Introduction to risk management in complex systems

Dr. John Thomas
Massachusetts Institute of Technology
Agenda

• Introduction to system safety
  – Challenges for complex systems
  – Goals

• System-theoretic Process Analysis

• Application to a proton beam therapy machine
Three Mile Island

Events: A critical relief valve fails (stuck open) and begins venting coolant. Despite best efforts, operators are unable to mitigate this problem in time and the reactor experiences a meltdown. Radioactive materials are released. $1B cleanup costs.
Component failure accidents

• These are accidents caused by physical component failures
  – E.g. valve stuck open

• What would you do about this?

• Beware of “tunnel vision”
  – Very easy to focus only on the physical failure
  – There are usually deeper systemic factors too
Three Mile Island

Events: A critical relief valve fails (stuck open) and begins venting coolant. Despite best efforts, **operators are unable to mitigate this problem in time** and the reactor experiences a meltdown. Radioactive materials are released. $1B cleanup costs.

**Systemic factors?**
Causal Factors:

• Post-accident examination discovered the “open valve” indicator light was configured to show presence of power to the valve (regardless of valve position).

Design flaw!
Communication problems!
Inadequate procedures!
Etc.
System safety

• Modern systems involve complex interactions between many components
  – Software, hardware, human operators, environment, management, maintenance etc.
  – Interactions can be overlooked when components considered in isolation
  – Need to understand the whole system of interactions
  – Unanticipated and unexpected emergent system behavior

• Need to include systemic factors
  – Not just component failures
Goals for a systemic approach

• Need to address component failure accidents
  – Identify important failures, but also go beyond the failures
  – Why weren’t the failures detected and mitigated?
  – Human-computer interaction issues?
  – Software-induced operator error?
  – Etc.

• What else is needed?
Mars Polar Lander

- During the descent to Mars, the legs were deployed at an altitude of 40 meters.
- Touchdown sensors (on the legs) sent a momentary signal.
- The software responded as it was required to: by shutting down the descent engines.
- The vehicle free-fell and was destroyed upon hitting the surface at 50 mph.

No single component failed. All components performed as designed.
Component interaction accidents

• ... are accidents caused by interactions among several components
  – May not involve any component failures
  – All components may operate as designed
    • But the design may be wrong
    • Requirements may be flawed
  – Related to complexity
    • Becoming increasingly common in complex systems
    • Complexity of interactions leads to unexpected system behavior
    • Difficult to anticipate unsafe interactions
  – Especially problematic for software
    • Software always operates as designed
Goals for a systemic approach

• Need to address component failure accidents
• Need to address component interaction accidents

• What else?
2013 Ford Fusion / Escape

*Images from:*
http://gearheads.org/stop-driving-your-ford-escape/

© Copyright John Thomas 2016
2013 Ford Fusion / Escape

• Engine fires
  – 13 reports of engine fire
  – Short time frame
    • (~Sept - Dec)
• Ford asks all owners to “park their vehicles until further notice”
• 99,153 brand new vehicles affected

Images from:
http://www.unionleader.com/article/20130119/NEWS03/130119090
The Problem

• Ford press release:
  – “The original cooling system design was not able to address a loss of coolant system pressure under certain operating conditions, which could lead to a vehicle fire while the engine was running.”

• Ford VP:
  – "We had a sequence of events that caused the cooling system software to restrict coolant flow," he says. Most of the time, that would not be a problem and is the intended behavior. But in rare cases the coolant pressure coupled with other conditions may cause the coolant to boil. When the coolant boils, the engine may go into extreme overheating causing more boiling and rapid pressure increase. This caused coolant leaks near the hot exhaust that led to an engine fire.
  – Ford has seen 12 fires in Escapes and one in a Fusion.

System requirements (and the engineers) never anticipated this worst-case possibility

Quotes from:
http://www.usatoday.com/story/money/cars/2012/12/10/ford-recall-escape-fusion-ecoboost/1759063/
“The hardest single part of building a software system is deciding precisely what to build.”

-- Fred Brooks, *The Mythical Man-Month*
“The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements ... No other part of the work so cripples the resulting system if done wrong. No other part is as difficult to rectify later.”

