Overview of the Afternoon

Session 1 (2:30 – 3:30) : STPA-Sec
Overview – STPA within Secure Systems Engineering (and Cyber Security)
• Introduction
• Observations on Cybersecurity today
• System Thinking and Security
• STPA-Sec overview
• Summary and Conclusion

Session 2 (3:30 – 5:00): STPA-Sec Practice
• Overview
• Concept Analysis
• Architectural Analysis
• Design Analysis
• User Q&A
• Summary and Conclusion

To Maximize the Available Time, I Will Assume Basic Familiarity With STAMP, STPA
an Will Leverage John Thomas’s Example from this Morning
System-Theoretic Process Analysis for Security (STPA-SEC):
Secure Systems Engineering, Cyber Security and STPA

William Young Jr, PhD

2019 STAMP Conference
Boston, MA

March 25, 2019
Disclaimer:

The views expressed in this presentation are those of the presenters and do not reflect the official policy or position of the United States Air Force, Department of Defense, Air Combat Command, MIT Lincoln Laboratory, Syracuse University, or the U.S. Government.
Introduction
Introduction

• Losses are growing and current approaches to securing complex, software intense, designed physical systems do not appear to be working as well as desired

• Origins of losses fall into at least one of two categories:
  • Disruption prevents engineered system from fulfilling its designed purpose
  • Disruption does not necessarily prevent the engineered system from fulfilling its primary purpose, but it produces an unacceptable “by-product”

• The side with individuals best able to conceptualize the most creative ways to exploit device/designed system functionality has competitive advantage (tactics)

Today, Security is Viewed Almost Universally as a Threat Problem
Introduction

Flawed logic
Conflicting goals
Poor Assumptions
Wrong Problem
Missing requirements
Incomplete requirements

Current Approaches Do Not Address Safety & Security Errors that lead to Losses When it is Most Effective and Cheapest to Do So

Design = Secure System Engineering
Construction = Secure System Development
O & M = Protect Data and IT Components

Ref: System Engineering
For Intelligent Transportation Systems

William.Young.3@US.AF.Mil  WYOUNG@MIT.EDU  © Copyright William Young, Jr, 2019
Observations on Cybersecurity
Today
Threat Based Approach to Developing a “Secure” Architecture

Current Security Analysis Depends on Identifying the Right Threat (Tactics), But Does Not Help Address the Larger Mission Assurance Goal (Strategy)

Ref: (Anderson, 2010; Shostack, 2014; Swiderski & Snyder, 2004)

© Copyright William Young, Jr, 2019
Schneier’s Attack Tree Model is the Intellectual Foundation of Most Thinking on Cybersecurity

“Clearly, what we need is a way to model threats against computer systems. If we can understand all the different ways in which a system can be attacked, we can likely design countermeasures to thwart those attacks...Security is not a product - it's a process. Attack trees form the basis of understanding that process.”

Schneier Based His Security Attack Trees on Fault Trees He Saw Used for Safety

Ref; *Dr. Dobb's Journal*, December 1999

William.Young.3@US.AF.Mil  WYOUNG@MIT.EDU  © Copyright William Young, Jr, 2019
Cybersecurity Through Today’s Analytic Lenses

The System Vulnerabilities are Driven by Threat Capability
Current Security Analysis

“When you ask an engineer to make your boat go faster, you get the trade-space. You can get a bigger engine but give up some space in the bunk next to the engine room. You can change the hull shape, but that will affect your draw. You can give up some weight, but that will affect your stability. When you ask an engineer to make your system more secure, they pull out a pad and pencil and start making lists of bolt-on technology, then they tell you how much it is going to cost.”

- Prof Barry Horowitz, UVA
What We Need to Get to

“The first thing we need in this process is the ability to state computer security requirements clearly and precisely... so that a competent professional can study it for a reasonably short amount of time and, say, "Oh, yes, I agree. If you build that particular system to that particular requirement, it's secure enough for that particular purpose.”

- Donald Good "The Foundations of Computer Security, We Need Some"
SYSTEM THINKING & SECURITY
Relooking Schneier’s Words

“Clearly, what we need is a way to model threats against computer systems. If we can understand all the different ways in which a system can be attacked, we can likely design countermeasures to thwart those attacks...Security is not a product - it's a process. STPA-Sec will form the basis of understanding that process.”
Cyber Security Through Different Analytic Lenses

Vulnerability Analysis

Business/Mission System Vulnerability

Mission or Business Operations

Impact Analysis

Threat Analysis

In Systems Engineering, Threats are Just One of Many Trades

William.Young.3@US.AF.Mil  WYOUNG@MIT.EDU  © Copyright William Young, Jr, 2019
New Approach: Secure Form Simply Realizes Secure Function

• “Form follows function” is a central tenant of system engineering and architecture

• Generate secure Business & Mission Systems by first defining the secure functionality to be realized

• Get to security via
  • Identify functionality required to solve the problem at hand (But we must understand problem)
  • Implement all required functionality securely based on understanding problem and context

• Architecture Defined (Crawley)
  • The embodiment of concept, and the allocation of physical/informational function to elements of form, and definition of interfaces among the elements and with the surrounding context
New Approach: Secure Form Simply Realizes Secure Function

• “Form follows function” is a central tenant of system engineering and architecture

• Generate secure Business & Mission Systems by first defining the secure functionality to be realized

• Get to security via
  • Identify functionality required to solve the problem at hand (But we must understand problem)
  • Implement all required functionality securely based on understanding problem and context

• Architecture Defined (Crawley)
  • The embodiment of concept, and the allocation of physical/informational function to elements of form, and definition of interfaces among the elements and with the surrounding context

We Can Use STAMP Model to Help Craft the Security Concept
STAMP Model & Security

• Focuses on function, not threat to guide realization (form)
  • Separates problem space from solution
  • Allows us to reason about function (and critique a proposed functional decomposition based on security related concerns)
• Provides a means to define and specify secure function clearly, unambiguously, and in context of the mission
• Functional Control Structure is simply a means to help envision how the necessary functionality can be implemented in a way that prevents losses identified
“Security” Losses Can Be Reframed as (Functionality) Control Problems

