

# Introduction to system safety and 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?



# Three Mile Island

## 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).
- Operators were not told how the light was designed, only that it indicated whether valve was open.



**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

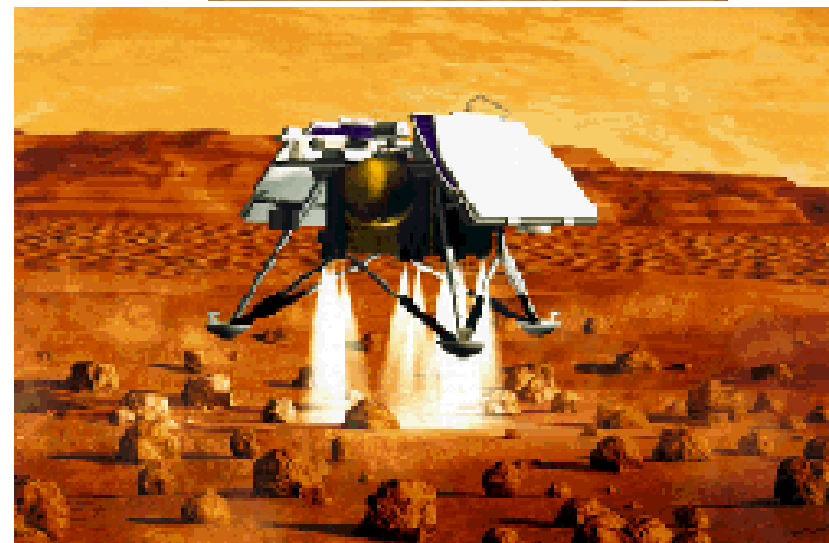
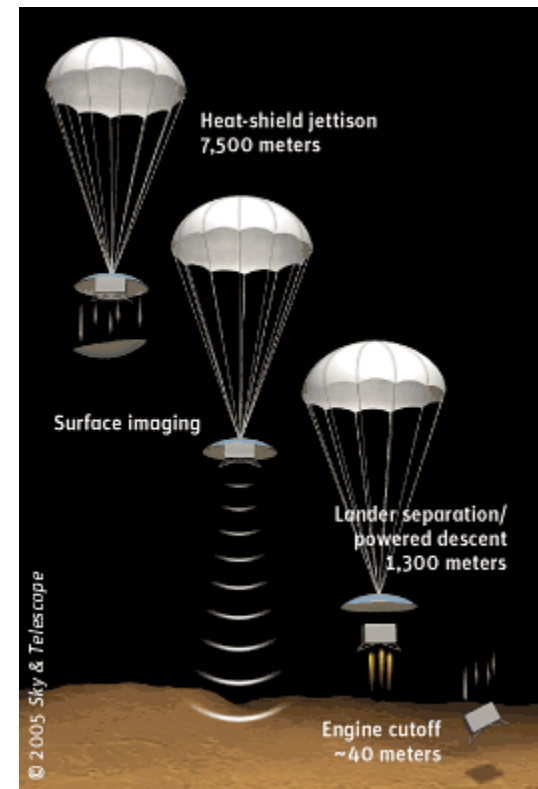
- Need to address component failure accidents
  - Identify important failures, but also go beyond the failures
  - Why weren't the failures detected and mitigated?
  - Human-computer interaction issues?
  - Software-induced operator error?
  - Etc.
- What else is needed?



# 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://www.newsomelaw.com/blog/2012/09/7/ford-announces-third-recall-of-escape-suvs-since-july>

<http://gearheads.org/stop-driving-your-ford-escape/>

# 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>

<http://www.thetruthaboutcars.com/2012/07/fire-escape-its-the-suppliers-fault/>



# 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://corporate.ford.com/news-center/press-releases-detail/pr-ford-produces-fix-in-voluntary-37491>

<http://www.usatoday.com/story/money/cars/2012/12/10/ford-recall-escape-fusion-ecoboost/1759063/>

# Quote

- “The hardest single part of building a software system is deciding precisely what to build.”  
-- Fred Brooks, *The Mythical Man-Month*

# Quote

- “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!**

# Toyota

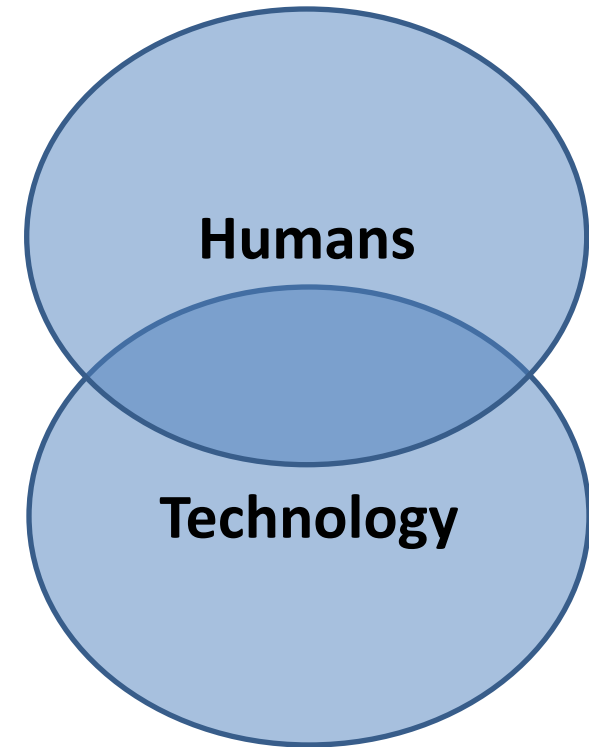
- **2004:** Push-button ignition
- **2004-2009**
  - 102 incidents of uncontrolled acceleration
  - Speeds exceed 100 mph despite stomping on the brake
  - 30 crashes
  - 20 injuries
- **2009, Aug:**
  - Car accelerates to 120 mph
  - Passenger calls 911, reports stuck accelerator
  - Car crashes killing 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**

