A Top Down Approach to Cyber-Physical *System* Security

Trae Span

Advisor: Dr. Jeremy Daily

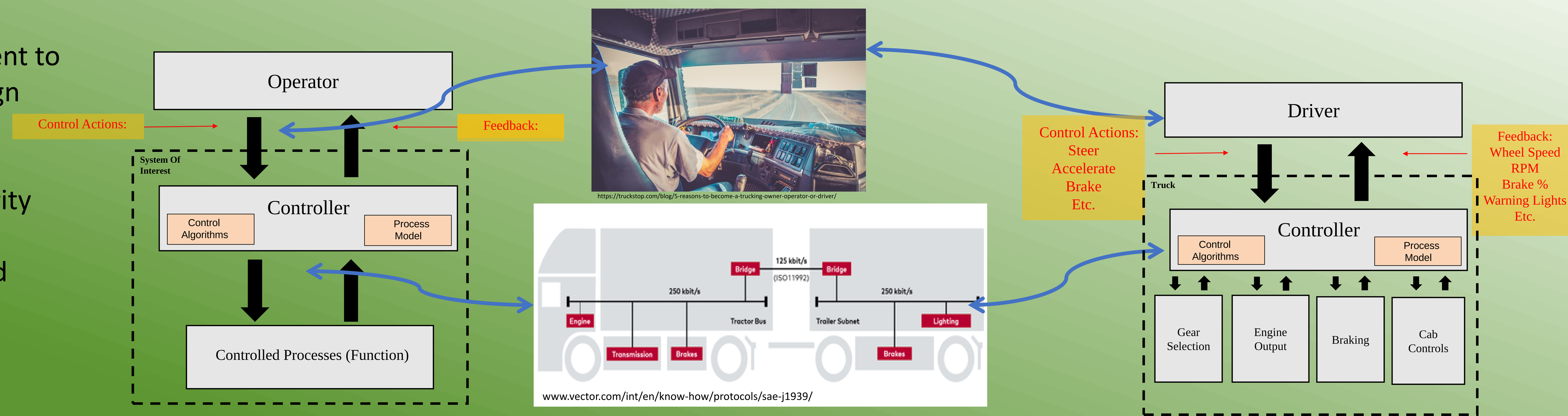
National Motor Freight Traffic Association, Inc.

Overview:

- Based System Theoretic Process Analysis
- Consider Complexity of Interactions
- Model System of Interest using control structure
- Treat Security as Functional Requirement to Allow trade offs from Conceptual Design

Difference from Traditional Methods:

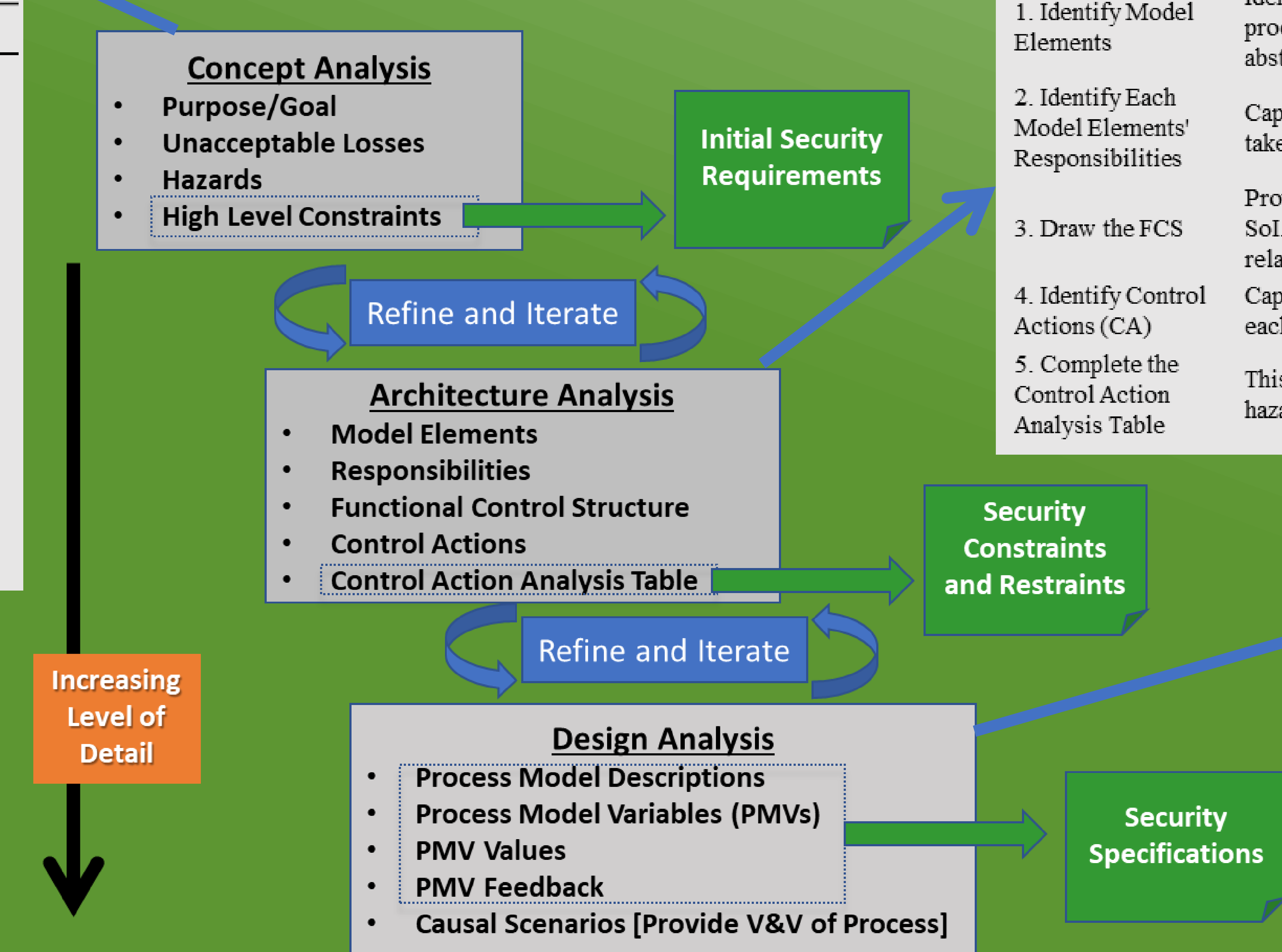
- Shift From 'Bolt On' to 'Baked In' Security
- Method is Threat Agnostic
  - Allows for security against undefined future threats
- Requirements generated address both Safety and Security
- Maps to existing SSE Processes