**Cause a Mid Air Collision**
- Process Model
- Control Algorithm

Aircraft must maintain minimum safe separation

**ENFORCE: Safe Separation**

**Cause Friendly Fire Loss**
- Process Model
- Control Algorithm

Only hostile forces must be engaged

**ENFORCE: Engagement Rules**

**Steal Customer PII**
- Process Model
- Control Algorithm

PII must only be exposed to authorized entities

**ENFORCE: Data Access Policy**

WYOUNG@MIT.EDU © Copyright William Young, Jr, 2019
From Systems Analysis to Secure Systems Analysis

“\textit{A systematic examination of a problem of choice in which each step of the analysis is made explicit wherever possible.}”


Secure Systems Analysis

Systems Analysis

Security Engineering

Systems Engineering

STPA-Sec Allows the Systems Analysis Framework to be Applied to Security
STPA-Sec

• Analysis process to generate a security concept and framework

• Examines a functional process through a security lens to gain insights and craft artifacts to enable additional reasoning

• Threats are just another environmental hindrance to function
  • In fact, the threats themselves don’t really matter…it’s the functional disruption they can deliver
  • We can engineer our systems to handle the most important functional disruptions

• Analysis methodology supports learning and facilitates stakeholder debates and trades (can imagine “what might be”)

William.Young.3@US.AF.Mil WYOUNG@MIT.EDU © Copyright William Young, Jr, 2019
STPA-Sec Extends STPA

- Synthesize (frame) the security problem
- Define purpose of the analysis
- Model the Control Structure
- Identify unsafe/unsecure control actions
- Step 2: Identify loss scenarios
- Wargame
Summary and Conclusion

• Security engineering and underlying systems thinking offers an alternative to address the challenge and bring strategy to bear

• Growing realization that security engineering must begin before architecture development...but we need a Security Engineering Analysis methodology
  • All analysis is based on models, so we require a model of how losses occur
  • Default model today is “threats cause our security-related losses” (but we don’t generally get to control the threats)

• STPA-Sec applies the STAMP model to provide a methodology to place security within a systems engineering context
  • Define “secure” functionality
  • Guide the development of an architecture to realize the functionality
  • We DO get to control our systems engineering
Concluding Thoughts from Sun Tzu

The opportunity to secure ourselves against defeat lies in our own hands.

The supreme art of war is to subdue the enemy without fighting.

Strategy without tactics is the slowest route to victory. Tactics without strategy is the noise before defeat.
My Contact Information

**WYOUNG@MIT.EDU** – Personal Email

**William.Young.3@US.AF.Mil** – Government Email

(for 6 more months)
Disclaimer:

The views expressed in this presentation are those of the presenters and do not reflect the official policy or position of the United States Air Force, Department of Defense, Air Combat Command, MIT Lincoln Laboratory, Syracuse University, or the U.S. Government.
Overview of the Practice Session

Session 2 (3:30 – 5:00): STPA-Sec Practice

• STPA-Sec for Security Engineering Analysis
• Concept Analysis
• Architectural Analysis
• Design Analysis
• User Q&A
• Summary and Conclusion

To Maximize the Available Time, I Will Assume Basic Familiarity With STAMP, STPA and Will Leverage John Thomas’s Example from this Morning
Rules of Engagement

• Extends aspects of Dr John Thomas’s morning STPA tutorial
  • Won’t cover the things he discussed
  • Will Identify security-related differences and additions
  • Will offer my techniques in a few areas

• Generally follows STPA Handbook guidelines

• Available time won’t allow for deep dive, but will have time over the next two days to discuss and answer detailed questions

• This is notional example and greatly simplified to fit within the time allotted

• Brevity prevents replication of the group learning that normally occurs

• Can’t simulate the iterative nature and the rich conversations that occur

• I want to save time at the end to address specific user questions encountered during real-world applications

We are Summarizing 40+ Hours of Instruction into 90 Minutes...We Will Only Hit Wavetops
STPA-Sec For Security Engineering Analysis

Satellite System Example Based on John Thomas Example Used in Earlier STPA Tutorial (Used With Dr Thomas’ Permission) and the Paper “A Top Down Approach for Eliciting Systems Security Requirements for a Notional Satellite System” by Mailoux, Span, Mills and Young
Problem Framework – Concept Analysis
• Goal / Purpose
• Unacceptable Losses
• Hazards
• High Level Constraints

Analysis / Synthesis (Refine & Iterate)

Functional Framework – Architectural Analysis
• Model Elements
• Responsibilities
• Functional Control Structure
• Control Actions
• Control Action Analysis Table (Step 1)

Analysis / Synthesis (Refine & Iterate)

Enterprise Architecture – Design Analysis
• Process Model Descriptions
• Process Model Variables (PMVs)
• PMV Values
• PMV Feedback
• Causal Scenarios (Adversary, Accident, Nature)
• War Gaming

Security-related material or techniques

Ends

Ways

Intent

Increasing Detail (Requirements)

Means

Initial Security Requirements

Security Constraints & Restraints

Security Specifications

Security

Constraints &

Restraints

Security

Specifications

William.Young.3@US.AF.Mil WYOUNG@MIT.EDU © Copyright William Young, Jr, 2019
Notional Spacecraft Through a Security Lens

From John Thomas’ Example this Morning

• Unmanned cargo transfer spacecraft
• Launched aboard rocket
• Rendezvous with International Space Station (ISS)
• Docks with ISS to deliver supplies
• Undocks and Returns to Earth

Additional Factors

• Proximity operations involve ISS (including crew), and ground stations
• Spacecraft employs proprietary software that company has invested significant IRAD to develop and patent
• System is commercially owned, operated, and maintained
• Company is liable for damage to supplies while enroute and for mission impact if supplies not delivered