# 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

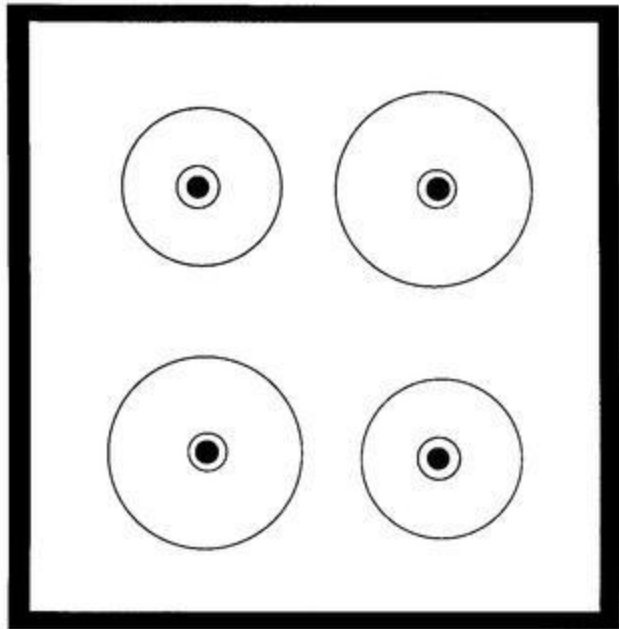
# Human Factors: **Systems View**

---

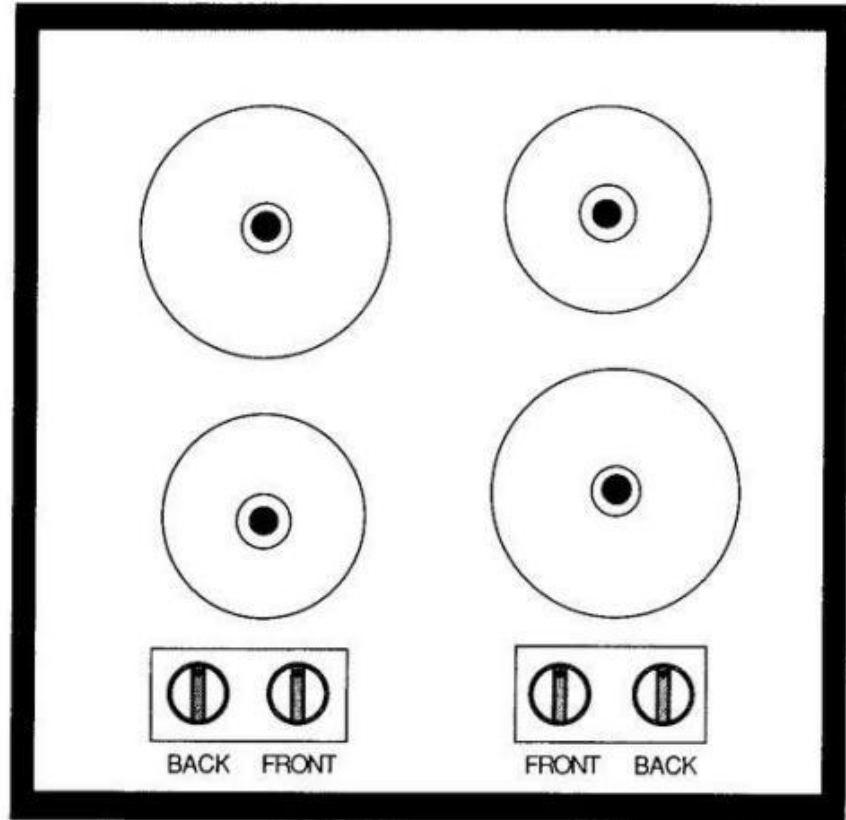
(Dekker, Rasmussen, Leveson, Woods, etc.)

