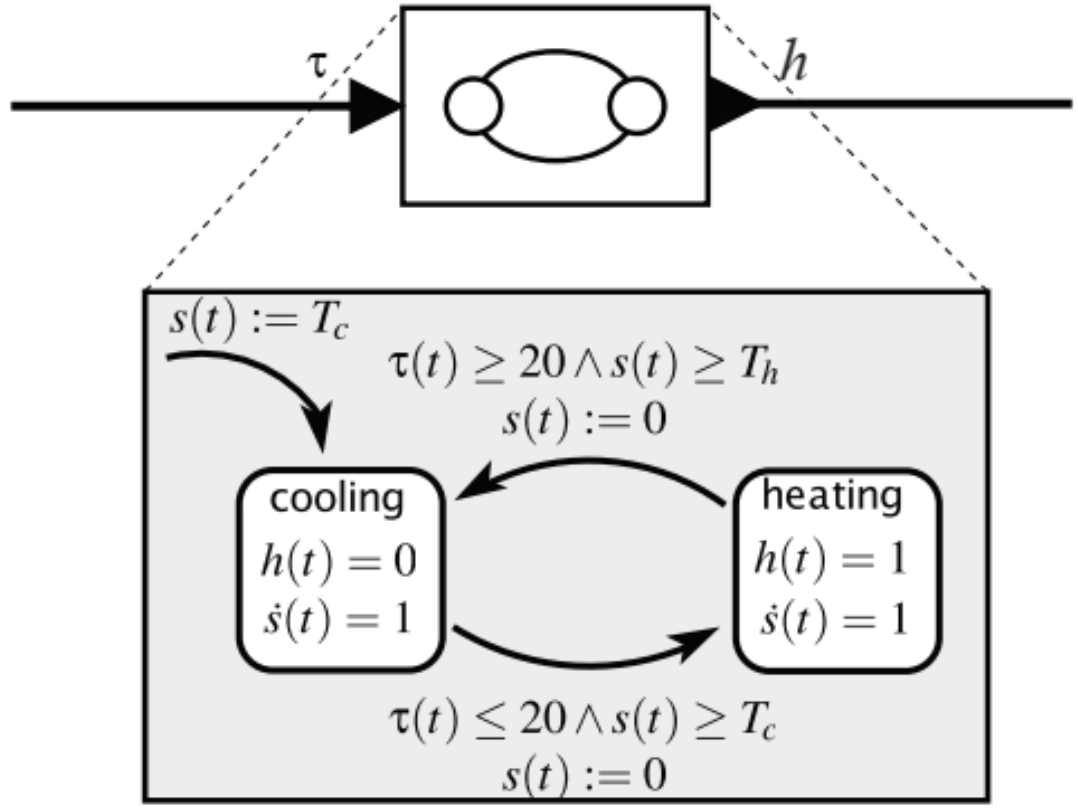
Automata uses in CPS Modeling

- Simulation of system behavior
- Verification system properties
- Analysis of timing constrains
- In an autonomous vehicle, automata can model different operational states like "detect obstacle," "slow down," and "stop." Transitions between these states are triggered by sensor inputs (like detecting an obstacle) or timing constraints (like when a certain time to impact is reached). Hybrid automata can even represent continuous dynamics, such as deceleration, allowing for a realistic model of the vehicle's physical behavior.

Timed Automata

1. States: Represent different configurations or phases of the system.
2. Transitions: Moves between states, which may depend on clock values.
3. Clocks: Real-valued variables that track the passage of time.
4. Clock Constraints: Conditions (like (x > 3) && (x < 5)) associated with transitions that define when a transition can occur based on the clock values.





A thermostat with continuous-time output.

Applications of Timed Automata

- Timed automata are commonly used in scheduling, network protocols, and safety-critical systems where precise timing is essential.
- Verification tools, like UPPAAL, are used to simulate and check that all timing requirements are met, which is vital for real-time embedded systems in CPS.

Hybrid Automata

Hybrid Automata further extend timed automata by adding continuous variables that evolve over time according to differential equations.

This enables hybrid automata to model the interactions between discrete events (like transitions in a digital system) and continuous dynamics (like changes in temperature, velocity, or pressure), which are common in CPS.