Additions to morning STPA Tutorial Scenario

Adapted from Dr Thomas’ STPA Tutorial  
William.Young.3@US.AF.Mil  
WYOUNG@MIT.EDU  
© Copyright William Young, Jr, 2019
Problem Framework: Concept Analysis

Determining Initial Security Requirements
## Concept Analysis Overview

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| **1. Define the System of Interest (SOI), SOI purpose and SOI 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)  
4) While: (Bounds) |
| **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. |

\* Security-related addition, modification, or technique
Big Picture: Synthesize (Frame) Security Problem

• Sets the foundation for the security analysis
• Must ID all relevant stakeholders
• Must understand how product / service fits into organizational strategy
• Surface key assumptions (and dependencies)
• Satisfies key aspects of Business or Mission Analysis (BMA) in ISO/IEEE/IEC 15288
• Examine required functionality from a security perspective

“Many systems fail because their designers protect the wrong things, or protect the right things in the wrong way” – Ross Anderson in *Security Engineering*
Define System Purpose and Goal

“A system to do {What = Purpose} by means of {How = Method} in order to contribute to {Why = Goals} while {Constraints, Restraints}

Specify a gap between “as is” and “to be” that will be addressed through a process (e.g. a transformation of some type)

Military parallel is Operational Design (applied Operational Art) as captured in Joint Pub 5-0

Iterative Process is Challenging, but Generates Rich Conversations in Practice (e.g. USAF MLV)
Define System Purpose and Goal

From John Thomas’ Example this Morning

- Unmanned cargo transfer spacecraft
- Launched aboard rocket
- Rendezvous with International Space Station (ISS)
- Docks with ISS to deliver supplies
- Undocks and Returns to Earth

Goal / Purpose

Unacceptable Losses

Hazards

High Level Constraints

Additional Factors

- Proximity operations involve ISS (including crew), and ground stations
- Spacecraft employs proprietary software that company has invested significant IRAD to develop and patent
- System is commercially owned, operated, and maintained
- Company is liable for damage to supplies while enroute and for mission impact if supplies not delivered

Format

“A system to do {What = Purpose} by means of {How = Method} in order to contribute to {Why = Goals} while {constraints, restraints}”

What Might Be a Possible Solution from the Spacecraft Example?
Spacecraft Example

“A system to do \{\text{What} = \text{Purpose}\}
by means of \{\text{How} = \text{Method}\}
in order to contribute to \{\text{Why} = \text{Goals}\}
while \{\text{constraints, restraints}\}”
A system to autonomously resupply ISS by means of launching, navigating, docking, and undocking a space vehicle in order to support the ongoing ISS mission and research while maintaining profitable operations, minimizing risk to ISS/cargo, and improving the company’s position and branding as a responsible world leader in space technology.

This is one Solution, But There Others

Adapted from Dr Thomas’ STPA Tutorial

William.Young.3@US.AF.Mil WYOUNG@MIT.EDU © Copyright William Young, Jr, 2019
Adding Security-Related Unacceptable Losses

• “Unacceptable Losses” and “Accidents” are the same thing
• Many of the security losses will overlap with safety accidents
• Security perspective may add nuance to a previous safety perspective
• Security perspective may also highlight important safety / security trades
• Focus on alternative “system” uses
• Focus on security concerns of non-traditional stakeholders
• Outcomes and final conditions, not failures

Simply Clarifying Unacceptable Losses May Provide a Significant Boost in Security Effectiveness!
Spacecraft Losses

- Unacceptable Losses (From Earlier Today)
  - A-1: HTV collides with ISS
  - A-2: Loss of delivery mission

- Unacceptable Losses (Modified From Earlier Today)
  - L-1: Loss of Vehicle or ISS
  - L-2: Significant Damage to ISS or Vehicle
  - L-3: Loss of Resupply Payload

Are these Safety or Security-Related Losses?
Spacecraft Unacceptable Losses

**Unacceptable Losses**

- L-1: Loss of Vehicle or ISS
- L-2: Significant Damage to ISS or Vehicle
- L-3: Loss of Resupply Payload

Are there other unacceptable losses Related to Security? (Take a Few Minutes to Discuss)
Expanded (Security-related) Spacecraft Unacceptable Losses

**Unacceptable Losses**
- L-1: Loss of Vehicle or ISS
- L-2: Significant Damage to ISS or Vehicle
- L-3: Loss of Resupply Payload
- L-4: Loss of Reputation
- L-5: Loss of Intellectual Property

A system to **autonomously resupply ISS** by means of **launching, navigating, docking, and undocking a space vehicle** in order to **support the ongoing ISS mission and research** while maintaining profitable operations, minimizing risk to ISS/cargo, and improving the company’s position and branding as a responsible world leader in space technology.

Are there other unacceptable losses Related to Security? (Take a Few Minutes to Discuss)
Expanded Spacecraft Unacceptable Losses

- **Unacceptable Losses**
  - L-1: Loss of Vehicle or ISS
  - L-2: Significant Damage to ISS or Vehicle
  - L-3: Loss of Resupply Payload
  - L-4: Loss of Reputation
  - L-5: Loss of Intellectual Property

**Goal / Purpose**

- A system to **autonomously resupply ISS**
- by means of **launching, navigating, docking, and undocking a space vehicle**
- in order to **support the ongoing ISS mission and research while maintaining profitable operations, minimizing risk to ISS/astronauts/cargo, and improving the company’s position and branding as a responsible world leader in space technology**.