-- Fred Brooks, The Mythical Man-Month
Goals for a systemic approach

• Need to address component failure accidents
• Need to address component interaction accidents
• Need a worst-case analysis, not best case or most likely case
• Handle broad array of causes
  – Incorrect assumptions
  – Incorrect/incomplete requirements
  – Complex software behavior
    • In fact, most software-related accidents are caused by requirements flaws, not coding errors or failures
  – Design errors
  – Component failures

• What else?
Toyota

- **2004**: Push-button ignition
- **2004-2009**
  - 102 incidents of uncontrolled acceleration
  - Speeds exceed 100 mph despite stomping on the brake
  - 30 crashes
  - 20 injuries
- **2009, Aug**:
  - Car accelerates to 120 mph
  - Passenger calls 911, reports stuck accelerator
  - Car crashes killing 4 people
  - Driver was offensive driving instructor for police
- **Today**
  - Software fixes for pushbutton ignition, pedals

Pushbutton was reliable, Software was reliable.
All requirements were met.
Didn’t account for human behavior!

http://www.reuters.com/article/2010/07/14/us-toyota-idUSTRE66D0FR20100714
Toyota

- **2004**: Push-button ignition
- **2004-2009**
  - 102 incidents of uncontrolled acceleration
  - Speeds exceed 100 mph despite stomping on the brake
  - 30 crashes
  - 20 injuries
- **2009, Aug**:
  - Car accelerates to 120 mph
  - Passenger calls 911, reports stuck accelerator
  - Car crashes killing 4 people
  - Driver was offensive driving instructor for police
- **Today**
  - Software fixes for pushbutton ignition, pedals

In complex systems, human and technical considerations cannot be isolated

http://www.reuters.com/article/2010/07/14/us-toyota-idUSTRE66D0FR20100714

© Copyright John Thomas 2016
Goals for a systemic approach

- Need to address component failure accidents
- Need to address component interaction accidents
- Need a worst-case analysis, not best case or most likely case
- Handle broad array of causes
- Must account for human behavior / social factors
  - Easy to treat human error as a separate issue
  - Easy to look no deeper than human-machine interfaces
  But must also consider:
    - “Clumsy automation”, mode confusion, etc.
    - How technology might induce human error
    - Human error often a symptom of deeper trouble (Dekker)
      - To fix, need to understand why it would make sense at the time
Human Factors: Old View

• Human error is cause of most incidents and accidents
• So do something about human involved
  – Fire them
  – Retrain them
  – Admonish them
  – Rigidify their work with more rules and procedures
• Or do something about humans in general
  – Marginalize them by putting in more automation

Leveson, 2012
Human Factors: **Systems View**

*(Dekker, Rasmussen, Leveson, Woods, etc.)*

• Human error is a symptom, not a cause

• All behavior affected by context (system) in which it occurs
  – To understand human error, look at the system
  – Systems are stretching limits of comprehensibility
  – System designs can make human error inevitable

• To do something about operator error, look at:
  – Design of equipment
  – Usefulness of procedures
  – Existence of goal conflicts and production pressures

• Human error is a symptom of the system and its design
Most stove tops

Human error?

*Image from D. Norman, 1988
Natural Mapping

Human error? Or design problem?

*Image from D. Norman, 1988

© Copyright John Thomas 2016
China Airlines 006

- Autopilot compensates for single engine malfunction
- Autopilot reaches max limits, aircraft turns slightly
- Pilots not notified Autopilot at its limits
- Pilots notice slight turn, disengage autopilot for manual control
  - Aircraft enters nosedive
Goals for a systemic approach

• Need to address component failure accidents
• Need to address component interaction accidents
• Need a worst-case analysis, not best case or most likely case
• Handle broad array of causes
• Must account for human behavior / social factors

• What else?
A note about hindsight bias

- After an accident, hindsight makes causes seem obvious
- In engineering, 1000s of variables and potential problems to consider
- Many of these problems only seem obvious after-the-fact

(Dekker, 2009)
Safety vs. reliability

Reliability $\leftrightarrow$ Failures

Safety $\leftrightarrow$ Accidents
Safety vs. Reliability

• Failure analysis is a *reliability* technique
  – Inefficient for safety: analyzes non-safety-related failures
  – Insufficient for safety: may overlook non-failure accidents

• Failure analysis sometimes used as part of a safety analysis
  – Can (inefficiently) establish the end effects of failures

© Copyright John Thomas 2016
Safe ≠ Reliable

- Safety often means making sure X never happens
- Reliability usually means making sure Y always happens

<table>
<thead>
<tr>
<th>Safe</th>
<th>Unsafe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliable</td>
<td>• Typical commercial flight</td>
</tr>
<tr>
<td>Unreliable</td>
<td>• Aircraft engine fails in flight</td>
</tr>
</tbody>
</table>
Safe ≠ Reliable

- Safety often means making sure X never happens
- Reliability usually means making sure Y always happens

