Toward a Calculus for Optimizing CPS to Trustworthiness

KTH, Stockholm SWEDEN
1 December, 2017

Dr. E.R. Griffor
Assoc. Director for Cyber-Physical Systems
US National Institute of Standards and Technology
What are Cyber-Physical Systems (CPS)?
Cyber-Physical Systems (CPS) comprise interacting digital, analog, physical, and human components engineered for function through integrated logic and physics.

What is the relation to the ‘Internet of Things’ (IoT)?
Internet of Things (IoT) emphasizes digital infrastructure for widely connected, interacting, physical ‘things’.

CPS Program Goals
This CPS Program aims to promote discussion around CPS by developing, in collaboration with industry and academia, the technical foundations and tools needed to conceive, design, build, deliver and maintain these systems:
1. What is CPS?
2. How do we design, build and assure CPS throughout their lifecycle?
3. What discipline do we need to address the concerns that drive requirements, engineering and operation?
4. How do we test small and large scale CPS?

Assurance of CPS
Property-Tree of a CPS

Smart Cities: Global Cities Teams Challenge (GCTC)

CPS Framework
- Unified Mathematics of Logic and Physics
- Applications to Transportation – Autonomous Vehicles
- Smart Cities/Smart Grid
- Trustworthiness – Safety, Security, Privacy, Resilience and Reliability
- Modelling and Reasoning about CPS – UML/XML and formal methods

Internet of Things (IoT) and Smart Communities/Cities
From Security to Trustworthiness
Reach of cyberattacks is expanding

Adequate protection mechanisms have to include privacy, safety, security, and other areas (reliability, resilience) treated in an Integrated fashion

Server attacks
Creeper

1970s
Server, PC
Brain
Morris Worm
Michelangelo
Leandro
Tribe Flood DDOS
Melissa

1980-90s
Server, PC, Mobile
Code Red
Sasser
SQL Slammer
Cabir Premium

2000-2005
Server, PC, Mobile
Confliker
ICS-CERT
I Love You
CAESS

2005-2010
Server, PC, Mobile, ICS
Stuxnet
Duqu
GSM interface attack
Steel plant attack
Ransomware
Heart Bleed

2010-2017
Future
Further expansion

2017+

Adequate protection mechanisms have to include privacy, safety, security, and other areas (reliability, resilience) treated in an Integrated fashion.
Trustworthiness: integrated concept


Safety

Ability to ensure the absence of catastrophic consequences on the life, health, property, or data of stakeholders

Security

Internal or external protection from unintended and unauthorized access, change, damage, destruction, or use of systems

Privacy

Risks to individuals arising from the processing of their personal information

Reliability

Ability to deliver stable and predictable performance in expected conditions

Resilience

Ability of withstand instability, unexpected conditions, and gracefully return to predictable, but possibly degraded, performance.

Definition: Demonstrable likelihood that the system performs according to designed behavior under a typical set of conditions as evidenced by its characteristics, such as safety, security, privacy, reliability and resilience.
Integrated trustworthiness: sample categories of use cases

- **Modeling**: Use the integrated approach to model current and future systems and complex environments.
- **Design and development**: Anticipate connections, features, and constraints in designing CPS.
- **Analysis and assessment**: Assess existing systems to understand their potential vulnerabilities and to optimize deployment.
Integrated trustworthiness: some challenges

- **Misalignment of metrics**: Even for probabilistic models, the values for failure rates for, e.g., security and safety are very different.

- **Mutually exclusive requirements**: Requirements for adjacent areas are misaligned. E.g., transparency requirements are fundamentally different for security and privacy.

- **No mechanism for composition**: No reliable tools to model a component in context, e.g., autonomous vehicle in a Smart City.
Next steps: from TW positioning to ontology & reasoning

Current state
- CPS framework positions trustworthiness within CPS and allows the technologists to decompose this concept

Ontology and reasoning
- Future work needs to create a trustworthiness ontology matching the current CPS framework and reasoning algorithms to analyze trustworthiness

Trust language
- Trust language will need to be developed as a tool for trustworthiness calculus (building on existing ‘reasoning languages’ such as OCL, OWL, etc.)
Useful concept: trust evidence
Ability to rely on a broader list of characteristics (evidence) to assess trustworthiness. Some examples below.