	Cyber-Physical System Security Phases		
	Concept Analysis	Architectural Analysis	Design Analysis
Purpose	Determine Initial Security Requirements	Determine "Design-To" Constraints and Restraints	Determine "Build-To" Criteria
NIST 800-160 SSE Processes	• BA - Business Analysis • SN - Stakeholder Needs • SA - Systems Analysis	• SR - System Requirements Definition • AR - Architectural Definition • SA - Systems Analysis	• DE - Design Definition • SA - Systems Analysis
SAE J3061	Feature Definition Initiation of Cybersecurity Lifecyle Initial Cybersecurity Concept Cybersecurity Goals	Threat Analysis and Risk Assessment Functional Cybersecurity Concept	Technical Cybersecurity Concept Verification/Validation of Cybersecurity Goals Refined Cybersecurity Assessment

CONCEPT ANALYSIS

Step	Description
1. Define the SoI's purpose and goal	Capture the mission statement and key activities of the system: 1) A system to: (What) 2) By Means of: (How) 3) In Order to: (Why)
2. Identify unacceptable losses	Define high level, intolerable system outcomes to key stakeholders (e.g., loss of life, injury, damage to equipment, reputation, mission, etc.).
3. Identify hazards	Identify system states that when coupled with worst case conditions lead to an unacceptable loss.
4. Develop system security constraints	Develop mission-informed security constraints that prevent the system from entering hazardous states. These constraints are synonymous with early safety, security, and resiliency functional requirements.



ARCHITECTURE ANALYSIS

Step	Description
1. Identify Model Elements	Identify actor(s), controller(s), and controlled process(es) for the SOI at the desired level of abstraction.
2. Identify Each Model Elements' Responsibilities	Capture the description and actions planned to be taken for the model elements identified.
3. Draw the FCS	Provide a visual functional-level depiction of the SoI. Depicts the model elements and control relationships between them.
4. Identify Control Actions (CA)	Captures (in verb form) the actions necessary for each element to execute their responsibilities.
5. Complete the Control Action Analysis Table	This table systematically enumerates which hazards are caused by each CA identified in step 4.

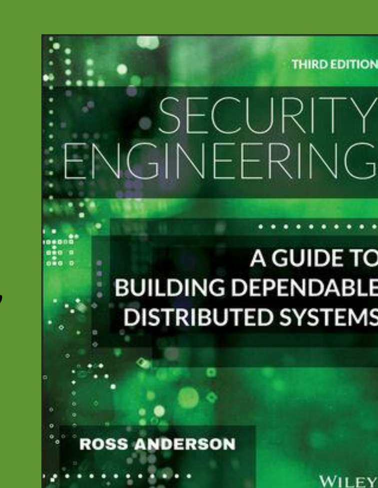
DESIGN ANALYSIS

Step	Description
1. Develop Process Model Descriptions	Describes the decision logic ("in plain English") for executing a given CA.
2. Identify Process Model Variables (PMV)	Process Model Variables are measurable indicators of the conditions that trigger a CA.
3. Identify PMV Values	PMV values are all the possible values a PMV can be assigned both acceptable and hazardous.
4. Identify PMV Feedback	Identifies which sensors provide PMV values to the actors and controller for decision making.
5. Carry out Causal Scenarios	Brainstorm how a specific implementation of the system may be compromised. Identifies critical CAs and validates the thoroughness of the model, CAs, and constraints.

Key Tenets:

- "Function Begets Form"
- Treat safety (and security) as emergent, system-level properties
- Leverage Systems Theory to provide alternative (more powerful) explanation for losses in complex, software intensive systems
- Losses involve a complex, dynamic "process"
  -- Not simply chains of failure events
  -- Arise in interactions among humans, machines and the environment

"Many systems fail because their designers protect the wrong things, or protect the right things in the wrong way"



SYSTEMS ENGINEERING

COLORADO STATE UNIVERSITY