<table>
<thead>
<tr>
<th>Safe</th>
<th>Unsafe</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reliable</strong></td>
<td>• Typical commercial flight</td>
</tr>
<tr>
<td></td>
<td>• Computer reliably executes unsafe commands</td>
</tr>
<tr>
<td></td>
<td>• Increasing tank burst pressure</td>
</tr>
<tr>
<td></td>
<td>• Retreating to safe state vs. achieving mission</td>
</tr>
<tr>
<td></td>
<td>• A nail gun without safety lockout</td>
</tr>
<tr>
<td><strong>Unreliable</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Aircraft engine fails in flight</td>
</tr>
</tbody>
</table>
Safe ≠ Reliable

- Safety often means making sure $X$ never happens
- Reliability usually means making sure $Y$ always happens

<table>
<thead>
<tr>
<th></th>
<th>Safe</th>
<th>Unsafe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliable</td>
<td>• Typical commercial flight</td>
<td>• Computer reliably executes unsafe commands</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Increasing tank burst pressure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Retreating to safe state vs. achieving mission</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• A nail gun without safety lockout</td>
</tr>
<tr>
<td>Unreliable</td>
<td>• Aircraft engine won’t start on ground?</td>
<td>• Aircraft engine fails in flight</td>
</tr>
<tr>
<td></td>
<td>• Missile won’t fire?</td>
<td></td>
</tr>
</tbody>
</table>
Fault Modelling, Fault Injection

• Not enough to ensure safety
• Faults must be known in advance
  – Works well for some components, well-understood & established history
  – May be unknown for new components, or old components in new environment
    • E.g. **NASA injector vibrations**, Apollo switches, Ariane 5, etc.
  – Unk Unks
• Effect of fault must be known, accurate
  – Non-deterministic effects can be tricky (e.g. noise in nuclear detonation circuits, car stereo EMI)
• Multiple-point failures
  – Simulating all combinations of faults can be impractical
• May overlook accidents that occur with no failures
Goals for a systemic approach

• Need to address component failure accidents
• Need to address component interaction accidents
• Need a worst-case analysis, not best case or most likely case
• Handle broad array of causes
• Must account for human behavior / social factors
• Need to distinguish safety vs. reliability goals

• What else?
Building Safety into the System

Early decisions tend to have biggest impact on safety

Need to address safety early

Illustration courtesy Bill Young, MIT
Goals for a systemic approach

• Need to address component failure accidents
• Need to address component interaction accidents
• Need a worst-case analysis, not best case or most likely case
• Handle broad array of causes
• Must account for human behavior / social factors
• Need to distinguish safety vs. reliability goals
• Must be applicable as early as possible
  – Drive the design and requirements instead of causing rework
Boeing 787 Lithium Battery Fires

- 2013 – 2014
- Reliability analysis predicted 10 million flight hours between battery failures
  - Two fires caused by battery failures in 52,000 flight hours
  - Does not include 3 other less-reported incidents of smoke in battery compartment

Another simple component failure accident?
Why is this so hard?

• Coupling
  – Highly coupled systems have more interdependence
  – Number of dependencies can increase exponentially

• Indirect causality
  – Cause and effect may not be related in an obvious or direct way

• Interactive complexity
  – Number of possible interactions can challenge our ability to analyze and identify dangerous interactions

• Intellectual manageability
  – A simple system has a small number of unknowns in its interactions (within system and with environment)
  – Intellectually unmanageable when level of interactions reaches point can no longer be thoroughly
    • Planned
    • Understood
    • Anticipated
    • Guarded against

Leveson, 2012, 1995
Safety vs. Reliability: another difference

Using standard engineering techniques of:
- Redundancy
- Increasing reliability
- Reusing designs in new environments

typically increases complexity:
- NASA pyrovalve example, Apollo computers

Solutions that add complexity will not solve problems that stem from intellectual unmanageability and interactive complexity

Redundancy does not work for component interaction accidents
How to manage complexity?

- A lesson from systems theory, cognitive science
- Human minds manage complexity through abstraction and hierarchy
- Use top-down processes
  - Start at a high abstract level
  - Iterate to drill down into more detail
  - Build hierarchical models of the system

© Copyright John Thomas 2016
Goals for a systemic approach

• Need to address component failure accidents
• Need to address component interaction accidents
• Need a worst-case analysis, not best case or most likely case
• Handle broad array of causes
• Must account for human behavior / social factors
• Need to distinguish safety vs. reliability goals
• Must be applicable as early as possible
• Provide ways to manage complexity
  – Top-down processes
  – Improve intellectual manageability
A systems approach to safety: STAMP and STPA
Systems approach to safety engineering (STAMP)

• Accidents are more than a chain of events, they involve complex dynamic processes.
• Treat accidents as a control problem, not a failure problem
• Prevent accidents by enforcing constraints on component behavior and interactions
• Captures more causes of accidents:
  – Component failure accidents
  – Unsafe interactions among components
  – Complex human, software behavior
  – Design errors
  – Flawed requirements
    • esp. software-related accidents
Controllers use a **process model** to determine control actions.