**Definition:** trust evidence is an agreed upon system of parameters that could help define trustworthiness in a complex environment
Relevant area: human/technology connection

- Intent semantics: The intent of the user or developer while designing and using a system.
- Usability: Ease, with which a system can be used, learned, and integrated into an environment or process.
- Transparency: Ease, with which a system can be understood for functions that a user or other systems need to know.

Although the majority of interactions are machine to machine, the human aspect is very important!
CPS Framework Background
Frameworks – NIST Convening of Stakeholders

NIST Special Publication 1108r3
NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0

Smart Grid and Cyber-Physical Systems Program Office and Energy and Environment Division, Engineering Laboratory

Dec 2007 EISA SG Legislation


2010 Cloud Computing

Feb 2013 Executive Order

Feb 2010 Cybersecurity

June 2013 Climate Action Plan

June 2013 Community Disaster Resilience

2010 Big Data

June 2013 Smart America/Global Cities

2014 Cyber-Physical Systems

Priority Action Plans (PAPs)
# NIST CPS Public Working Group

## NIST CPS PWG leadership: David Wollman and Chris Greer

<table>
<thead>
<tr>
<th>Co-Chairs</th>
<th>Reference Arch</th>
<th>Use Cases</th>
<th>Security</th>
<th>Timing</th>
<th>Data Interop</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST</td>
<td>Abdella Battou, Ed Griffor</td>
<td>Eric Simmon</td>
<td>Vicky Pillitteri, Steve Quinn</td>
<td>Marc Weiss</td>
<td>Marty Burns</td>
</tr>
<tr>
<td>Academia</td>
<td>Janos Sztipanovits</td>
<td>John Baras</td>
<td>Bill Sanders</td>
<td>Hugh Melvin</td>
<td>Larry Lannom</td>
</tr>
<tr>
<td>Industry</td>
<td>Stephen Mellor, Shi-Wan Lin</td>
<td>Stephen Mellor</td>
<td>Claire Vishik</td>
<td>Sundeep Chandhoke</td>
<td>Peggy Irelan, Eve Schooler</td>
</tr>
</tbody>
</table>

## NIST SP 1500-201 and 1500-202

**Framework for Cyber-Physical Systems**

**Release 1.0**

May 2016

Cyber Physical Systems Public Working Group

[pages.nist.gov/cpspwg](pages.nist.gov/cpspwg)
What is a cyber-physical system (CPS)?

Cyber-Physical Systems comprise interacting digital, analog, physical, and human components engineered for function through integrated logic and physics.

Internet of Things (IoT) emphasizes digital infrastructure for widely connected, interacting physical ‘things’.

NIST Smart Grid and Cyber-Physical Systems Program Office
CPS Framework Development

CPS Stakeholders (Societal, Business & Technical) → Raw CPS concerns → Aspects and Concerns → CPS Framework

- Conceptualization Facet: Model of CPS
- Realization Facet: Instance of CPS
- Assurance Facet: CPS Assurance

CPS Framework Development

What things should be and what things are supposed to do

How things should be made and operate

How to prove things actually work the way they should

CPS Stakeholders (Societal, Business & Technical) → Raw CPS concerns → Aspects and Concerns → CPS Framework

- Conceptualization Facet: Model of CPS
- Realization Facet: Instance of CPS
- Assurance Facet: CPS Assurance

CPS Framework Development

What things should be and what things are supposed to do

How things should be made and operate

How to prove things actually work the way they should
NIST CPS PWG – CPS Framework

‘Concern-driven’: holistic, integrated approach to CPS/IoT concerns.

CPS Framework Structure

- Domains
  - Manufacturing
  - Transportation
  - Energy
  - Healthcare
  - Others...