- 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

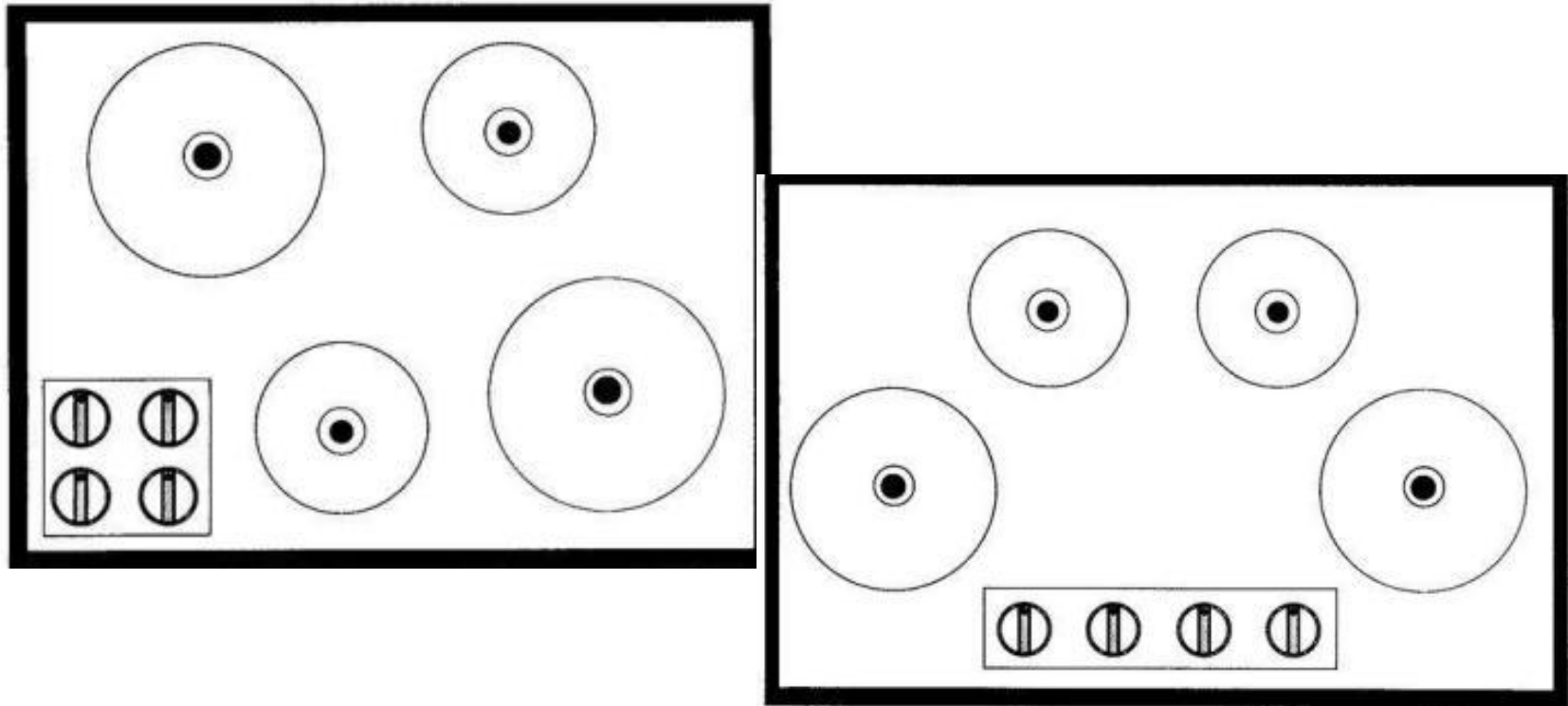


Back Right   Front Left   Back Left   Front Right



**Human error?**

# Natural Mapping

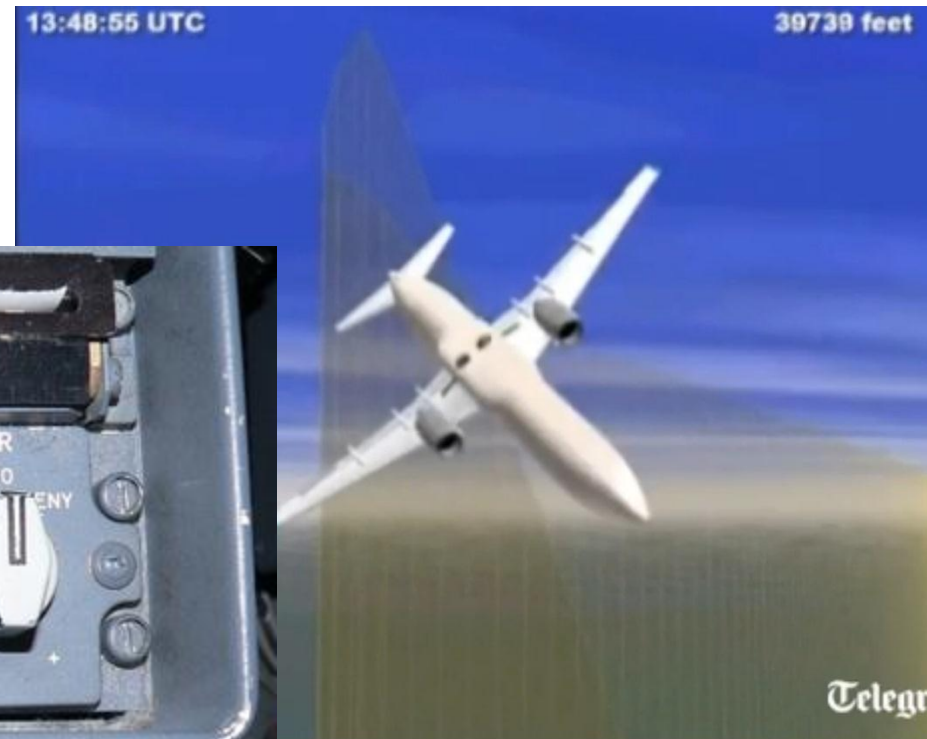


**Human error? Or design problem?**



# ANA B737, Sept 2011

- Drops 2000 feet in a 30-second fall, exceeded designed mach and G forces
- Deeper problems?