Components

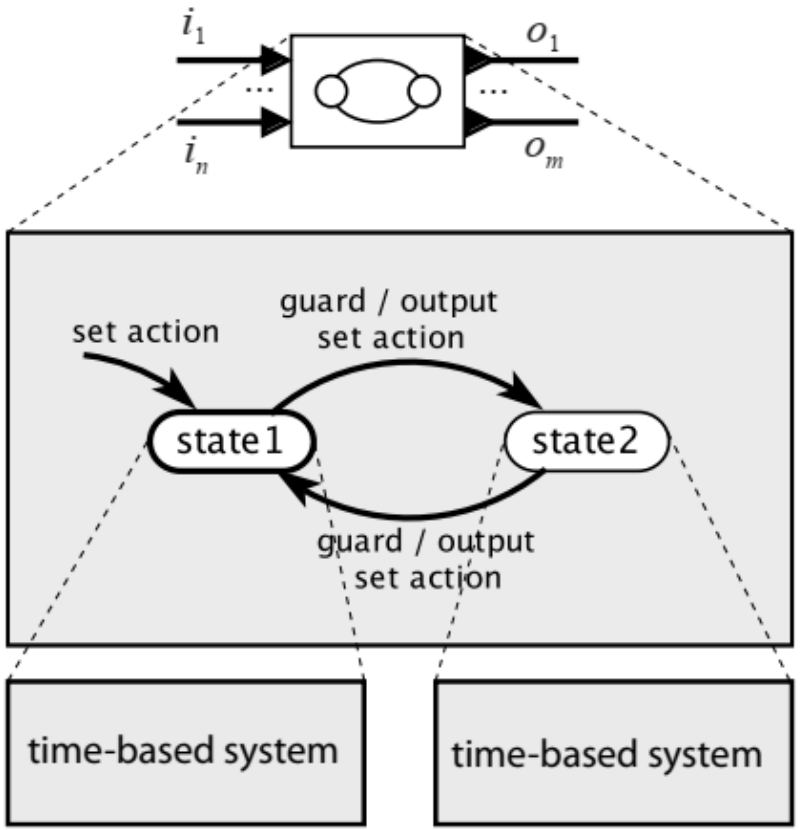
1. States: Represent discrete modes or phases of the system, each with its own dynamics.

2. Transitions: Switches between states that may depend on time, continuous variables, or other conditions.

3. Continuous Variables: Variables that evolve continuously over time according to specified rates or differential equations.

4. Invariants and Guards: Conditions that specify how long a state can be maintained and when a transition should occur based on both discrete and continuous variables.

Notations for Hybrid Systems



Example: Autonomous Drone

States: Climbing, Cruising, Descending
Continuous variables: altitude, vertical velocity
Differential equations of Each States
Transitions: 1. Climbing to cruising while certain conditions
2. Cruising to descending while certain conditions

Guards: Ensuring transition happened with certain conditions only

Summary

- In CPS, timed automata are typically used for systems where the focus is on the scheduling and timing of events, like network communications and control systems.
- Hybrid automata are used in applications where the interaction of digital commands with physical, continuously changing environments is critical.
- Together, they provide a robust toolkit for modeling, verifying, and optimizing the complex behaviors of modern CPS.

Advanced Automata based modeling

- CPS include systems like autonomous vehicles, industrial automation, smart grids, and medical devices, where reliability, safety, and real-time responsiveness are critical.
- Automata-based modeling is especially suitable for CPS because it allows capturing the discrete transitions (software behavior) and continuous dynamics (physical behavior) that these systems typically exhibit.

Advanced Automata types

- **Stochastic Automata**: For systems with uncertain or probabilistic behavior, stochastic automata extend traditional models by associating probabilities with transitions. This is particularly useful in CPS where components might behave unpredictably due to environmental factors or sensor inaccuracies.

- **Networked Automata**: In CPS that involve networked communication (like sensor networks), networked or distributed automata can model multiple interacting subsystems. This approach supports analyzing communication protocols and fault tolerance in distributed settings.

Verification and Analysis

- Reachability Analysis: This involves calculating all possible states the system can reach from a given initial state.
- For CPS, reachability helps ensure the system avoids unsafe states (e.g., collisions in autonomous driving).
- For complex hybrid systems, reachability can be challenging and often requires advanced algorithms or approximations.

Verification and Analysis

- Model Checking: This is a technique to verify that a model satisfies specified properties, typically expressed in temporal logic (such as Linear Temporal Logic, LTL, or Computation Tree Logic, CTL).
- For example, one might specify that an autonomous vehicle must never enter a forbidden area or must eventually reach its destination.