- Aspects
  - Functional
  - Business
  - Human
  - Trustworthiness
  - Timing
  - Data
  - Boundaries
  - Composition
  - Lifecycle

Facets

<table>
<thead>
<tr>
<th>Conceptualization</th>
<th>Realization</th>
<th>Assurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case, Requirements, …</td>
<td>Design / Produce / Test / Operate</td>
<td>Argumentation, Claims, Evidence</td>
</tr>
</tbody>
</table>

Activities and their Artifacts

- Model of a CPS
- CPS
- CPS Assurance

- CPS Framework Release 1.0 (May2016) available at https://pages.nist.gov/cpspwg/
Purpose of the CPS Framework

• **Concern-driven structuring of development artifacts:** to facilitate assurance cases (by representing or analyzing a system along these dimensions, points of commonality or interoperability with other systems are revealed)

• **A normal-form for CPS/IoT system** (common way of presenting CPS/IoT that enables comparison of what is done, across the system, for the sake of any individual concern)

• Provides a **method for integrating CPS/IoT across domains** – the future of CPS/IoT is cross-domain integration. While some domains may have robust, integrated approaches to some concerns, there are typically radically different standards across domains.

CPS Framework is NOT A PROCESS!!
It is a method for integrating concerns into systems engineering processes!
CPS Framework
Workshop Goals

This workshop aims to promote discussion around CPS trustworthiness as it relates to how we conceive, design, build, deliver and maintain these systems.

1. What is CPS?
2. How do we design, build and assure CPS throughout their lifecycle?
3. What discipline do we need to address the concerns that drive requirements and engineering?
4. What needs to be the common core tooling?

Dashboard for Continuous Integration of CPS Development
How do we design, build and test CPS?

• Develop requirements.
• Specify the system, sub-systems and components.
• Build components.
• Unit test components.
• Assemble and test sub-systems.
• Assemble and test/validate full system.
What discipline do we need to address the concerns?

Concern Structure:
• Develop a full set of concerns.
• Develop the relationships between the concerns.

Systems Engineering Activities:
• Determine requirements needed to address each concern.
• Design, build and test to each set. (composition of concerns).
• Build the Assurance Case

CPS Framework Structure

Facets

- Conceptualization
- Realization
- Assurance

Activities and their Artifacts
- Model of a CPS
- CPS
- CPS Assurance
Analyzing and Developing CPS: Decomposition

Functional Decomposition (Logical and Physical)

CPS/Function Types
- Business Case
- Use Case ‘feature’
- CPS
- Logical
- Messages
- Info
- Physical
- Influences
- Energy

Safety “Properties” of a Function: Automatic Emergency Braking (AEB)
- Vehicle provides automated collision safety function
- Vehicle provides/maintains safe stopping
- Braking function reacts as required
- Stopping algorithm provides safe stopping
- Messaging function receives distance to obstacles and speed from propulsion function
- Distance and speed info is understood by braking function
- Friction function provides appropriate friction, depending on the road, tire pressure, etc.

*transductions

Apply Aspects/Concerns

Generate System Properties
What needs to be the common core tooling?

CPS Framework Open Source provides:

1) ‘Type Structure’ for:
   • Aspects and concern; and
   • Facets, engineering activities and outcomes

2) That type and sort compositionally:
   • properties/requirements and
   • artifacts

3) Encoded in a portable, reusable XML format.
Trustworthiness Aspect

CPS Framework Open Source Tools

System Artifacts and Data

High-level Aspects/Concerns

NIST CPS Framework XML Model

Concerns & Sys Properties

CPS Framework

Tools have both scope and depth!

Operational Leads

CPS N

CPS 2

CPS 1

CEO

CTO

CIO

Sr. Mgr.

DIR

Eng

Eng

Eng

NIST

CPS 1

CPS 2

CPS N

High-level Aspects/Concerns

Concerns & Sys Properties

NIST CPS Framework XML Model

Tools have both scope and depth!

Operational Leads

CPS Framework

System Artifacts and Data

High-level Aspects/Concerns

NIST CPS Framework XML Model

Concerns & Sys Properties

CPS Framework

Tools have both scope and depth!

Operational Leads

CPS Framework

System Artifacts and Data

High-level Aspects/Concerns

NIST CPS Framework XML Model

Concerns & Sys Properties

CPS Framework

Tools have both scope and depth!