Accidents often occur when the process model is incorrect.

Four types of **unsafe control actions**:
1. Control commands required for safety are not given
2. Unsafe ones are given
3. Potentially safe commands but given too early, too late
4. Control action stops too soon or applied too long

Tends to be a good model of both software and human behavior.
Explains software errors, human errors, interaction accidents, ...
Controlled Process

Controller

Process Model

Control Actions

Feedback

Operating Process

- Human Controller(s)
- Automated Controller
- Actuator(s)
- Sensor(s)
- Physical Process

Operating Assumptions
Operating Procedures

Revised operating procedures
Software revisions
Hardware replacements

Problem Reports
Incidents
Change Requests
Performance Audits

© Copyright John Thomas 2016
Example
Safety
Control
Structure

SYSTEM DEVELOPMENT

Congress and Legislatures
- Legislation
  - Government Reports
  - Lobbying
  - Hearings and open meetings
  - Accidents

Government Regulatory Agencies
- Industry Associations, User Associations, Unions, Insurance Companies, Courts
  - Regulations
  - Standards
  - Certification
  - Legal penalties
  - Case Law

Company Management
- Safety Policy
  - Standards
  - Resources
- Status Reports
  - Risk Assessments
  - Incident Reports

Design, Documentation
- Safety Standards
  - Test Requirements
- Hazard Analyses
  - Progress Reports
- Test reports
  - Hazard Analyses
  - Review Results

Implementation and assurance
- Safety Reports
- Operating Process
  - Human Controller(s)
  - Automated Controller
  - Actuator(s)
  - Sensor(s)
  - Physical Process

Manufacturing Management
- Work Procedures
  - Audits
  - Work logs
  - Inspections

Maintenance and Evolution
- Problem Reports
  - Incidents
  - Change Requests
  - Performance Audits

SYSTEM OPERATIONS

Congress and Legislatures
- Legislation
  - Government Reports
  - Lobbying
  - Hearings and open meetings
  - Accidents

Government Regulatory Agencies
- Industry Associations, User Associations, Unions, Insurance Companies, Courts
  - Regulations
  - Standards
  - Certification
  - Legal penalties
  - Case Law

Company Management
- Safety Policy
  - Standards
  - Resources
- Operations Reports
  - Change requests
  - Audit reports
  - Problem reports

Operations Management
- Work Instructions
  - Operating Assumptions
  - Operating Procedures
- Software revisions
  - Hardware replacements

Leveson, 2012
STAMP and STPA

STAMP Model

Accidents are caused by inadequate control
Accidents are caused by inadequate control in a design. How do we find inadequate control in a design?
STPA
(System-Theoretic Process Analysis)

- Identify accidents and hazards
- Construct the control structure
- **Step 1:** Identify unsafe control actions
- **Step 2:** Identify causal factors and control flaws

(Leveson, 2012)
Definitions

• Accident (Loss)
  – An undesired or unplanned event that results in a loss, including loss of human life or human injury, property damage, environmental pollution, mission loss, etc.

• Hazard
  – A system state or set of conditions that, together with a particular set of worst-case environment conditions, will lead to an accident (loss).

Leveson, 2012, 1995
Definitions

• Accident (Loss)
  – An undesired or unplanned event that results in a loss, including loss of human life or human injury, property damage, environmental pollution, mission loss, etc.
  – May involve environmental factors outside our control

• Hazard
  – A system state or set of conditions that, together with a particular set of worst-case environment conditions, will lead to an accident (loss).
  – Something we can control in the design

<table>
<thead>
<tr>
<th>Accident</th>
<th>System Hazard</th>
</tr>
</thead>
<tbody>
<tr>
<td>People die from exposure to toxic chemicals</td>
<td>Toxic chemicals from the plant are in the atmosphere</td>
</tr>
<tr>
<td>People die from radiation sickness</td>
<td>Nuclear power plant radioactive materials are not contained</td>
</tr>
<tr>
<td>Vehicle collides with another vehicle</td>
<td>Vehicles do not maintain safe distance from each other</td>
</tr>
<tr>
<td>People die from food poisoning</td>
<td>Food products for sale contain pathogens</td>
</tr>
</tbody>
</table>
# Definitions

- **Accident (Loss)**
  - An undesired or unplanned event that results in a loss, including loss of human life or human injury, property damage, environmental pollution, mission loss, etc.