**Tip:** The “Why” and “While” provide insights to guide Unacceptable Losses

---

Adapted from Dr Thomas’ STPA Tutorial  William.Young.3@US.AF.Mil WYOUNG@MIT.EDU © Copyright William Young, Jr, 2019
Using “How” Verbs to Help Identify System Level Hazards

<table>
<thead>
<tr>
<th>Losses</th>
<th>Verbs</th>
<th>Launch</th>
<th>Navigate</th>
<th>Dock</th>
<th>Undock</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1: Loss of Vehicle or ISS</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>L2: Significant Damage to ISS or Vehicle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L3: Loss of Resupply Payload</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L4: Loss of Reputation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L5: Loss of Intellectual Property</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

High-level Functionality that is Required to Accomplish Goal

Unacceptable Losses that Must be Avoided

Must Control “1” sufficiently to accomplish mission while not causing “2”
(NOTE: This is true regardless of architecture!)
### Using “How” Verbs to Help Identify System Level Hazards

<table>
<thead>
<tr>
<th>Losses</th>
<th>Verbs</th>
<th>Launch</th>
<th>Navigate</th>
<th>Dock</th>
<th>Undock</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1: Loss of Vehicle or ISS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2: Significant Damage to ISS or Vehicle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L3: Loss of Resupply Payload</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L4: Loss of Reputation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L5: Loss of Intellectual Property</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We can use the *functional relationship* to gain insight into our Hazards ("A **condition** with the potential to cause injury, illness, or death of personnel; damage to or loss of equipment or property; or **mission degradation**." [DoD])

---

William.Young.3@US.AF.Mil  WYOUNG@MIT.EDU  © Copyright William Young, Jr, 2019
# Using “How” Verbs to Help Identify System Level Hazards

<table>
<thead>
<tr>
<th>Losses</th>
<th>Launch</th>
<th>Navigate</th>
<th>Dock</th>
<th>Undock</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1: Loss of Vehicle or ISS</td>
<td>Improper launch functionality may place vehicle in unrecoverable orbit</td>
<td>Navigation to wrong point or at wrong time can lead to loss of vehicle</td>
<td>Excessive closure during docking can cause damage to ISS or ship</td>
<td>Inadvertent undocking may compromise vehicle or ISS</td>
</tr>
<tr>
<td>L2: Significant Damage to ISS or Vehicle</td>
<td>Excessive launch forces may damage vehicle or cargo</td>
<td>Navigation through space radiation fields may damage vehicle</td>
<td>Excessive closure during docking can cause damage to ISS or ship</td>
<td>Inadvertent undocking may compromise vehicle or ISS</td>
</tr>
<tr>
<td>L3: Loss of Resupply Payload</td>
<td>Excessive forces during launch may damage payload</td>
<td>Excessive forces on payload during enroute portion</td>
<td>Docking attempted when ISS not ready or docking functionality applied when not docking</td>
<td>Undocking functionality applied before desired</td>
</tr>
<tr>
<td>L4: Loss of Reputation</td>
<td>Failed launch attempt or vehicle destruction</td>
<td>Losing vehicle enroute</td>
<td>Vehicle colliding with ISS when under control of company</td>
<td>Vehicle undocking with ISS when commanded</td>
</tr>
<tr>
<td>L5: Loss of Intellectual Property</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
</tr>
<tr>
<td>Losses</td>
<td>Verb(s)</td>
<td>Launch</td>
<td>Navigate</td>
<td>Dock</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>L1: Loss of Vehicle or ISS</td>
<td>Improper launch functionality may place vehicle in unrecoverable orbit</td>
<td>Navigation to wrong point or at wrong time can lead to loss of vehicle</td>
<td>Excessive closure during docking can cause damage to ISS or ship</td>
<td>Inadvertent undocking may compromise vehicle or ISS</td>
</tr>
<tr>
<td>L2: Significant Damage to ISS or Vehicle</td>
<td>Excessive launch forces may damage vehicle or cargo</td>
<td>Navigation through space radiation fields may damage vehicle</td>
<td>Excessive closure during docking can cause damage to ISS or ship</td>
<td>Inadvertent undocking may compromise vehicle or ISS</td>
</tr>
<tr>
<td>L3: Loss of Resupply Payload</td>
<td>Excessive forces during launch may damage payload</td>
<td>Navigation through space radiation fields may damage vehicle</td>
<td>Excessive closure during docking can cause damage to ISS or ship</td>
<td>Inadvertent undocking may compromise vehicle or ISS</td>
</tr>
<tr>
<td>L4: Loss of Reputation</td>
<td>Failed launch attempt or vehicle destruction</td>
<td>Navigation through space radiation fields may damage vehicle</td>
<td>Excessive closure during docking can cause damage to ISS or ship</td>
<td>Inadvertent undocking may compromise vehicle or ISS</td>
</tr>
<tr>
<td>L5: Loss of Intellectual Property</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
</tr>
</tbody>
</table>

Telemetry must be provided for remote operations. But it may also potentially disclose propriety data.
Using “How” Verbs to Help Identify System Level Hazards

<table>
<thead>
<tr>
<th>Losses</th>
<th>Launch</th>
<th>Navigate</th>
<th>Dock</th>
<th>Undock</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1: Loss of Vehicle or ISS</td>
<td>Improper launch functionality may place vehicle in unrecoverable orbit</td>
<td>Navigation to wrong point or at wrong time can lead to loss of vehicle</td>
<td>Excessive closure during docking can cause damage to ISS or ship</td>
<td>Inadvertent undocking may compromise vehicle or ISS</td>
</tr>
<tr>
<td>L2: Significant Damage to ISS or Vehicle</td>
<td>Excessive launch forces may damage vehicle or cargo</td>
<td>Navigation through space radiation fields may damage vehicle</td>
<td><strong>Excessive closure during docking can cause damage to ISS or ship</strong></td>
<td>Inadvertent undocking may compromise vehicle or ISS</td>
</tr>
<tr>
<td>L3: Loss of Resupply Payload</td>
<td>Excessive forces during launch may damage payload</td>
<td>Excessive forces on payload during enroute portion</td>
<td>Docking attempted when ISS not ready or docking functionality applied when not docking</td>
<td>Undocking functionality applied before desired</td>
</tr>
<tr>
<td>L4: Loss of Reputation</td>
<td>Failed launch attempt or vehicle destruction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L5: Loss of Intellectual Property</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
</tr>
</tbody>
</table>