Operational Leads

CPS Framework

System Artifacts and Data

High-level Aspects/Concerns

NIST CPS Framework XML Model

Concerns & Sys Properties

CPS Framework

Tools have both scope and depth!
**Expanded Concern Risk and Risk Mitigation Surface**

<table>
<thead>
<tr>
<th>IT System</th>
<th>CPS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Primary Impact of Failure</th>
<th>Digital</th>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>◅ ◅ ◅ ◅</td>
<td>◅ ◅ ◅ ◅</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mitigation Mechanisms</th>
<th>Digital</th>
<th>Analog</th>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>◅ ◅ ◅ ◅</td>
<td>◅ ◅ ◅ ◅</td>
<td>◅ ◅ ◅ ◅</td>
<td></td>
</tr>
</tbody>
</table>

“E.g. Better cybersecurity through physics!”
Autonomous Vehicle Braking - Example

1. A small LKAS camera beside the rear-view mirror monitors the road markings on either side and feeds data to a computer.

2. LKAS applies correct steering torque to keep car in the centre of the lane.

3. ACC radar sensors behind the Honda badge monitor the distance from the car in front.

4. If the gap with the car in front decreases, the car automatically brakes, then accelerates again to maintain a safe distance.

5. The ACC radar is also used for CMBS and recognises if a collision is imminent. CMBS warns the driver to take action first by an audio and visual warning on the dash. If no action is taken seat belt pre-tensioners will lightly tug the driver to give a physical warning and if still no action is taken, the system will apply strong braking to reduce the impact of a collision.
Example: Automotive System Functional Level
Braking System

Derived from original figure by Ricardo
www.ricardo.com
CPS Framework Ontology

Use-Cases, Formalization and Reasoning Examples
Progress

• Studied 3 scenarios
  o Automotive/Cam - Lane Keeping/Assist
  o Automotive/Sensors - Adaptive Cruise Control
  o Check colors and line-styles

• Single level of decomposition considered

• Generalized CPS diagram
  o Scenarios are instances of it

• Formalized interactions between concerns, properties

• Reasoning
  o Satisfaction of properties, concerns, aspects
  o Exogenous events, interactions with surrounding environment
  o Determination of mitigations
Goals

Demonstrate

- Formalization of CPS Model and surrounding domain
- Formalization of attacks, their effects, mitigation actions
- Reasoning about aspects, concerns, properties
  - Recursive traversal of concern trees
- Reasoning about physical ramifications
- Finding mitigations, reasoning about their effects
Approach

Combination of:
- Ontologies and ontology-based reasoning
- Boolean satisfiability
- Action languages
- Non-monotonic logic
- Constraint satisfaction
- Soft constraints
Formalization
Goals of the Formalization

• Capture relevant knowledge about a CPS
  o CPS-independent: Aspects/Concerns hierarchy
  o CPS-specific: Properties
  o Constraints, dependencies, tradeoffs
    ▪ General-purpose, e.g. thermodynamics
    ▪ CPS-specific, e.g.:
      • Component dependencies
      • Property dependencies
      • Effects of malware

• Enable reasoning about:
  o The current state of the CPS
    ▪ Both logical and physical
  o Which Properties are violated
  o Which Aspects/Concerns are currently satisfied
  o How to reach a desired state
    ▪ Recovery to mission capability
Proposed Formalization Components

1. Aspect/Concern hierarchy
2. Properties
   - Each concern is associated with a set of properties
     - \(\text{encrypted_mem(sam)}^{\text{confid}}\): property of concern
     - \(\text{encrypted_mem(sam)} \in P(\text{confidentiality})\)
   - Each property is represented by an atomic statement
     - Case 1: truth value is not specified
       - “Camera is capable of recording at 25 fps”
     - Case 2: truth value is specified \(\Rightarrow\) must be satisfied for concern to be satisfied
       - “Camera uses encrypted memory”
   - Will later include continuous features
3. Configuration of the CPS
   - Provides us with the **truth value of atomic statements**
     - “Basic camera is in use”
     - \( \text{in\_use(basic\_camera)}^{\text{cfg}} \)
     - “Basic camera is currently using encrypted memory and recording at 50 fps”