<table>
<thead>
<tr>
<th>People die from radiation sickness</th>
<th>Nuclear power plant radioactive materials are not contained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle collides with another vehicle</td>
<td>Vehicles do not maintain safe distance from each other</td>
</tr>
<tr>
<td>People die from food poisoning</td>
<td>Food products for sale contain pathogens</td>
</tr>
</tbody>
</table>

---

**Broad view of safety**

“Accident” is anything that is unacceptable, that must be prevented. Not limited to loss of life or human injury!
## System Safety Constraints

<table>
<thead>
<tr>
<th>System Hazard</th>
<th>System Safety Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toxic chemicals from the plant are in the atmosphere</td>
<td>Toxic plant chemicals must not be released into the atmosphere</td>
</tr>
<tr>
<td>Nuclear power plant radioactive materials are not contained</td>
<td>Radioactive materials must not be released</td>
</tr>
<tr>
<td>Vehicles do not maintain safe distance from each other</td>
<td>Vehicles must always maintain safe distances from each other</td>
</tr>
<tr>
<td>Food products for sale contain pathogens</td>
<td>Food products with pathogens must not be sold</td>
</tr>
</tbody>
</table>

Additional hazards / constraints can be found in ESW p355
Proton Radiation Therapy System
Paul Scherrer Institute, Switzerland

• Accidents?
• Hazards?
Proton Therapy Machine (Antoine)

• Accidents
  – ACC1. Patient injury or death
  – ACC2. Ineffective treatment
  – ACC3. Loss to non-patient quality of life (esp. personnel)
  – ACC4. Facility or equipment damage

• Hazards
Proton Therapy Machine (Antoine)

- Accidents
  - ACC1. Patient injury or death
  - ACC2. Ineffective treatment
  - ACC3. Loss to non-patient quality of life (esp. personnel)
  - ACC4. Facility or equipment damage

- Hazards
  - H-R1. Patient tissues receive more dose than clinically desirable
  - H-R2. Patient tumor receives less dose than clinically desirable
  - H-R3. Non-patient (esp. personnel) is unnecessarily exposed to radiation
  - H-R4. Equipment is subject to unnecessary stress
Control Structures
Chemical Plant

Image from: http://www.cbgnetwork.org/2608.html
Chemical Plant

Citichem Safety Control Structure

- Corporate Management
- Corporate Inventory Control
- Plant Management
- Plant Inventory Control
- Operations Management
- Maintenance Management
- Control Room Operators
- Plant Operators
- Physical Equipment
- Chemical Process

OSHA

Oakbridge Community Safety Control Structure

- City Government
- Local Citizens
- Developers
- Local Business
- Emergency Response
- Public Health

Image from: http://www.cbgnetwork.org/2608.html

© Copyright John Thomas 2016
Adaptive Cruise Control

Ballistic Missile Defense System


Safeware Corporation
U.S. pharmaceutical safety control structure

Congress → FDA → Pharmaceutical Companies → Doctors → Patients

Proton Radiation Therapy System
Paul Scherrer Institute, Switzerland
Proton Radiation Therapy System
Paul Scherrer Institute, Switzerland

- 250 MeV Proton accelerator (superconducting cyclotron)
- Beamlines to 4 user areas
- OPTIS
- Gantry 1
- Gantry 2
- Experimental area
Proton Radiation Therapy System
Gantry 1
Proton Radiation Therapy System
Spot Scanning Technique

Elements of spot scanning:

- Beam on/off 50 μs
- Sweeper magnet 5 ms/step
- Range shifter 30 ms
- Patient table 1 cm/s

- 10,000 spots to treat 1 liter volume
Proton Radiation Therapy System
Gantry 2
Proton Therapy Machine Overview

- Gantry
- Beam path and control elements
- Cyclotron

© Copyright John Thomas 2016
Proton Therapy Machine
High-level Control Structure

• How big do you think the high-level control structure is?
Proton Therapy Machine

High-level Control Structure

Figure 11 - High-level functional description of the PROSCAN facility (D0)
Proton Therapy Machine Control Structure

Proton Therapy Machine Detailed Control Structure

STPA
(System-Theoretic Process Analysis)

- Identify accidents and hazards
- Construct the control structure
- Step 1: Identify unsafe control actions
- Step 2: Identify causal factors and control flaws

(Leveson, 2012)
STPA Step 1: Unsafe Control Actions (UCA)

4 ways unsafe control may occur:

- A control action required for safety is not provided or is not followed
- An unsafe control action is provided that leads to a hazard
- A potentially safe control action provided too late, too early, or out of sequence
- A safe control action is stopped too soon or applied too long (for a continuous or non-discrete control action)
Unsafe Control Actions

Start Treatment Command

- Not provided causes hazard?
- Providing causes hazard?
- Too early/late?
- Wrong order?
- Stopped too soon, applied too long?
Step 1: Identify Unsafe Control Actions