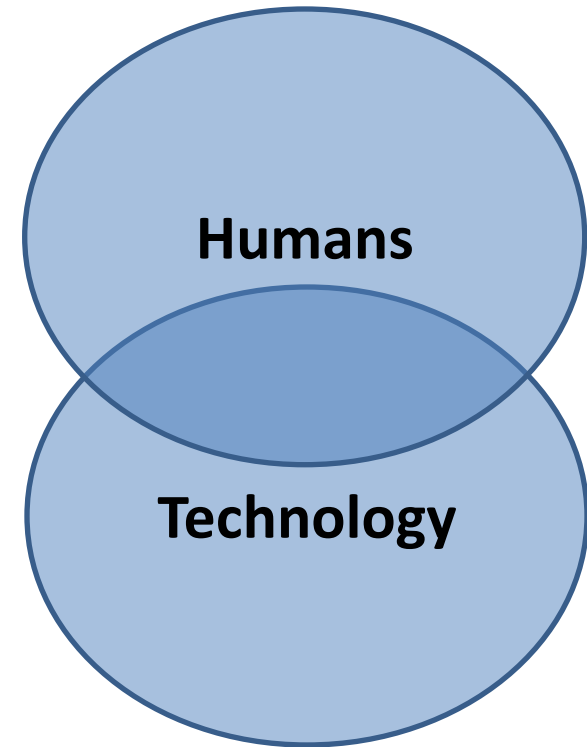
# China Airlines 006

- 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?

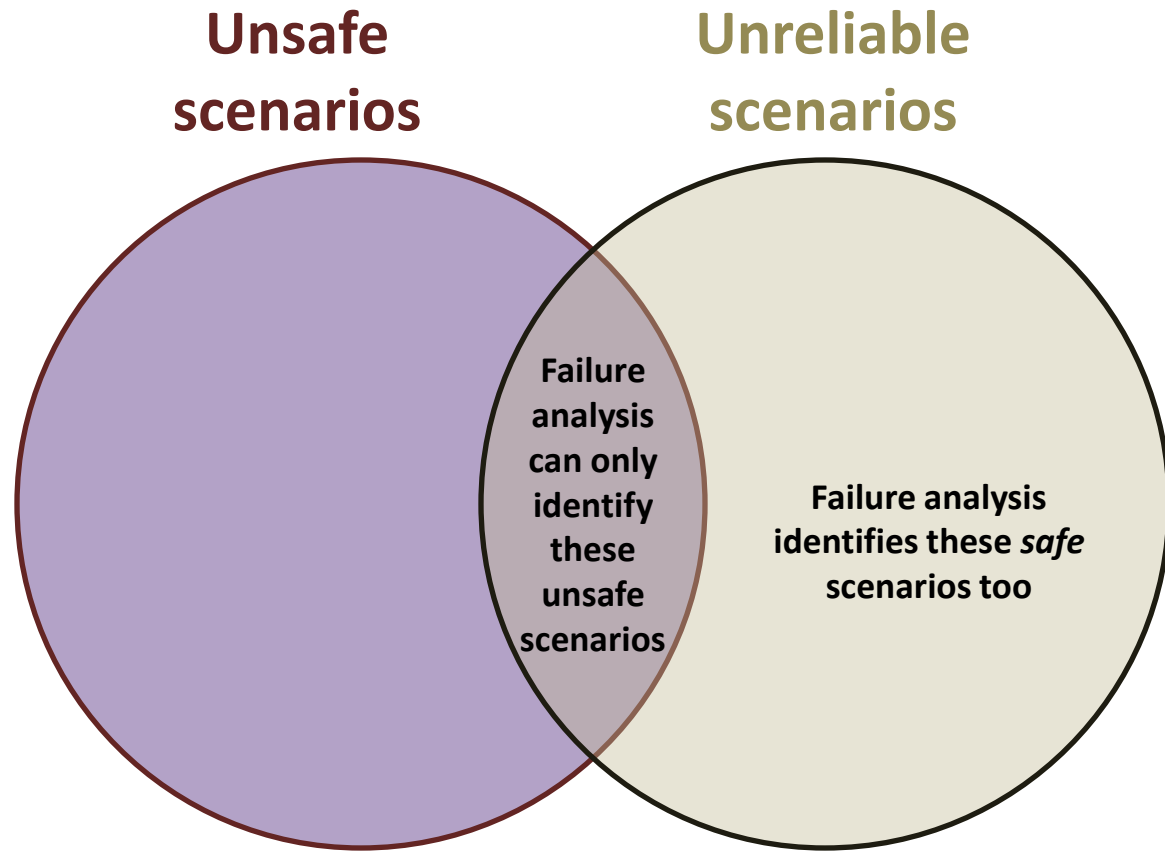


# Safety vs. reliability

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

# Safe ≠ Reliable

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

	Safe	Unsafe
Reliable	•Typical commercial flight	
Unreliable		•Aircraft engine fails in flight