4. Constraints, dependencies, trade-offs
   - Provide us with **dependencies among atomic statements**
     - (properties + configuration)
     - “It is impossible for basic camera to use encrypted memory while recording at 50 fps”
     - “If basic camera uses encrypted memory while recording at 30+ fps, average frame drop rate is 25%” or “… is 25% \( \times \alpha \times \text{recoding\_rate} \)”
   - Describe how actions affect the configuration of the CPS
   - Recursively affect satisfaction of concerns, aspects
     - \( \text{encrypted\_mem(camera)}^{\text{confid}} \land \cdots \text{IMPACTS}_{\text{neg}} \text{drop\_frames}^{\text{funcity}} \)
     - \( \text{drop\_frames}^{\text{funcity}} \text{IMPACTS}_{\text{neg}} \text{functionality} \)

**General form**

(conjunction of atomic statements)

\( \text{IMPACTS}_{\text{pos}}^{\text{neg}} \text{atomic statement} \)
Generalized Integrated Monitoring & Control System
Can be specialized into any of the example systems discussed later

"Generalized" components circled in red
# Aspects, Concerns, Properties for Generalized Integrated Monitoring & Control System

<table>
<thead>
<tr>
<th>Aspects</th>
<th>Concerns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing</td>
<td>Time-interval and latency control</td>
</tr>
<tr>
<td>Functional</td>
<td>Functionality</td>
</tr>
<tr>
<td>Trustworthiness Elements</td>
<td>Safety</td>
</tr>
<tr>
<td></td>
<td>Physical security</td>
</tr>
<tr>
<td></td>
<td>Cybersecurity</td>
</tr>
<tr>
<td></td>
<td>Confidentiality</td>
</tr>
<tr>
<td></td>
<td>Integrity</td>
</tr>
<tr>
<td></td>
<td>Availability</td>
</tr>
</tbody>
</table>

Note: unless otherwise specified, “camera” refers to both basic and advanced camera

- Input Device 1 maintains consistent frequency of readings
- Input Device 1 should be capable of operating in low-speed mode and in high-speed mode
- SAM uses secure boot
- Input Device 1 uses secure boot
- SAM uses encrypted memory
- Input Device 1 uses encrypted memory
Summary of Examples

• Simplified “Body Camera”
  o Camera
  o SAM
  o Minimal assumptions on knowledge about connections
  o Link to broader domain: admissibility of video in court

• Self-Driving Car (a)
  o Camera
  o SAM
  o Broader domain: consequences on car’s operations

• Self-Driving Car (b)
  o Sensors
  o SAM
  o Broader domain: consequences on car’s operations
Example 1

Body Camera

An integrated body security system for situational awareness and safety of security personnel
Example 1
“BodyCam” Diagram

**Operating Environment**

- **Camera**
  - Influence: Angle change
  - Energy: Mechanical

- **Mic**
  - Influence: Video signal
  - Energy: Electromagnetic waves

- **Indicor**
  - Influence: Audio signal
  - Energy: Sound waves

- **SAM**
  - Message: Byte seq.
  - Information: Videofeed contains vf1 encoding AES

- **Command Center**
  - Message: Command packet
  - Information: Type of warning, attributes

- **Officer**
  - Message: Byte seq.
  - Information: Audiofeed contains af1 encoding AES

- **Storage**
  - Influence: Audio-video signal
  - Energy: Sound, light waves

- **Integrated Body Security System (Stand-alone part)**

**Physical Connection**

- **Direct Logical Connection**

**Physical Element**

- **Logical Element**

**Physical & Logical System**

- **Description of Interaction**
- **Annotation**

**Text**

- **Physical Connection**
- **Direct Logical Connection**

**Physical Element**

- **Logical Element**

**Physical & Logical System**

**Description of Interaction**

**Annotation**

**Text**

**Physical Connection**

- **Direct Logical Connection**

**Physical Element**

- **Logical Element**

**Physical & Logical System**

**Description of Interaction**

**Annotation**

**Text**
Example 1
Simplified “BodyCam”

Actors
- Camera
  - Basic camera
  - Advanced camera
  - Two modes:
    - Record at 25 fps
    - Record at 50 fps
  - Security features (may be enabled)
    - Encrypted memory
    - Secure boot
- SAM (Situational Awareness Module)
  - SAM controls the recording rate
Aspects, Concerns, Properties