Verification and Analysis

- Simulation and Testing: While formal methods are powerful, they are computationally intensive and may be infeasible for very complex models.
- Simulation, guided by the automata model, can help analyze system behavior under specific scenarios, which is useful for detecting potential issues without exhaustive formal analysis.

Stability under slow switching for CPS

- In Cyber-Physical Systems (CPS), many systems are modeled as hybrid systems that consist of both continuous dynamics (described by differential equations) and discrete events (such as switching between different modes or controllers).
- In such systems, stability under slow switching refers to conditions under which the system remains stable when switching between different modes of operation happens infrequently enough.
- This concept is particularly important in switched systems and mode-based control systems, where the system can switch between different subsystems or controllers, each with its own set of dynamics.
- The challenge is to ensure that the overall system remains stable despite these switches, especially when the individual subsystems are stable.

Performance under Packet drop and Noise for CPS

- In Cyber-Physical Systems (CPS), performance can be significantly affected by packet drops and noise, particularly in systems where the control and communication processes are interconnected via a network.
- This situation is common in applications such as industrial automation, autonomous vehicles, and smart grids, where sensors, controllers, and actuators communicate over wireless or wired networks.
- Understanding how packet drops and noise affect the system's performance and designing strategies to mitigate these effects are crucial for ensuring reliable and safe operations.

Packet drops in CPS

- Sensor to controller packet drops
- Controller to actuator packet drops
- Controller performance degradation
- Stability issues
- Increased latency
- Error propagation

Mitigation strategies for packet drops

- Redundancy: Duplicate packets using multiple channels
- Packet drop compensation: actions based on drops
- State estimation
- Delay compensation
- Various protocols

Noises in CPS

- In CPS, noise refers to unwanted disturbances or variations in signals, which can affect both sensor measurements and communication channels.
- Noise can degrade system performance by introducing errors into the data used for decision-making, leading to incorrect control actions.
- Sensor Noise
- Actuator Noise
- Communication Noise

Impact of Noise

- Reduced control accuracy
- State estimation error
- Instability
- Delayed convergence

Mitigation strategies of noise

- Noise filtering
- Robust control design
- Adaptive control
- Error correction techniques
- Sensor fusion

Reachability Analysis of CPS

- Reachability analysis is a crucial technique in the verification and validation of cyber-physical systems (CPS).
- It involves determining the set of states a system can reach from a given initial state or set of states over time, considering the system's dynamics, inputs, and potential disturbances.
- This analysis is essential for ensuring that CPS operate safely and meet their performance requirements, especially in safety-critical applications like automotive systems, robotics, and medical devices.

What is Reachability Analysis?

1. Defining the State Space: Identifying all the variables that describe the system's state (both continuous and discrete).
2. Modeling System Dynamics: Representing how the state evolves over time, which may involve:

- Differential equations for continuous dynamics.
- Transition relations for discrete events.