# Safe ≠ Reliable

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

	Safe	Unsafe
Reliable	<ul style="list-style-type: none"><li>• Typical commercial flight</li></ul>	<ul style="list-style-type: none"><li>• Computer reliably executes unsafe commands</li><li>• Increasing tank burst pressure</li><li>• Retreating to safe state vs. achieving mission</li><li>• A nail gun without safety lockout</li></ul>
Unreliable		<ul style="list-style-type: none"><li>• Aircraft engine fails in flight</li></ul>

# Safe ≠ Reliable

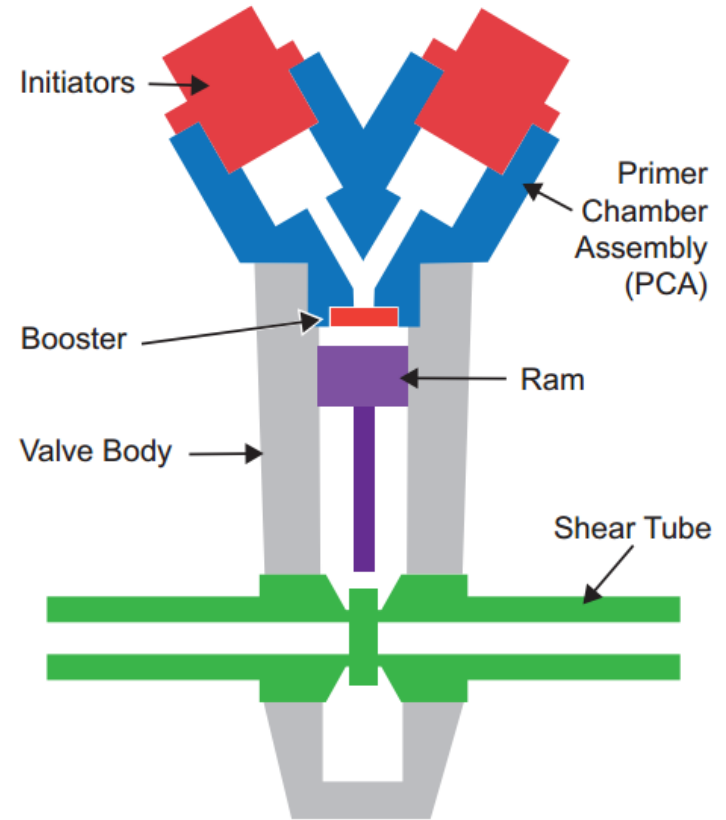
- Safety often means making sure X never happens
- Reliability usually means making sure Y always happens

	Safe	Unsafe
Reliable	<ul style="list-style-type: none"><li>• Typical commercial flight</li></ul>	<ul style="list-style-type: none"><li>• Computer reliably executes unsafe commands</li><li>• Increasing tank burst pressure</li><li>• Retreating to safe state vs. achieving mission</li><li>• A nail gun without safety lockout</li></ul>
Unreliable	<ul style="list-style-type: none"><li>• Aircraft engine won't start on ground?</li><li>• Missile won't fire?</li></ul>	<ul style="list-style-type: none"><li>• Aircraft engine fails in flight</li></ul>