<table>
<thead>
<tr>
<th>Control Action</th>
<th>Not providing causes hazard</th>
<th>Providing causes hazard</th>
<th>Too early/too late, wrong order</th>
<th>Stopped too soon/ applied too long</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Treatment Command</td>
<td></td>
<td>Operator provides Start Treatment cmd while personnel is in room (↑H-R3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

System Hazards
H-R1. Patient tissues receive more dose than clinically desirable
H-R2. Patient tumor receives less dose than clinically desirable
H-R3. Non-patient (esp. personnel) is unnecessarily exposed to radiation
H-R4. Equipment is subject to unnecessary stress
Structure of an Unsafe Control Action

Example:
“Operator provides start treatment cmd while personnel is in room”

Four parts of an unsafe control action
– Source Controller: the controller that can provide the control action
– Type: whether the control action was provided or not provided
– Control Action: the controller’s command that was provided / missing
– Context: conditions for the hazard to occur
  • (system or environmental state in which command is provided)
Unsafe control action summary

• UCA1. Treatment is started while personnel is in room (H-R3)
• UCA2. Treatment is started while patient is not ready to receive treatment (H-R1, H-R2)
  – Note: This includes “wrong patient position”, “patient feeling unwell”, etc.
• UCA3. Treatment is started when there is no patient at the treatment point (H-R2)
• UCA4. Treatment is started with the wrong treatment plan (H-R1, H-R2)
• UCA5. Treatment is started without a treatment plan having been loaded (H-R1, H-R2)
• UCA6. Treatment is started while the beamline is not ready to receive the beam (H-R1, H-R4)
• UCA7. Treatment is started while not having mastership (H-R1, H-R2, H-R3)
• UCA8. Treatment is started while facility is in non-treatment mode (e.g. experiment or trouble shooting mode) (H-R1, H-R2)
• UCA9. Treatment start command is issued after treatment has already started (H-R1, H-R2)
• UCA10. Treatment start command is issued after treatment has been interrupted and without the interruption having adequately been recorded or accounted for (H-R1, H-R2)
• UCA11. Treatment does not start while everything else is otherwise ready (H-R1, H-R2)
## Component Safety Constraints

<table>
<thead>
<tr>
<th>Unsafe Control Action</th>
<th>Component Safety Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatment is started while personnel is in room</td>
<td>Treatment must not be started while personnel are in the room</td>
</tr>
<tr>
<td>Treatment is started while the beamline is not ready to receive the beam</td>
<td>Treatment must not start before beamline is fully configured</td>
</tr>
<tr>
<td>Treatment is started when there is no patient at the treatment point</td>
<td>Treatment must not start until when patient is at the treatment point</td>
</tr>
<tr>
<td>Treatment is started without a treatment plan having been loaded</td>
<td>Treatment must not start until a new treatment plan has been loaded</td>
</tr>
</tbody>
</table>
STPA
(System-Theoretic Process Analysis)

- Identify accidents and hazards
- Construct the control structure
- Step 1: Identify unsafe control actions
- Step 2: Identify causal factors and control flaws

(Leveson, 2012)
STPA Step 2: Identify Control Flaws

- **Unsafe Control Actions**
  - Inappropriate, ineffective, or missing control action

- **Controller**
  - Inadequate Control Algorithm (Flaws in creation, process changes, incorrect modification or adaptation)
  - Process Model (inconsistent, incomplete, or incorrect)
  - Control input or external information wrong or missing
  - Missing or wrong communication with another controller

- **Actuator**
  - Inadequate operation
  - Delayed operation

- **Sensor**
  - Inadequate operation
  - Incorrect or no information provided
  - Incorrect or no information provided
  - Feedback Delays

- **Controlled Process**
  - Component failures
  - Changes over time
  - Unidentified or out-of-range disturbance
  - Process output contributes to system hazard

- **Controller**
  - Process input missing or wrong
  - Conflicting control actions

© Copyright John Thomas 2016
STPA Step 2: Identify Causal Factors

• Select an Unsafe Control Action
  A. Identify causal factors that explain how it could happen
     • Develop causal accident scenarios
  B. Identify causal factors that explain how control actions may not be followed or executed properly
     • Develop causal accident scenarios

• Identify controls and mitigations for the accident scenarios
Step 2A: Potential causes of UCAs

UCA2. Operator starts treatment while patient is not ready to receive treatment

Controller
- Inadequate Procedures (Flaws in creation, process changes, incorrect modification or adaptation)
- Process Model (inconsistent, incomplete, or incorrect)
  - Control input or external information wrong or missing
  - Missing or wrong communication with another controller