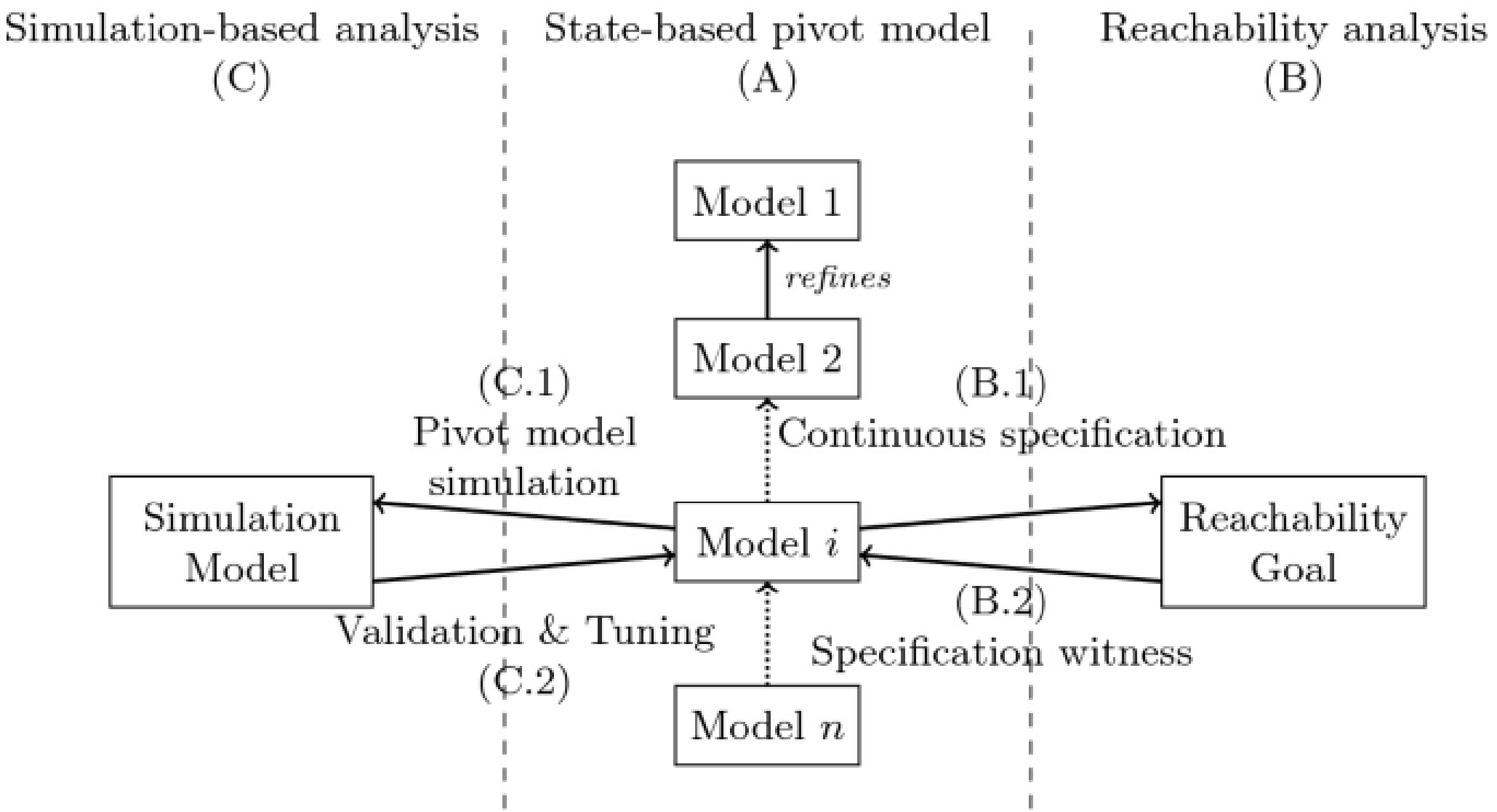
What is Reachability Analysis?

3. Identifying Initial Conditions: Specifying the initial state or set of states from which the analysis starts.

4. Computing Reachable States: Using algorithms to explore the state space and identify all states that can be reached within a certain time frame.

Importance of Reachability Analysis

1. Safety Verification: Ensures that the system does not reach unsafe states, which is essential in applications like autonomous vehicles, industrial control systems, and medical devices.
2. Performance Evaluation: Assesses whether the system meets performance criteria, such as timing requirements or resource constraints.
3. Fault Detection: Helps identify potential faults or erroneous behaviors by exploring how disturbances or unexpected inputs can lead to undesirable states.
4. Design Optimization: Provides insights into system behavior, which can guide design improvements and optimizations.


Simulation-based analysis
(C)
State-based pivot model
(A)
Reachability analysis
(B)
Model 1
refines
Model 2
(C.1)
Pivot model
simulation
(B.1)
Continuous specification
Simulation
Model
Model i
Reachability
Goal
(B.2)
Validation & Tuning
(C.2)
Specification witness
Model n

Types of Reachability Analysis

1. Symbolic Reachability Analysis:

- Uses symbolic representations (like binary decision diagrams or polynomials) to handle high-dimensional state spaces.

2. Numerical Reachability Analysis:

- Employs numerical methods to simulate the behavior of continuous systems, computing reachable sets over time.

3. Abstraction-based Reachability Analysis:

- Simplifies the system model by abstracting away certain details to reduce complexity while preserving essential behaviors.

Steps in Reachability Analysis

1. Model the System: Create a formal model of the CPS, typically using frameworks like hybrid automata, timed automata, or state-space representations.