# 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

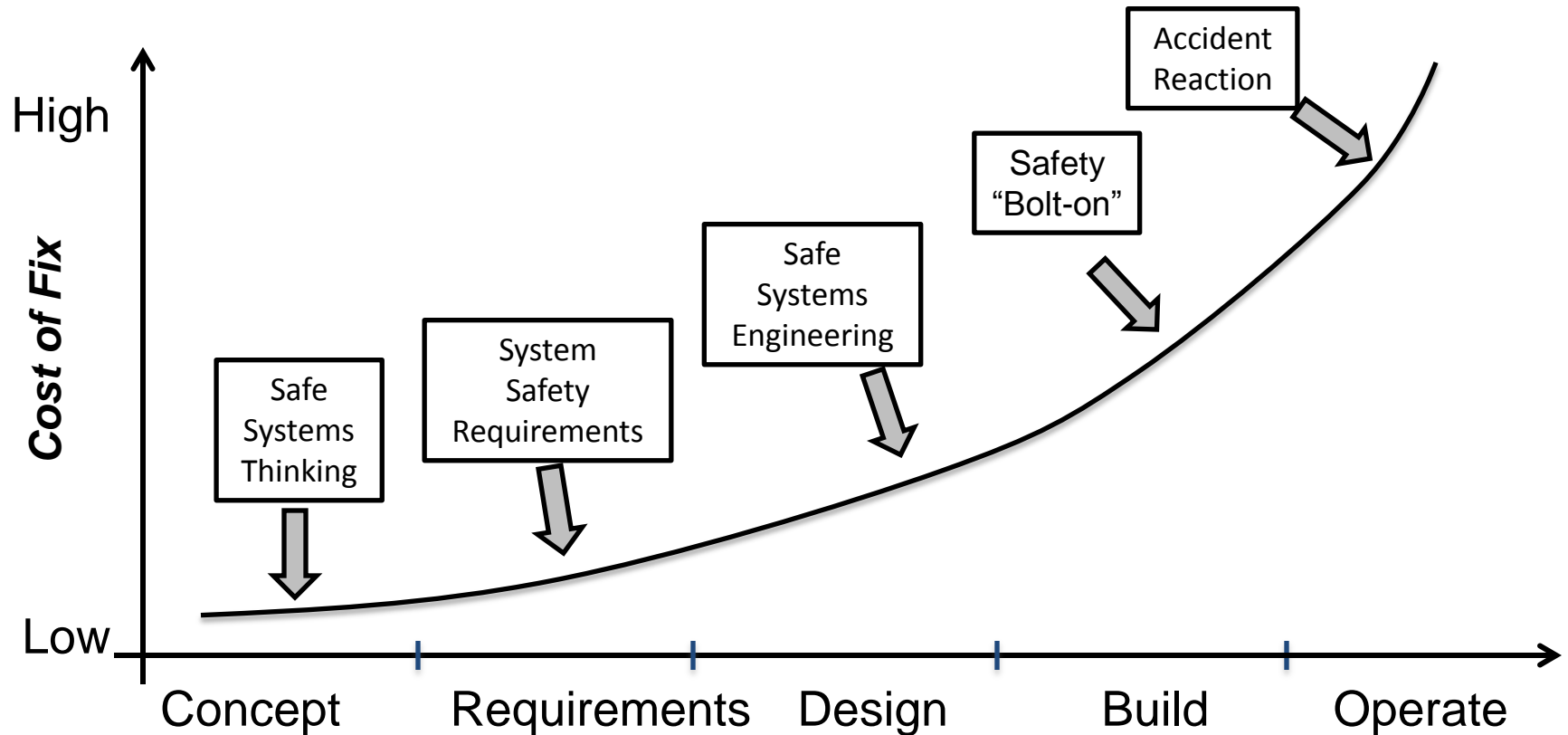


A normally closed pyrovalve

# 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?

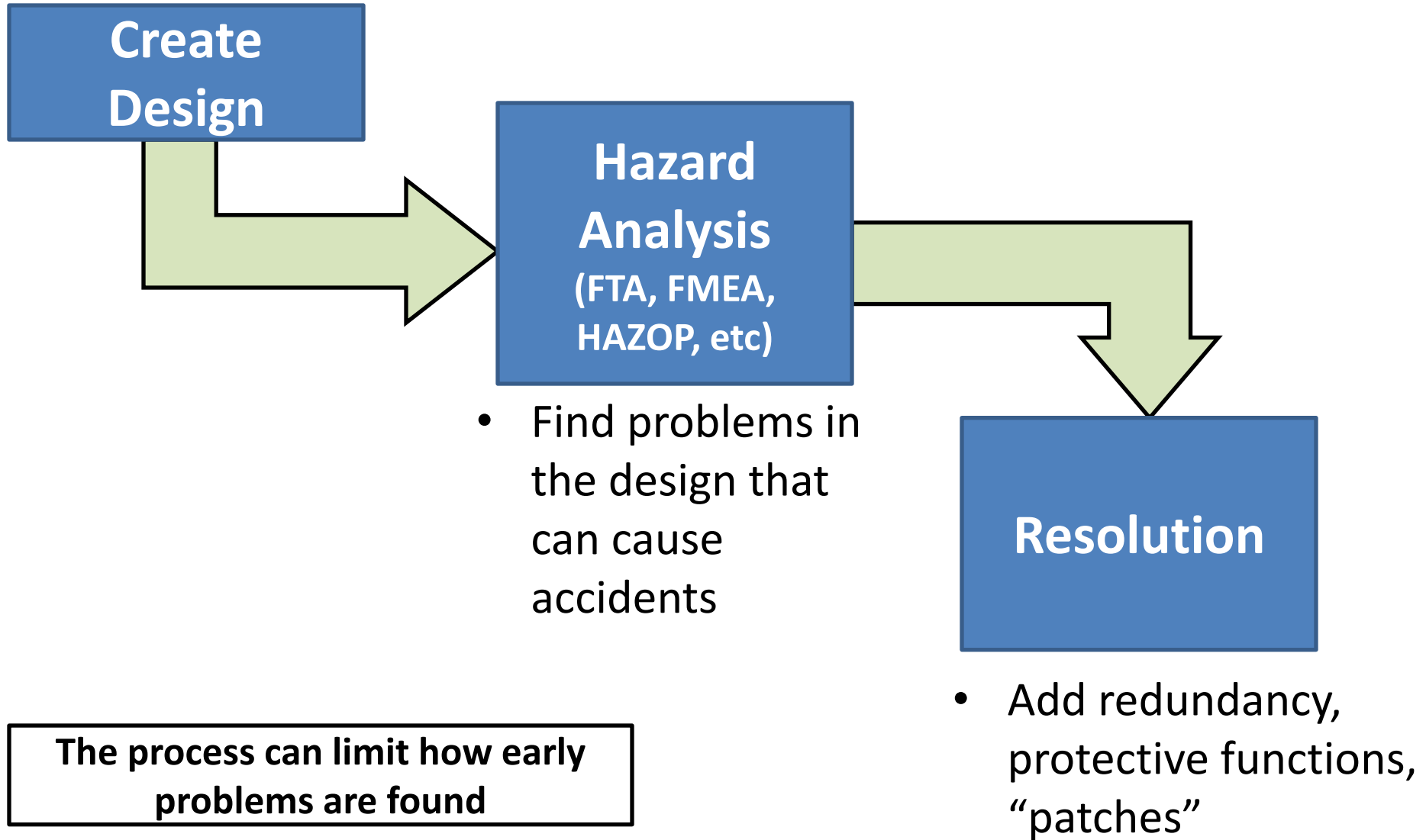
# Building Safety into the System



**Need to address safety early**

**Early decisions tend to have biggest impact on safety**

# Traditional Safety Engineering



# 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
- What else?

# Boeing 787 Lithium Battery Fires

- 2013 – 2014
- Two fires caused by battery failures in 52,000 flight hours
  - Vs. 10 million flight hours predicted by the extensive reliability analysis for certification
- Does not include 3 other less-reported incidents of smoke in battery compartment