**Docking Maneuver (e.g. thrust) must be constrained within limits while vehicle is in close proximity to ISS**
<table>
<thead>
<tr>
<th>Losses</th>
<th>Launch</th>
<th>Navigate</th>
<th>Dock</th>
<th>Undock</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1: Loss of Vehicle or ISS</td>
<td>Improper launch functionality may place vehicle in unrecoverable orbit</td>
<td>Navigation to wrong point or at wrong time</td>
<td>Excessive closure during docking can cause damage</td>
<td>Inadvertent undocking may compromise vehicle or ISS</td>
</tr>
<tr>
<td>L2: Significant Damage to ISS or Vehicle</td>
<td>Excessive launch forces may damage vehicle or cargo</td>
<td>Navigation through space radiation fields may damage vehicle</td>
<td>Excessive closure during docking can cause damage</td>
<td>Inadvertent undocking may compromise vehicle or ISS</td>
</tr>
<tr>
<td>L3: Loss of Resupply Payload</td>
<td>Excessive forces during launch may damage payload</td>
<td>Excessive forces on payload during enroute portion</td>
<td>Docking attempted when ISS not ready or docking functionality applied when not docking</td>
<td>Undocking functionality applied before desired</td>
</tr>
<tr>
<td>L4: Loss of Reputation</td>
<td>Failed launch or vehicle destruction</td>
<td></td>
<td></td>
<td>Vehicle undocking with ISS when commanded</td>
</tr>
<tr>
<td>L5: Loss of Intellectual Property</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
<td>Monitored telemetry may reveal proprietary data</td>
</tr>
</tbody>
</table>

We can also use the matrix to help ID previously missed functionality.
## Identifying a Missing Verb

<table>
<thead>
<tr>
<th>Losses</th>
<th>Launch</th>
<th>Navigate</th>
<th>Dock</th>
<th>Undock</th>
<th>Maintain (environment)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1: Loss of Vehicle or ISS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2: Significant Damage to ISS or Vehicle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>L3: Loss of Resupply Payload</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L4: Loss of Reputation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L5: Loss of Intellectual Property</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

L3 Highlights functionality that is required to achieve the goal and has an associated unacceptable loss, but no associated verb.
<table>
<thead>
<tr>
<th>Hazard</th>
<th>Description</th>
<th>Worst Case Environment</th>
<th>Associated Losses</th>
</tr>
</thead>
<tbody>
<tr>
<td>H2: Safe Closure Rate Between Space Vehicle and ISS exceeded</td>
<td>Commanded or uncommanded thrust provided in close proximity to ISS that takes vehicle out of safe closure parameters</td>
<td>ISS Crew or GSS crew does not detect deviation and/or is unable to take corrective actions to prevent a collision</td>
<td>L1, L2, L3</td>
</tr>
</tbody>
</table>

What system state or set of conditions together with a set of worst-case environmental conditions will lead to a loss? (Just like this Morning’s STPA Tutorial)
<table>
<thead>
<tr>
<th>Hazards</th>
<th>Losses</th>
<th>L1: Loss of Vehicle or ISS</th>
<th>L2: Significant Damage to ISS or Vehicle</th>
<th>L3: Loss of Resupply Payload</th>
<th>L4: Loss of Reputation</th>
<th>L5: Loss of Intellectual Property</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>H1: H1: Failure to Maintain Safe Separation between the Space Vehicle and the ISS</strong></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td><strong>H2: Exceed Safe Closure Rate Between Space Vehicle and ISS</strong></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>H3: Payload Environment not Maintained Within Operational Limits</strong></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>H4: Launch parameter limits exceeded</strong></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>H5: Proprietary data disclosed to unauthorized entity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Develop High-level System Security Constraints

<table>
<thead>
<tr>
<th>Hazard</th>
<th>System Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>H1</strong>: Failure to Maintain Safe Separation between the Space Vehicle and the ISS</td>
<td></td>
</tr>
<tr>
<td><strong>H2</strong>: Exceed Safe Closure Rate Between Space Vehicle and ISS</td>
<td>C2: System must maintain safe closure rate with ISS during docking and undocking</td>
</tr>
<tr>
<td><strong>H3</strong>: Payload Environment not Maintained Within Operational Limits</td>
<td></td>
</tr>
<tr>
<td><strong>H4</strong>: Launch parameter limits exceeded</td>
<td></td>
</tr>
<tr>
<td><strong>H5</strong>: Proprietary data disclosed to unauthorized entity</td>
<td></td>
</tr>
</tbody>
</table>

We Will Leverage ABORT functionality to Enforce this Constraint

Adapted from Dr Thomas’ STPA Tutorial  William.Young.3@US.AF.Mil  WYOUNG@MIT.EDU  © Copyright William Young, Jr, 2019
Functional Framework: Architectural Analysis

Developing Security Constraints and Restraints
Spacecraft Example—Architectural Analysis Overview

Need

Functional Equivalent

Adapted from Dr Thomas’ STPA Tutorial  William.Young.3@US.AF.Mil  WYOUNG@MIT.EDU  © Copyright William Young, Jr, 2019
## Architectural Analysis Overview

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Identify model elements</td>
<td>Identify actor(s), controller(s), and controlled process(es) for the SoI at the desired level of abstraction.</td>
</tr>
<tr>
<td>2. Identify each elements’ responsibilities</td>
<td>Capture the description and actions planned to be taken for the model elements identified.</td>
</tr>
<tr>
<td>3. Build Initial Functional Control Structure to Model control relationships</td>
<td>Organize the model elements to pictorial show the relationships between elements in a functional control structure.</td>
</tr>
<tr>
<td>4. Identify Control Actions (CA)</td>
<td>Captures (in verb form) the actions necessary for each element to execute their responsibilities.</td>
</tr>
<tr>
<td>5. Complete the CA analysis table</td>
<td>The CA analysis table systematically enumerates which hazards are caused by each CA identified in step 4.</td>
</tr>
</tbody>
</table>
Spacecraft–Model Elements

Problem Space (Function)

A system to **autonomously resupply ISS** by means of **launching, navigating, docking, and undocking a space vehicle and maintaining cargo** in order to **support the ongoing ISS mission and research** while **preserving payload, maintaining cost effective operations, minimizing risk to the astronauts, and improving the organization’s position and branding as a responsible community partner and world leader in technology**.