Actuator
- Inadequate operation
  - Delayed operation

Sensor
- Inadequate operation
  - Inaccurate or no information provided
  - Measurement inaccuracies
  - Feedback delays

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

Controller
- Process input missing or wrong

© Copyright John Thomas 2016
STPA Step 2: Identify Causal Factors

• Select an Unsafe Control Action
  
  A. Identify causal factors that explain how it could happen
     • Develop causal accident scenarios
  
  B. Identify causal factors that explain how control actions may not be followed or executed properly
     • Develop causal accident scenarios

• Identify controls and mitigations for the accident scenarios
Step 2B: Potential control actions not followed

Treatment is started while patient is ready

- Inadequate Procedures (Flaws in creation, process changes, incorrect modification or adaptation)
- Controller
- Process Model (inconsistent, incomplete, or incorrect)
- Control input or external information wrong or missing
- Missing or wrong communication with another controller
- Controller
- Controller

Treatment is administered while patient is not ready

- Inadequate operation
- Actuator
- Inadequate operation
- Sensor
- Controller
- Delayed operation
- Controller
- Conflicting control actions
- Process input missing or wrong
- Component failures
- Controlled Process
- Changes over time
- Unidentified or out-of-range disturbance
- Process output contributes to system hazard
- Feedback Delays
- Incorrect or no information provided
- Measurement inaccuracies
- Feedback delays
- Inadequate or missing feedback

© Copyright John Thomas 2016
STPA Step 2: Identify Causal Factors

• Select an Unsafe Control Action
  A. Identify causal factors that explain how it could happen
     • Develop causal accident scenarios
  B. Identify causal factors that explain how control actions may not be followed or executed properly
     • Develop causal accident scenarios

Identify controls and mitigations for the accident scenarios
Example Controls for Causal Scenarios

• **Scenario 1** – Operator provides Start Treatment command when there is no patient on the table or patient is not ready. Operator was not in the room when the command was issued, as required by other safety constraints. Operator was expecting patient to have been positioned, but table positioning was delayed compared to plan (e.g. because of delays in patient preparation or patient transfer to treatment area; because of unexpected delays in beam availability or technical issues being processed by other personnel without proper communication with the operator).

• **Controls:**
  – Provide operator with direct visual feedback to the gantry coupling point, and require check that patient has been positioned before starting treatment (M1).
  – Provide a physical interlock that prevents beam-on unless table positioned according to plan
Example Controls for Causal Scenarios

• **Scenario 2** — Operator provides start treatment command when there is no patient. The operator was asked to turn the beam on outside of a treatment sequence (e.g. because the design team wants to troubleshoot a problem, or for experimental purposes) but inadvertently starts treatment and does not realize that the facility proceeds with reading the treatment plan and records the dose as being administered.

• **Controls:**
  – Reduce the likelihood that non-treatment activities have access to treatment-related input by creating a non-treatment mode to be used for QA and experiments, during which facility does not read treatment plans that may have been previously been loaded (M2);
  – Make procedures (including button design if pushing a button is what starts treatment) to start treatment sufficiently different from non-treatment beam on procedures that the confusion is unlikely.
Example Controls for Causal Scenarios
Command not followed

• **Scenario 3** — The operator provides the Start Treatment command, but it does not execute properly because the proper steering file failed to load (either because operator did not load it, or previous plan was not erased from system memory and overwriting is not possible) or the system uses a previously loaded one by default.

• **Controls:**
  – When fraction delivery is completed, the used steering file could for example be automatically dumped out of the system’s memory (M4).
  
  – Do not allow a Start Treatment command if the steering file does not load properly

  – Provide additional checks to ensure the steering file matches the current patient (e.g. barcode wrist bands, physiological attributes, etc.)

How does STPA compare?

- **MIT: TCAS**
  - Existing high quality fault tree done by MITRE for FAA
  - MIT comparison: STPA captured everything in fault tree, plus more

- **JAXA: HTV**
  - Existing fault tree reviewed by NASA
  - JAXA comparison: STPA captured everything in fault tree, plus more

- **EPRI: HPCI/RCIC**
  - Existing fault tree & FMEA overlooked causes of real accident
  - EPRI comparison: Blind study, only STPA found actual accident scenario

- **NRC: Power plant safety systems**
  - Proposed design that successfully completed Final Safety Analysis Report
  - STPA found additional issues that had not been considered

- **Safeware: U.S. Missile Defense Agency BMDS**
  - Existing hazard analysis per U.S. military standards
  - Safeware comparison: STPA captured existing causes plus more
  - STPA took 2 people 3 months, MDA took 6 months to fix problems