Camera should be capable of recording at 25 fps or at 50 fps

Camera records at constant frame-rate

SAM uses encrypted memory
Camera uses encrypted memory

SAM uses secure boot
Camera uses secure boot

Note: unless otherwise specified, “camera” refers to both basic and advanced camera.
Formalization

Aspects
- Timing
- Functional
  - Time-interval and latency control
  - Functionality
  - Safety
  - Security
  - Privacy
  - Resilience

Concerns
- Physical security
- Cybersecurity

Aspect/Concern Hierarchy

Concerns:
- Actuation
- Adaptability
- Aggregation_Disaggregation
- Analytics
- Authentication
- Authorization
- Availability
- Behavioral
- Class_Identification
- Communication
- Complexity
- Component_Inventory
**Industrial laboratory**

Functional Aspect

- Camera records at constant frame-rate

**Property & Concern-Tree Satisfaction Axioms**

\[-\text{holds}(\text{sat\_concern}(C),S) \iff \text{step}(S), \text{addressedBy}(C,P), \neg \text{holds}(\text{sat\_property}(P),S).\]

**Concern’s Property**

- holds(\text{sat\_property}(P),S) \iff \text{step}(S), \neg \text{holds}(P,S).

**Property’s Description**

- \text{The camera records at constant frame-rate}

**Annotations:**

<table>
<thead>
<tr>
<th>Annotations</th>
<th>Input1ConsistentReadingFreq</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs:comment</td>
<td>The camera records at constant frame-rate</td>
</tr>
</tbody>
</table>
Formalization

- SAM uses secure boot
- Camera uses secure boot

Trustworthiness Aspect

Security Concern

Integrity Concern

Concern’s Properties
Example 1

System Configuration
- SAM and camera use encrypted memory and secure boot
- Recording rate set at 25 fps by SAM
- Basic camera is used

Formalization

```plaintext
obs("encrypted_mem_input1",true).
obs("encrypted_mem_sam",true).
obs("sec_boot_input1",true).
obs("sec_boot_sam",true).
obs("slow_mode_input1",true).
obs("using_basic_input1",true).
```
Example 1

Constraints, Dependencies, Trade-offs

“When the basic camera is in use, using encrypted memory and recording at 50 fps impacts negatively the dropping of frames”

\[
\text{in}\_\text{use(basic\_camera)}^{\text{cfg}} \land \text{encrypted\_mem(camera)}^{\text{confid}} \land \neg \text{rec\_rate(25fps)}^{\text{tim\_intvl}}
\]

\[\text{IMPACTS}_{\text{neg}}\]

\[\text{constant\_rate}^{\text{func\_ty}}\]

**Formalization**

impacted(neg,"consist\_reading\_freq\_input1",S) :-
    step(S),
    holds("using\_basic\_input1",S),
    holds("encrypted\_mem\_input1",S),
    -holds("slow\_mode\_input1",S).
Example 1

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set at 25 fps by SAM

Questions
- Is the functionality concern satisfied?
Example 1

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set at 25 fps by SAM

Questions
- Is the functionality concern satisfied?
  - Outcome of the reasoning process:
    - `holds(sat_concern("cpsf:Functionality"), 0)`
  - Functionality concern is satisfied
Example 1

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set at 25 fps by SAM

Questions
- Is the functional aspect satisfied?
- Outcome of the reasoning process:

```prolog
holds(sat_concern("cpsf:Functional"), 0)
```

Functional concern tree

Functional aspect is satisfied
Example 1

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set at 25 fps by SAM

Questions
- Are all aspects satisfied?
- Outcome of the reasoning process:

\[
\text{holds(sat\_all,0)}
\]

All aspects are satisfied
Example 1

Also tested:

- Asking questions about arbitrary aspects and their concerns
Example 1 – Step 2

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set at 25 fps by SAM

A cyberattack occurs and hacks the SAM to activate recording at 50 fps

Approximate First-Order Logic equivalent:

\[
\text{Occurs} (\text{Cyberattack}) \land \\
(Occurs (\text{Cyberattack}) \Rightarrow \cdots)
\]
Example 1 – Step 2

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set at 25 fps by SAM

A cyberattack occurs and hacks the SAM to activate recording at 50 fps

Questions
- Is the functionality concern satisfied?