Developed in Initial Problem Framing

Solution Space (Form)

**ISS**

**GSS**

**Maneuver Control Subsystem**

**Onboard Controller**

Entities are Specified and Implied in Initial Documentation (But must Parse)

Adapted from Dr Thomas’ STPA Tutorial

William.Young.3@US.AF.Mil

WYOUNG@MIT.EDU

© Copyright William Young, Jr, 2019
A system to **autonomously resupply ISS**
by means of **launching, navigating, docking, and undocking a space vehicle and maintaining cargo**
in order to **support the ongoing ISS mission and research**
while **preserving payload, maintaining cost effective operations, minimizing risk to the astronauts, and improving the organization’s position and branding as a responsible community partner and world leader in technology**.

### Our Example Problem

Our Example Problem will focus on analyzing the statement: “System will be capable of ABORTING docking maneuver if unsafe conditions arise”

Adapted from Dr Thomas' STPA Tutorial
### Entity Activity Diagram

<table>
<thead>
<tr>
<th>Verbs</th>
<th>Launch</th>
<th>Navigate</th>
<th>Dock</th>
<th>Undock</th>
<th>Maintain (environment)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISS Segment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSS Segment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Onboard Vehicle Control System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maneuver Subsystem</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environmental control subsystem</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other Subsystems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Identify data (Parse) documents and place specified and implied responsibilities for the entities inside the various boxes**
### Spacecraft– Model Elements

<table>
<thead>
<tr>
<th>High-Level Functional Activity</th>
<th>Model Elements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dock</td>
<td>ISS</td>
<td>ISS be capable of commanding an ABORT if unsafe conditions arise during docking</td>
</tr>
<tr>
<td>Dock</td>
<td>GSS</td>
<td>GSS be capable of commanding an ABORT if unsafe conditions arise during docking</td>
</tr>
<tr>
<td>Dock</td>
<td>Onboard Control System</td>
<td></td>
</tr>
</tbody>
</table>

**Do we Expect the Spacecraft to be capable of internally (OCS) directed ABORT? (Implied Functionality ?)**
## Spacecraft—Model Elements

<table>
<thead>
<tr>
<th>High-Level Functional Activity</th>
<th>Model Elements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dock</td>
<td>ISS</td>
<td>GSS be capable of commanding an ABORT if unsafe conditions arise during docking</td>
</tr>
<tr>
<td>Dock</td>
<td>GSS</td>
<td>GSS be capable of commanding an ABORT if unsafe conditions arise during docking</td>
</tr>
<tr>
<td>Dock</td>
<td>Onboard Control System</td>
<td><strong>OCS</strong> receive (encrypted) ABORT when issued by ISS or GSS, decrypt (if required), terminate unsafe maneuver, command Attitude Control System to return vehicle to a safe distance from ISS and safe operational parameters. OCS will be capable of automatically sensing and commanding the Attitude Control System to ABORT docking maneuver if unsafe conditions arise during docking</td>
</tr>
</tbody>
</table>
## Spacecraft—Responsibilities

### Key Activity: Docking

<table>
<thead>
<tr>
<th>Element</th>
<th>Responsibility Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Segment</td>
<td>• Initiate process</td>
</tr>
<tr>
<td></td>
<td>• Send ABORT signal (encrypt?)</td>
</tr>
<tr>
<td></td>
<td>• Monitor progress</td>
</tr>
<tr>
<td>ISS Segment</td>
<td>• Monitor progress</td>
</tr>
<tr>
<td></td>
<td>• Manually Intervene if required</td>
</tr>
<tr>
<td>Onboard Control System</td>
<td>• Receive ABORT signal</td>
</tr>
<tr>
<td></td>
<td>• Command ABORT to ACS</td>
</tr>
<tr>
<td></td>
<td>• Command ABORT if required and not otherwise commanded</td>
</tr>
<tr>
<td></td>
<td>• Decrypt?</td>
</tr>
</tbody>
</table>

**Adapted from Dr Thomas’ STPA Tutorial**

William.Young.3@US.AF.Mil  WYOUNG@MIT.EDU  © Copyright William Young, Jr, 2019
Spacecraft—HCAs (Unsafe / Unsecure)

Adapted from Dr Thomas’ STPA Tutorial

© Copyright William Young, Jr, 2019
## Control Action Analysis Table (Step 1)

<table>
<thead>
<tr>
<th>Control Action</th>
<th>Not providing causes hazard</th>
<th>Providing causes hazard</th>
<th>Incorrect Timing or Order</th>
<th>Stopped too soon or applied too long</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA1: ABORT</td>
<td>OCS not providing ABORT command is hazardous when spacecraft closure is outside planned parameters in close proximity to ISS [H-1, H-2]</td>
<td>OCS providing ABORT command is hazardous when command places vehicle outside safe operating envelope [H-1, H-2]</td>
<td>OCS providing ABORT command too late is hazardous when corrective measures allow insufficient time to prevent collision [H-1, H-2]</td>
<td>OCS providing ABORT command for too short a period is hazardous when corrections are not applied long enough to prevent collision [H-1, H-2]</td>
</tr>
</tbody>
</table>