- **Automotive: EPS**
  - Compare STPA results to FMECA using SAE J1739

- **MIT: NextGen ITP**
  - Existing fault tree & event tree analysis by RTCA
  - MIT comparison: STPA captured everything in fault tree, plus more

- **MIT: Blood gas analyzer**
  - Existing FMEA found 75 accident causes
  - STPA by S.M. student found 175 accident causes
  - STPA took less effort, found 9 scenarios that led to FDA Class 1 recall
MIT March Workshop (free)

**Industries:**
Automotive  
Oil and Gas  
Space  
Aviation  
Defense  
Nuclear  
Healthcare and Healthcare IT  
Medical Devices  
Academia  
Insurance  
Academia (Education)  
Hydropower  
Chemicals  
Software/Computing  
Government  
Industrial Automation  
Electric Utility  
Security  
Think Tank  
Transportation  
Maritime (security)  
Environmental  
Pharmaceuticals  
Internet

**Organizations:**
The Boeing Company  
Boeing Environment Health and Safety  
Boeing Engineering and Operations  
Embraer  
U.S. Nuclear Regulatory Commission  
GE Aviation  
Sikorsky  
Thoratec Corporation  
University of Alabama in Huntsville  
Liberty Mutual Safety Research Institute  
ITA (Instituto Tecnologico de Aeronautica)  
Jeppesen  
Beijing Institute of Technology  
TEGMA Gestao Logistica S.A.  
Amsterdam University of Applied Sciences  
Dutch Safety Agency  
University of Stuttgart  
BC Hydro  
Therapeutic Goods Administration  
Institute of Aeronautics and Space (IAE), Brazil  
Shell Oil  
University of Braunschweig  
StiKi  
Reykjavik University  
National Nuclear Energy Commission, Brazil  
FAA  
U.S. Department of Transportation  
U.S. Air Force  
U.S. Navy  
IPEV (Institute for Research and Flight Testing), Brazil  
Japan Aerospace Exploration Agency (JAXA)  
U.S. Department of Energy Rockwell Automation  
Democritus University of Thrace  
Dependable Management Excellence and Safety  
Washington CORE  
Florida Institute of Technology  
U.S. Navy Strategic Systems Programs  
IPEN (Institute for Nuclear and Energy Research), Brazil  
Duke Energy  
Synensis  
Japan MOT Society  
Tufts University  
Southern Company  
U.S. Army Aviation Engineering U.S. Army Corps of Engineers (Kansas City District)  
University of Houston, Clear Lake  
Lincoln Lab  
Hanscom AFB  
U.S. Army Research, Development, and Engineering Command  
McMaster University  
Bechtel  
Kyushu University (Japan)  
Analog Devices  
Cummins  
University of Massachusetts Dartmouth  
Syracuse Safety Research  
National Civil Aviation Agency (ANACO, Brazil)  
State Nuclear Power Automation System  
Engineering Company (China)  
Toyota Central R&D Labs  
Massachusetts General Hospital  
AstraZeneca  
STM (Defense Technology Engineering and Trading Corp., Turkey)  
Varian Medical Systems  
Fort Hill Group  
TUBITAK-UZAY (Scientific and Technological Research Council U.S. Chemical Safety Board of TURKEY-Space Technologies  
Research Institute)  
Cranfield University (U.K.)  
U.S. Air Force Test Pilot School  
NASA/Bastion Technologies  
U.S. Customs and Border Protection  
Second Curve Systems Vequria  
Akamai Technologies  
Canadian Dept. of Defense (DND)  
University of Virginia  
MSAG  
Novartis  
U.S. Coast Guard  
EPRI (Electric Power Research Institute)  
Sandia National Laboratories  
Lawrence Livermore National Laboratories  
Tapestry Solutions  
Kansas State University  
Systems Planning and Analysis  
Zurich University of Applied Sciences  
IBM  
Lawrence Berkeley National Laboratory (LBNL)  
U.S. Navy School of Aviation Safety  
JAMSS (Japanese Manned Space Systems)

**Countries:** USA, Brazil, Japan, China, Netherlands, Germany, Canada, Australia, Iceland, Greece, United Kingdom, Turkey, Estonia, Australia
For more information...

• Email: jthomas4@mit.edu

• Website
  – mit.edu/psas
  – Free annual MIT conference in March
  – Presentations with cross-industry examples available

• Classes
  – Tutorials
  – Training
  – Project-focused workshops

• Radiation therapy application
  – “SYSTEMS THEORETIC HAZARD ANALYSIS (STPA) APPLIED TO THE RISK REVIEW OF COMPLEX SYSTEMS: AN EXAMPLE FROM THE MEDICAL DEVICE INDUSTRY”, Antoine, 2012
  – Includes more examples

© Copyright John Thomas 2016