2. Define Initial States: Specify the initial state(s) from which the reachability analysis begins.

3. Partition Time: Divide the time into intervals to incrementally compute reachable states.

4. Compute Reachable Sets:

- For each time interval, calculate the reachable states using the system's dynamics. This can be done through:
- Flow Pipe Construction: Determine the flow pipes that represent all possible states the system can occupy.
- Simulation: Execute simulations to explore how the state evolves over time.

5. Analyze Results:

- Examine the reachable states to verify safety properties, performance metrics, and identify potential issues.

Example: Reachability Analysis in Autonomous Vehicles

- Consider an autonomous vehicle that must navigate through an environment with obstacles. The reachability analysis process might involve:

- 1. Modeling the Vehicle Dynamics: Using a hybrid automata model that captures both the discrete states (e.g., accelerating, braking, turning) and continuous states (e.g., position, speed).
- 2. Defining Initial Conditions: Specifying the vehicle's initial position, speed, and direction.

- 3. Computing Reachable Sets: Determining the possible positions and speeds the vehicle can reach within a certain time frame, given potential inputs (e.g., acceleration, steering angles) and disturbances (e.g., road conditions).
- 4. Verifying Safety: Ensuring that the reachable sets do not overlap with the positions of static or moving obstacles, thereby confirming that the vehicle can navigate safely through its environment.

Summary

- Reachability analysis is a fundamental technique for ensuring the safety, reliability, and performance of cyber-physical systems.
- By determining the set of reachable states from an initial configuration, engineers can verify that a system will operate correctly under various conditions and inputs.
- While challenges exist, advancements in analysis techniques and tools continue to enhance the capabilities of reachability analysis, making it an indispensable part of CPS development and verification.

Flow pipe construction for cyber physical systems

- Flow pipe construction is a key method in the analysis of cyber-physical systems (CPS) with continuous dynamics, particularly those that are modeled by hybrid automata.
- The term "flow pipe" refers to a set of reachable states of a system over time, given an initial state or set of states.
- By constructing flow pipes, engineers can determine the range of possible behaviors of a CPS, which is essential for verifying safety and performance requirements.

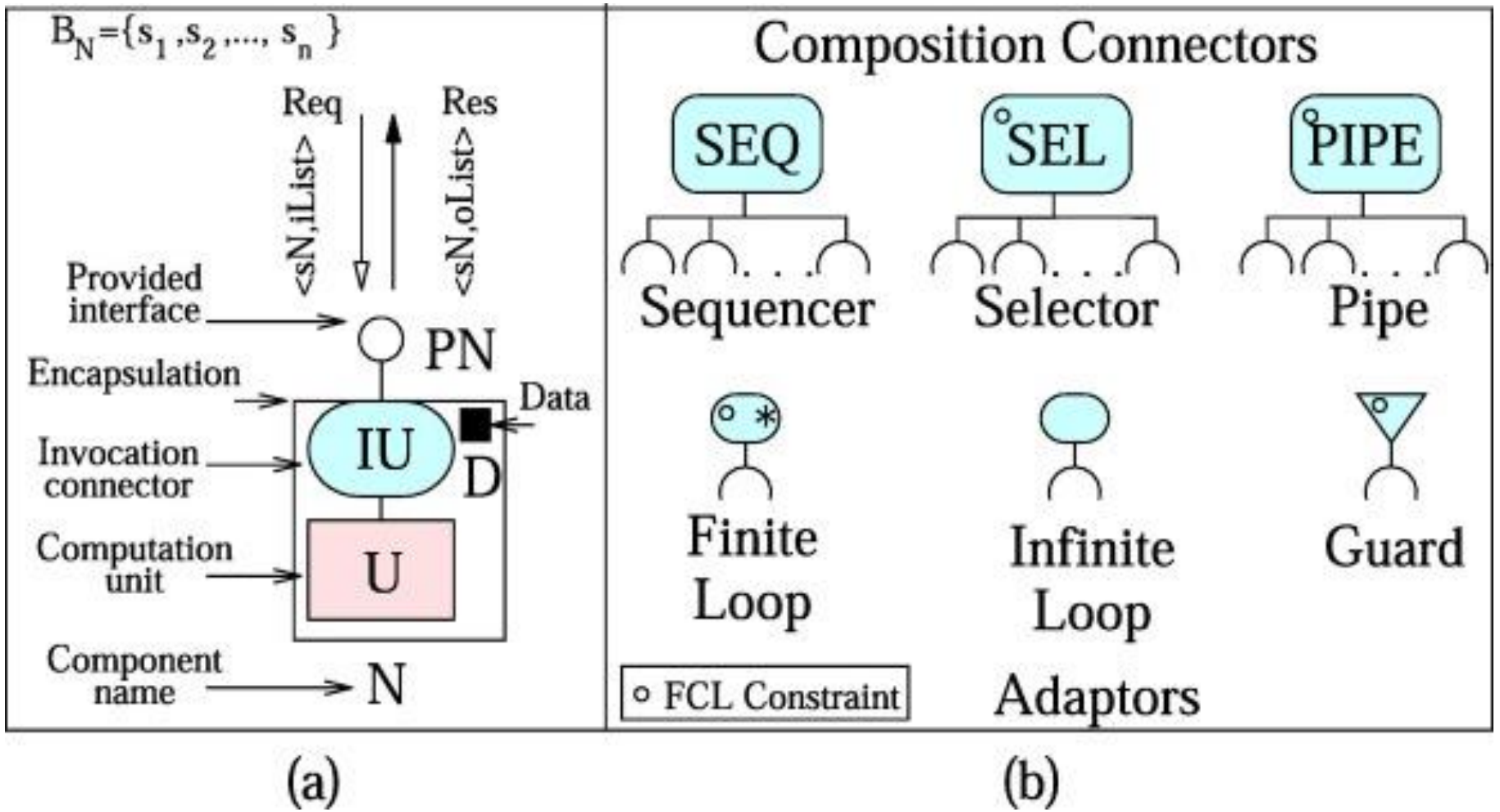
What is Flow Pipe Construction?