**HCA - Hazardous Control Action**

Adapted from Dr Thomas’ STPA Tutorial
Enterprise Architecture: Design Analysis

Establishing Initial Security Specifications
## Design Analysis Overview

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. <strong>Develop process model descriptions</strong></td>
<td>Describes the decision logic (“in plain English”) for executing a given CA.</td>
</tr>
<tr>
<td>2. <strong>Identify Process Model Variables (PMV)</strong></td>
<td>PMVs are measurable indicators of the conditions that trigger a CA.</td>
</tr>
<tr>
<td>3. <strong>Specify PMV values</strong></td>
<td>PMV values are all the possible values a PMV can be assigned both acceptable and hazardous.</td>
</tr>
<tr>
<td>4. <strong>Identify PMV sensors</strong></td>
<td>Identifies which sensors provide PMV values to the actors and controller for decision making.</td>
</tr>
<tr>
<td>5. <strong>Develop causal scenarios</strong></td>
<td>Brainstorm how a specific implementation of the system may be compromised. Identifies critical CAs and validates the thoroughness of the model, CAs, and constraints.</td>
</tr>
</tbody>
</table>
## Developing Process Model Descriptions

### Element: Onboard Control System

**Responsibilities:** Receive (encrypted) ABORT when issued by ISS or GSS, decrypt (if required), terminate unsafe maneuver, command Attitude Control System to return vehicle to a safe distance from ISS and safe operational parameters. OCS will be capable of automatically sensing and commanding the Attitude Control System to ABORT docking maneuver if unsafe conditions arise.

<table>
<thead>
<tr>
<th>Control Actions</th>
<th>Key Activity</th>
<th>Process Model Description / Decision Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>Docking</td>
<td>Issue ABORT Signal when___{context}___</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Issue ABORT Signal when___{context}___</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Issue ABORT Signal when___{context}___</td>
</tr>
</tbody>
</table>
### Developing Process Model Descriptions

**Element: Onboard Control System**

**Responsibilities:** Receive (encrypted) ABORT when issued by ISS or GSS, decrypt (if required), terminate unsafe maneuver, command Attitude Control System to return vehicle to a safe distance from ISS and safe operational parameters. OCS will be capable of automatically sensing and commanding the Attitude Control System to ABORT docking maneuver if unsafe conditions arise.

<table>
<thead>
<tr>
<th>Control Actions</th>
<th>Key Activity</th>
<th>Process Model Description / Decision Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>Docking</td>
<td>Issue ABORT when <strong>ABORT SIGNAL RECEIVED FROM GSS</strong> and Vehicle is X Distance from ISS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Issue ABORT when <strong>ABORT SIGNAL RECEIVED FROM ISS</strong> and Vehicle is X Distance from ISS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Issue ABORT Signal when <strong>UNSAFE MANEUVER SENSED</strong> and Vehicle is X Distance from ISS</td>
</tr>
</tbody>
</table>

William.Young.3@US.AF.Mil  WYOUNG@MIT.EDU  © Copyright William Young, Jr, 2019
### Identify Process Model Variables

**Element: Onboard Control System**

Responsibilities: Receive (encrypted) ABORT when issued by ISS or GSS, decrypt (if required), terminate unsafe maneuver, command Attitude Control System to return vehicle to a safe distance from ISS and safe operational parameters. OCS will be capable of automatically sensing and commanding the Attitude Control System to ABORT docking maneuver if unsafe conditions arise.

<table>
<thead>
<tr>
<th>Control Actions</th>
<th>Key Activity</th>
<th>Process Model Description / Decision Logic</th>
<th>Process Model Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>Docking</td>
<td>Issue ABORT when <em>ABORT SIGNAL RECEIVED FROM GSS</em> and <em>Vehicle is X Distance from ISS</em></td>
<td>1) ABORT Signal Received from GSS 2) Distance from ISS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Issue ABORT when <em>ABORT SIGNAL RECEIVED FROM ISS</em> and <em>Vehicle is X Distance from ISS</em></td>
<td>1) ABORT Signal Received from ISS 2) Distance from ISS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Issue ABORT when <em>UNSAFE MANEUVER SENSED</em> and <em>Vehicle is X Distance from ISS</em></td>
<td>1) Unsafe Maneuver Sensed 2) Distance from ISS</td>
</tr>
</tbody>
</table>
Specify Process Model Variable Values

- **ABORT Signal Received From GSS**
  - Yes
  - No
  - Unknown

- **ABORT Signal Received From ISS**
  - Yes
  - No
  - Unknown

- **Unsafe Maneuver Sensed**
  - Match
  - Mismatch
  - Unknown

- **Distance from ISS**
  - Close
  - Not Close
  - Unknown

How Should We Initially Specify the Values for “Distance to ISS”?
Specify Process Model Variable Values

### Issue ABORT (YES / NO)

<table>
<thead>
<tr>
<th>Issue ABORT (YES / NO)</th>
<th>ABORT Rec’d from GSS</th>
<th>ABORT Rec’d from ISS</th>
<th>UNSAFE Maneuver Sensed</th>
<th>Distance from ISS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>Unk</td>
<td>Close</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>Unk</td>
<td>Not Close</td>
</tr>
<tr>
<td></td>
<td>Mat</td>
<td>Mis</td>
<td>Unk</td>
<td>Unk</td>
</tr>
</tbody>
</table>

Complete Context Table (Truth Table for Potential Contexts)

Can Now Define When Onboard Control System Must and Must Not Invoke ABORT functionality

Entire Context Table Can Be Captured in Leveson’s SpecTRM-RL Tables
### Specify Process Model Variable Values

<table>
<thead>
<tr>
<th>Issue ABORT (YES / NO)</th>
<th>ABORT Rec’d from GSS</th>
<th>ABORT Rec’d from ISS</th>
<th>UNSAFE Maneuver Sensed</th>
<th>Distance from ISS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>Unk</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Complete Context Table (Truth Table for Potential Contexts)

Can Now Define When Onboard Control System Must and Must Not Invoke ABORT functionality

SpecTRM-RL Tables are Testable Software Specifications
Identify Process Model Variable Sensor Feedback

- Establish required feedback for each PMV
- How will each value be determined?
  - ABORT Command Received From GSS, ISS
  - Distance from ISS
  - Unsafe maneuver sensed
- Easily catch missing feedback in documents
Scenarios should be used to facilitate deeper insights and understanding, they are not a checklist.