Functionality concern is **not** satisfied
Example 1 – Step 2

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set at 25 fps by SAM

A cyberattack occurs and hacks the SAM to activate recording at 50 fps

Questions
- Is the functional aspect satisfied?

No
Example 1 – Step 2

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set at 25 fps by SAM

A cyberattack occurs and hacks the SAM to activate recording at 50 fps

Questions
- What happens in the environment?

The recording is thrown out of court

Conclusion based on the knowledge:
\[
\text{event(recipient_rejects, S) :- holds(sat_concern("cpsf:Functional"), S).}
\]
Example 1 – Step 3

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set at 25 fps by SAM

A cyberattack occurs and hacks the SAM to activate recording at 50 fps

Questions
- How can it be mitigated?

Mitigation Generation Module

\[
\text{do}(A,S) \lor \neg \text{do}(A,S) \iff \text{step}(S), \text{action}(A).
\]

\[
\neg \text{last_step}(S), \neg \text{holds}(\text{sat_all},S).
\]

Approximate First-Order Logic equivalent:

\[
\text{Do}(a) \lor \neg \text{Do}(a) \text{ for every action } a \quad \neg \text{Sat(All)} \supset \perp
\]
Example 1 – Step 3

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set at 25 fps by SAM

A cyberattack occurs and hacks the SAM to activate recording at 50 fps

Questions
- How can it be mitigated?

```
Run query

<table>
<thead>
<tr>
<th>Query Results (1 answers):</th>
</tr>
</thead>
<tbody>
<tr>
<td>unsatisfied concern/aspect/property</td>
</tr>
<tr>
<td>-------------------------------------</td>
</tr>
</tbody>
</table>
| disable("Basic camera is in use")  | action | 1
```

“Replace basic camera with advanced camera”
Example 2
Self-Driving Car (a)

A self-driving car uses SAM and camera for lane keeping/assist
Example 2
Self-Driving Car (a)

Operating environment

Camera
Targeting mechanism
Robotic camera
Mic
Indicator
Remote assist
Driver
Storage
Integrated lane keeping/assist system (stand-alone part)
Integrated lane keeping/assist system
Steering wheel

Physical connection
Direct logical connection
text
Description of interaction
text
Annotation

Physical element
Logical element
Physical & logical system

Message: byte seq.
Information: combined videofeed, audiofeed contains vf2 contains af2 encoding AES

af2
Influence: audio signal
Energy: sound waves

af1
Influence: audio signal
Energy: sound waves

vf1
Influence: video signal
Energy: electrical waves

vf2
Message: byte seq.
Information: combined videofeed, audiofeed contains vf1 encoding AES

camera
mechanical

Remote assist

Message: command packet
Information: type of warning, attributes

Command packet

Message: command packet
Information: degrees, speed

Integrated lane keeping/assist system (stand-alone part)

Storage

Message: byte seq.
Information: combined videofeed, audiofeed contains vf2 contains af2 encoding AES

O1
Influence: angle change
Energy: mechanical

Tomography

Message: byte seq.
Information: videofeed contains vf1 encoding AES

Command packet

Message: alarm packet
Information: type of alarm, timestamp, location, driver ID

Alarm packet

Message: warning signal
Energy: waves TBD

Warning signal

Message: angle change
Information: mechanical

Angle change

Physical & logical system

Description of interaction

Annotation

Physical element

Logical element

Physical connection

Direct logical connection

Text

Text

Example 2
Self-Driving Car (a)
Example 2: Self-Driving Car (a)

A self-driving car uses SAM and camera for lane keeping/assist

Actors

- Camera
  - Basic camera
  - Advanced camera
  - Two modes:
    • Record at 25 fps
    • Record at 50 fps
  - Security features (may be enabled)
    • Encrypted memory
    • Secure boot

- SAM (Situational Awareness Module)
  - SAM controls the recording rate
Example 2: Self-Driving Car (a)

System Configuration
- SAM and camera use encrypted memory and secure boot
- Basic camera is used
- Recording rate set to 25 fps by SAM

A cyberattack occurs and hacks the SAM to activate recording at 50 fps

Questions
- Is the functionality concern satisfied?
- Is the functional aspect satisfied?
- What are the physical ramifications?
- How can it be mitigated?