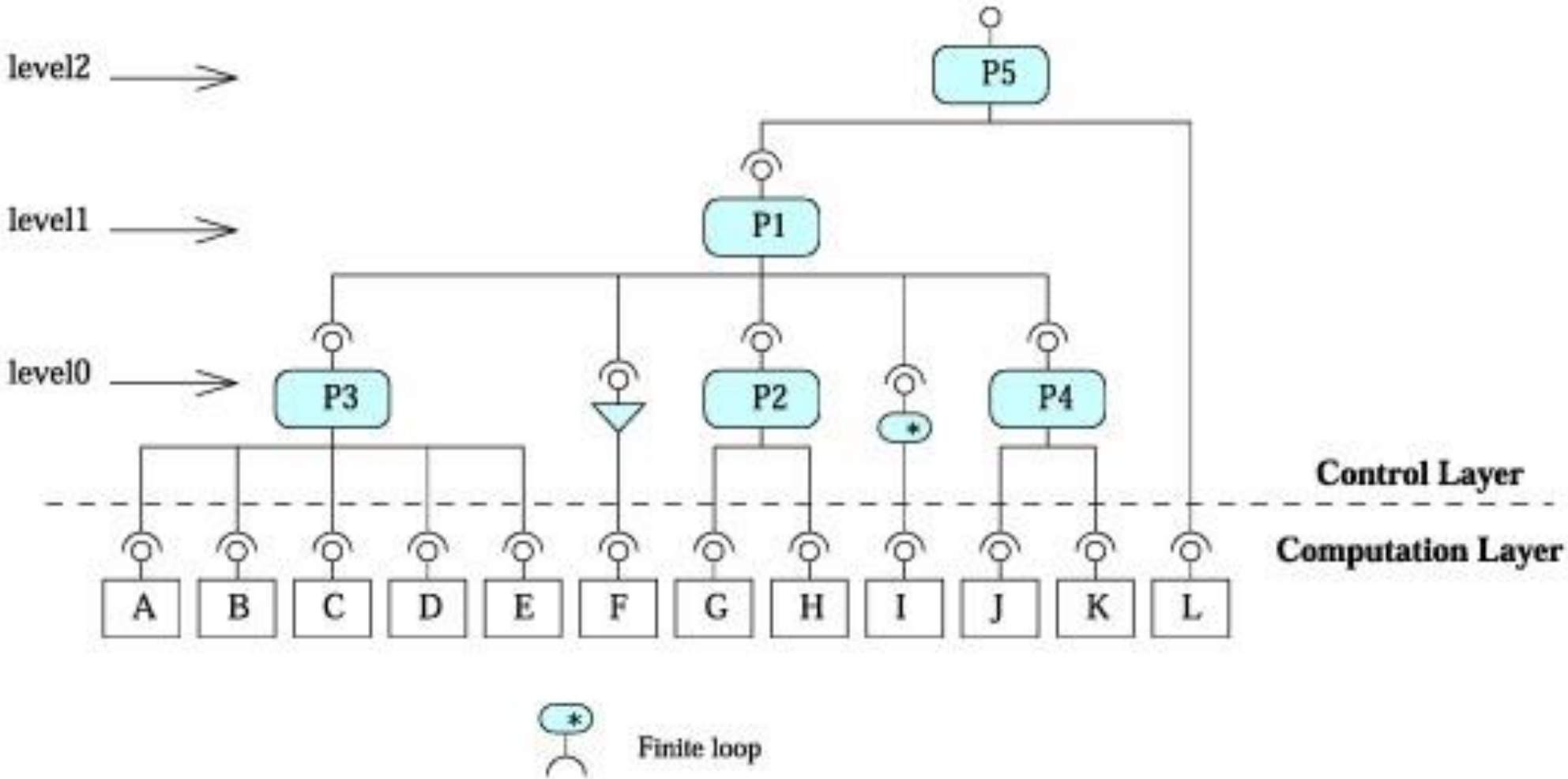
- In essence, flow pipe construction is a way to compute the reachable set of states for a system over time. It provides a sequence of "flow pipes" each representing the set of all possible states the system could occupy within a specific time interval.
- By analyzing this sequence of reachable states, engineers can:

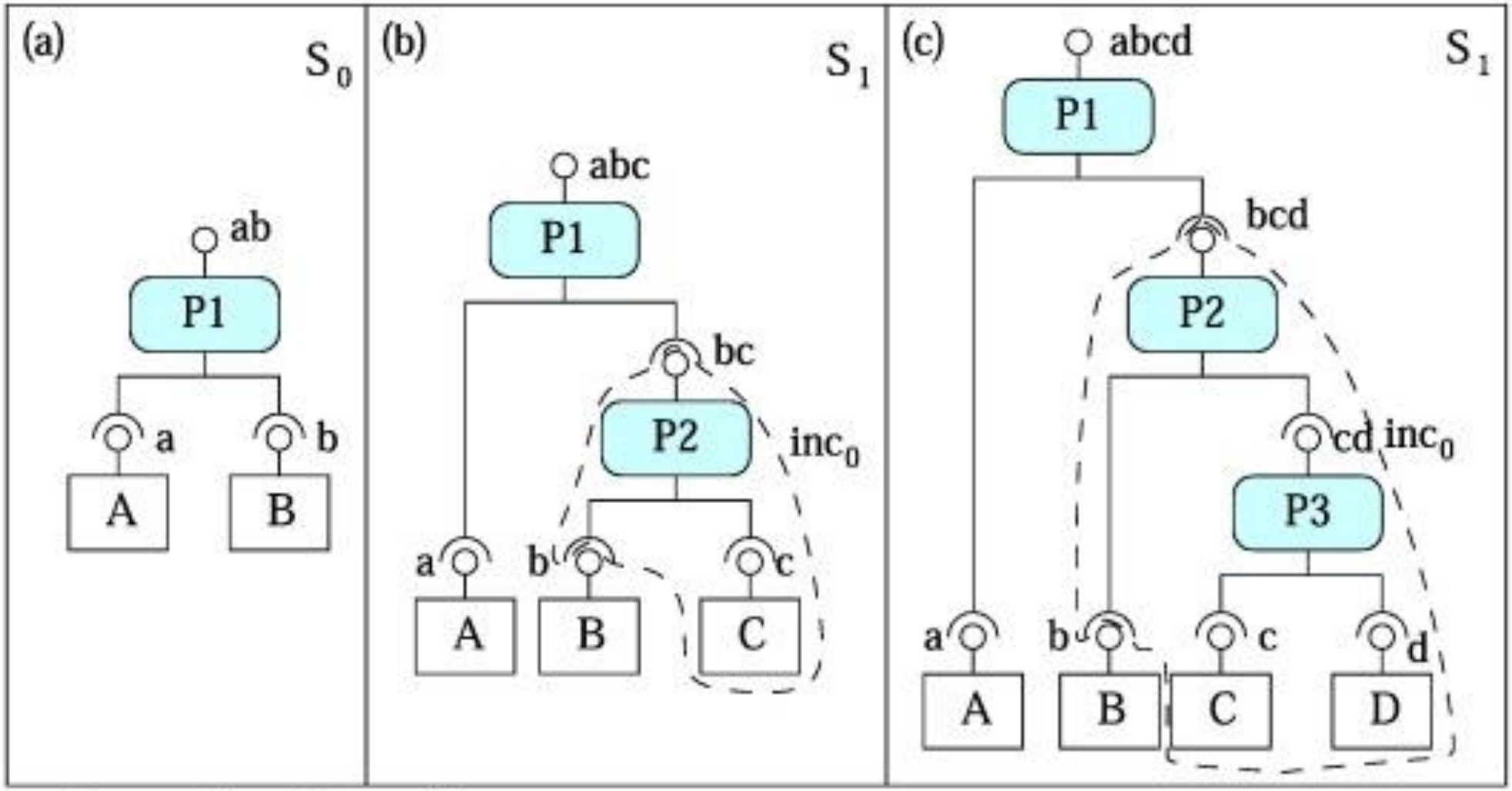
1. Predict System Behavior: Determine how the system behaves as it evolves over time.