**Another simple component failure accident?**

# Boeing 787 Lithium Battery Fires

- A module monitors for smoke in the battery bay, controls fans and ducts to exhaust smoke overboard.
- Power unit experienced low battery voltage, shut down various electronics including ventilation.
- Smoke could not be redirected outside cabin



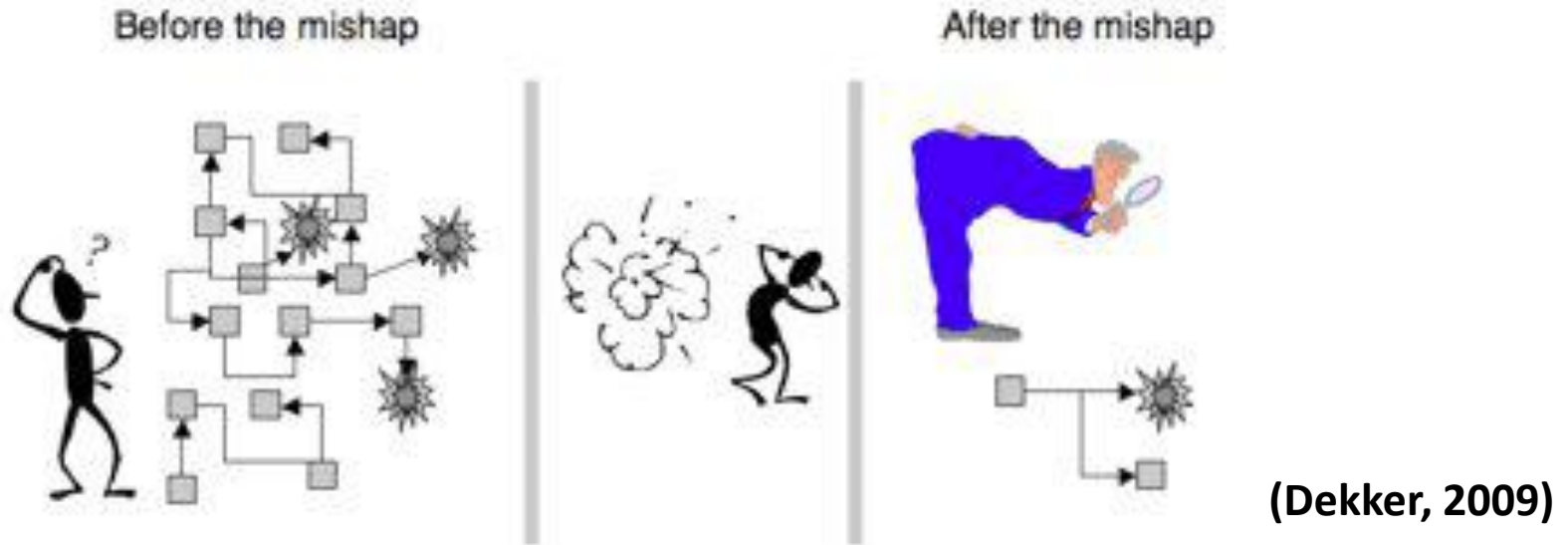
**All requirements were satisfied!  
The requirements were inadequate**

# 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



# Hindsight bias



- After an accident, hindsight can make causes seem obvious
- But during engineering there are 1000s of variables and potential problems to consider

# Safety vs. Reliability: another difference

Using standard engineering techniques of

- Preventing failures through redundancy
- Increasing component 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?