“Replace the basic camera with the advanced camera”
Example 3
Self-Driving Car (b)

A self-driving car uses SAM and sensors for adaptive cruise control
Example 2
Self-Driving Car (b)
Example 3: Self-Driving Car (b)

A self-driving car uses SAM and sensors for adaptive cruise control

Actors

- Sensor array
  - Interpolating sensor array (inaccurate hardware, interpolation compensates)
  - Non-interpolating sensor array (accurate hardware)

- SAM (Situational Awareness Module)
  - SAM requests sensor readings either at low rate or high rate
  - In interpolating sensor array:
    - High request rate causes interpolation to be cut short
    - Potentially leads to inaccurate data returned to SAM
Example 3: Self-Driving Car (b)

System Configuration
- SAM and sensors use encrypted memory and secure boot
- Interpolating sensor array is used
- SAM requests sensor readings at low rate

Questions
- Is the functionality concern satisfied?
- Is the functional aspect satisfied?
- Are all aspects satisfied?
Example 3 – Step 2

System Configuration
- SAM and sensor array use encrypted memory and secure boot
- Interpolating sensor array is used
- SAM requests sensor readings at low rate

A cyberattack occurs and hacks the SAM to request readings at a high rate

Questions
- Is the functionality concern satisfied?
- Is the functional aspect satisfied?
- What are the physical ramifications?

<table>
<thead>
<tr>
<th>cpsf:Functional</th>
<th>aspect</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpsf:Functionality</td>
<td>concern</td>
<td>1</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>accelerator control shuts off</td>
<td>event</td>
<td>1</td>
</tr>
<tr>
<td>brake control shuts off</td>
<td>event</td>
<td>1</td>
</tr>
</tbody>
</table>
Example 3 – Step 2

System Configuration
- SAM and sensor array use encrypted memory and secure boot
- Interpolating sensor array is used
- SAM requests sensor readings at low rate

A cyberattack occurs and hacks the SAM to request readings at a high rate

Questions
- Is the functionality concern satisfied?
- Is the functional aspect satisfied?
- What are the physical ramifications?
- How can it be mitigated?

"Replace interpolating sensor array with non-interpolating"
Example 3 (Variant)
Multiple Possible Mitigations
Example 3 – Step 3 (Extended)

System Configuration
- SAM and sensor array use encrypted memory and secure boot
- Interpolating sensor array is used
- SAM requests sensor readings at low rate

A cyberattack occurs and hacks the SAM.

Unless a patch is applied, the SAM will request readings at a high rate.

Questions
- How can it be mitigated?

Solution #1
“Patch the SAM”

Solution #2

```
Effect of patching
- holds(affected(sam),S+1) :-
  step(S), step(S+1), do(patch(sam),S).
holds("slow_mode_input1",S+1) :-
  step(S), step(S+1), do(patch(sam),S).
```

```
Result
Query Results (1 answers):
unsatisfied concern/aspect/property | type | step
patch the SAM | action | 1

Query Results (1 answers):
unsatisfied concern/aspect/property | type | step
disable("The interpolating sensor array is in use") | action | 1
```
Conclusions

Demonstrated

- Formalization of CPS Model and surrounding domain, attacks, mitigations
- Reasoning about concerns, properties, physical ramifications
- Finding (multiple) mitigations, reasoning about their effects
- Three examples: “BodyCam”, Self-driving Car (lane assist/keep & adaptive cruise control)

Next Steps

- Formalization, leverage of knowledge about CPS structure
- Reason over multiple levels of decomposition