2. Check Safety Properties: Ensure the system does not enter unsafe states.

3. Analyze System Performance: Assess if the system meets timing and performance criteria.


$B_N=\{s_1, s_2, ..., s_n\}$
Req
Res
<sN,iList>
<sN,oList>
Provided interface
PN
Encapsulation
Data
Invocation connector
IU
D
Computation unit
U
Component name
N
(a)
Composition Connectors
SEQ
SEL
PIPE
Sequencer
Selector
Pipe
Finite Loop
Infinite Loop
Guard
FCL Constraint
Adaptors
(b)


level2
level1
level0
P5
P1
P3
P2
P4
Control Layer
Computation Layer
A
B
C
D
E
F
G
H
I
J
K
L
Finite loop



$Bs_0 \subseteq Bs_1$, $T_{P1} \neq T_{P2}$, where T_{Pi} represents the type of a connector Pi.

Key Steps in Flow Pipe Construction

1. Define the Initial State Set:

- - Start with an initial condition or a set of initial states that describe where the system begins. This could be a single point or a bounded region in the state space.

2. Partition Time into Small Intervals:

- - Divide the time into small intervals, typically for each discrete state of the hybrid automata. This allows you to compute the system's behavior incrementally.

3. Compute Reachable Sets for Each Interval:

- For each time interval, use the system’s dynamics (typically represented by differential equations) to compute the set of states the system could reach.
- This set of states forms a "flow pipe segment" and represents where the system can be at the end of the interval, starting from any point within the initial set.

4. Handle Transitions Between States:

- When the system reaches a condition that triggers a discrete transition (like a guard in a hybrid automaton), the flow pipe construction must account for this transition.

- The system's behavior is then recomputed for the new state, potentially with different dynamics or constraints.

5. Repeat Until Termination:

- Continue computing reachable sets until a stopping criterion is met (e.g., a specific time horizon, reaching a target state, or verifying that the system stays within safe bounds).

Applications of Flow Pipe Construction in CPS

1. Autonomous Vehicles: In autonomous driving, flow pipe construction helps predict possible future positions of the vehicle under different inputs and disturbances. This allows for verifying that the vehicle won't collide with obstacles or exceed speed limits under all possible conditions.

2. Medical Devices: In implantable medical devices, like pacemakers, flow pipes can ensure that the device will function safely across all possible physiological conditions of the patient.

3. Aerospace Systems: For aircraft autopilot systems, flow pipe construction verifies that the aircraft can operate safely under changing atmospheric conditions and control commands, such as during takeoff, cruising, and landing phases.

4. Industrial Robotics: In robotic arms, flow pipe analysis ensures that movements stay within safe bounds, preventing collisions with other equipment or humans in the workspace.

Challenges in Flow Pipe Construction

- 1. Computational Complexity
- 2. Handling Nonlinear Dynamics
- 3. Guard and Invariant Conditions
- 4. Approximation Errors

Summary

- Flow pipe construction is a crucial technique in the analysis of cyber-physical systems.
- By computing the set of reachable states over time, it enables verification of safety and performance requirements in systems with complex, hybrid dynamics.
- Despite challenges in handling high-dimensional or nonlinear systems, flow pipe construction provides a structured approach for analyzing CPS behavior, helping ensure that such systems operate safely and predictably in real-world conditions.