Scenarios provide an opportunity to engage technical experts and ask key questions necessary to support improved requirements.

Scenarios form a connected narrative to understand and explain interactions across the system (and set appropriate requirements).

Scenarios should provide useful insight or generate additional questions for deeper debate and discussion.

- Scenarios such as “denial of service attack prevents controller from issuing ABORT command” aren’t really as useful as “controller doesn’t issue ABORT command when vehicle exceeds safe closure rate because ISS and GSS disagreed on need to ABORT.”
Potential causes of HCAs

Controller

- Onboard Control System does NOT Issue ABORT Command when required
- Inadequate Control Algorithm (Flaws in creation, process changes, incorrect modification or adaptation)
- Controller Process Model (inconsistent, incomplete, or incorrect)
- Control input or external information wrong or missing or malformed

Sensor

- Inadequate operation
- Wrong or missing feedback
- Feedback Delays

Actuator

- Inadequate, malformed or missing feedback
- Incorrect, partial or no information provided
- Measurement inaccuracies
- Feedback Delays

Controlled Process

- Component failures
- Changes over time
- Unidentified or out-of-range disturbance
- Process output contributes to system hazard
- Conflicting control actions

Controller

- Component failures
- Changes over time
- Unidentified or out-of-range disturbance
- Process output contributes to system hazard
- Conflicting control actions

Adapted from Dr Thomas’ STPA Tutorial

© Copyright William Young, Jr, 2019

WYOUNG@MIT.EDU
HCA: Onboard Control System (OCS) Does Not Command ABORT to Maneuver Subsystem after receiving ABORT signal from ISS and in close proximity BECAUSE ______SCENARIO______

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Associated Causal Factors</th>
<th>Rationale/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSS did not issue or confirm the command.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Scenario Discussion

**HCA: Onboard Control System (OCS) Does Not Command ABORT to Maneuver Subsystem after receiving ABORT signal from ISS and in close proximity BECAUSE _____SCENARIO_____**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Associated Causal Factors</th>
<th>Rationale/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSS did not issue or confirm the command.</td>
<td>• Malformed signal from GSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Partial signal from GSS</td>
<td>Malicious logic on OCS reports false/delayed/malformed information.</td>
</tr>
<tr>
<td></td>
<td>• Missing signal from GSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Inconsistent process model</td>
<td></td>
</tr>
</tbody>
</table>

- Malicious logic on computer modifies process model variable to indicate that ISS is NOT in close proximity.
Potential control actions not followed

- Missing or wrong or unauthorized communication with another controller
- Inadequate, malformed or missing feedback
- Incorrect, partial or no information provided
- Incorrect Control input or external information wrong or missing or malformed
- Inappropriate, ineffective, malformed, or missing control action
- Inadequate operation
- Component failures
- Changes over time
- Conflicting control actions
- Delayed, partial, or malformed operation
- Unidentified or out-of-range disturbance
- Process output contributes to system hazard
- Process input input missing or wrong

Adapted from Dr Thomas’ STPA Tutorial

© Copyright William Young, Jr, 2017

WYOUNG@MIT.EDU
Scenario Discussion

HCA: Onboard Control System provides ABORT command in close proximity to ISS after receiving ABORT signal from ISS & GSS and close proximity but Maneuver Subsystem does not execute ABORT functionality BECAUSE ___ Scenario ___

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Associated Causal Factors</th>
<th>Rationale/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maneuver subsystem prioritizes inputs from its internal measurements on whether or not vehicle has exceeded safe docking parameters. Does not adequately handle a case where local sensor data is incorrect AND there are still good comms with ISS / GSS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Adapted from Dr Thomas’ STPA Tutorial
HCA: Onboard Control System provides ABORT command in close proximity to ISS after receiving ABORT signal from ISS & GSS and close proximity but Maneuver Subsystem does not execute ABORT functionality BECAUSE ___ Scenario___

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Associated Causal Factors</th>
<th>Rationale/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maneuver subsystem prioritizes inputs from its internal measurements on whether or not vehicle has exceeded safe docking parameters. Does not adequately handle a case where local sensor data is incorrect AND there are still good comms with ISS / GSS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
  • Inadequate control algorithm  
  • Potential conflicting control between Maneuver subsystem and Onboard control system | Attacking sensor inside Maneuver Subsystem creates the potential to block GSS/ISS if the ABORT logic requires onboard confirmation that the vehicle is in close proximity or outside parameters. |
Blue focus on Enforcing Constraint, Red focus on violating constraint...
Goal is to “Fix” Problem Through Elimination or Mitigation Above Component Level
User Questions and Answers
Summary and Conclusions
Lessons Learned Applying STPA-Sec

• Often heard comments:
  • “You’re starting at a much higher level of abstraction…”
  • “We try to do something like that, but STPA-Sec is much more rigorous…”
  • “This requires a great deal of thought…from more than just security experts”

• Difficult or impossible to implement if system owner is unable cannot specify what system is supposed to do

• Initial expert guess on what is most important to assure tends to be too broad to be actionable
  • E.g. “Power grid”

STPA-Sec is NOT a silver bullet, but appears to enable increased rigor “Left of Design”
Safety and Security

• Goal is loss prevention and risk management

• Source is probably irrelevant and may be unknowable

• Method is the development and engineering of controls

• Focus on what we have the ability to address, not the environment

• STPA/STPA-Sec provide opportunity for a unified and integrated effort through shared control structure!
Conclusion

• Must think carefully about defining the security problem

• Perfectly solving the wrong security problem doesn’t really help

• STPA-Sec provides a means to clearly link security to the broader mission or business objectives

• STPA-Sec does not replace existing security engineering methods, but enhances their effectiveness
Concluding Thoughts from Sun Tzu

The opportunity to secure ourselves against defeat lies in our own hands.

The supreme art of war is to subdue the enemy without fighting.

Strategy without tactics is the slowest route to victory. Tactics without strategy is the noise before defeat.
QUESTIONS ??