- Lesson from 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

# 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

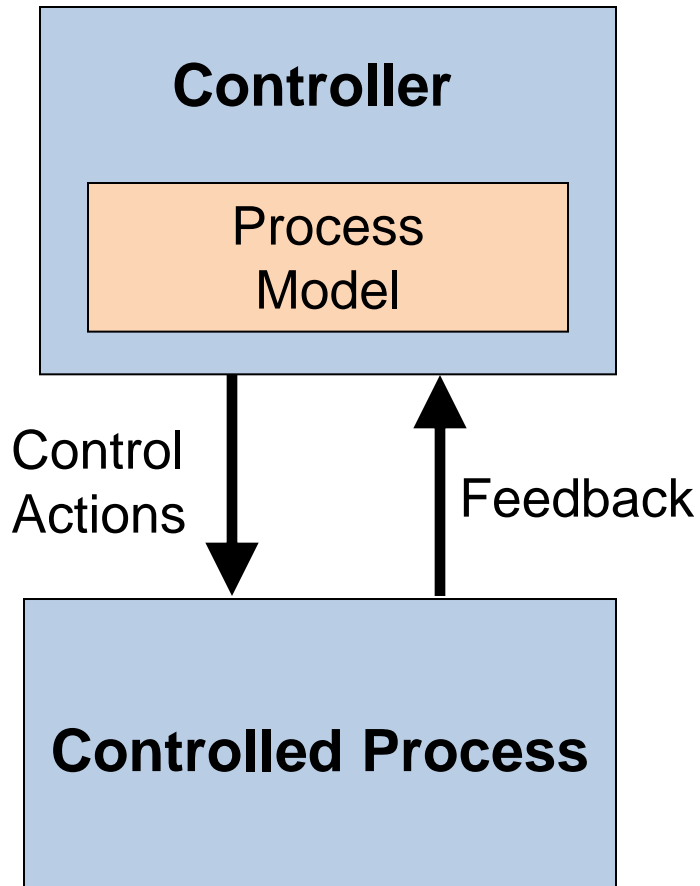
# Systems approach to safety engineering (STAMP)



## STAMP Model

- 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

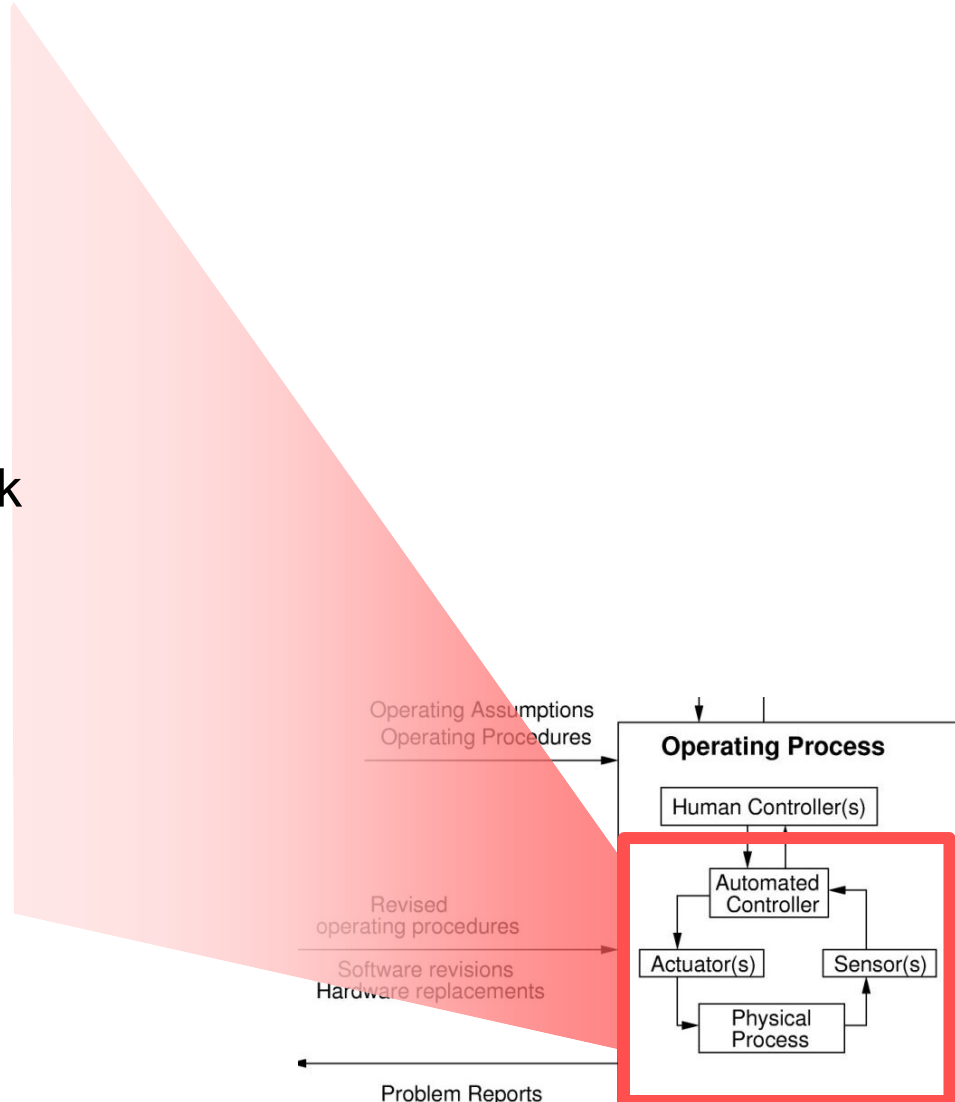
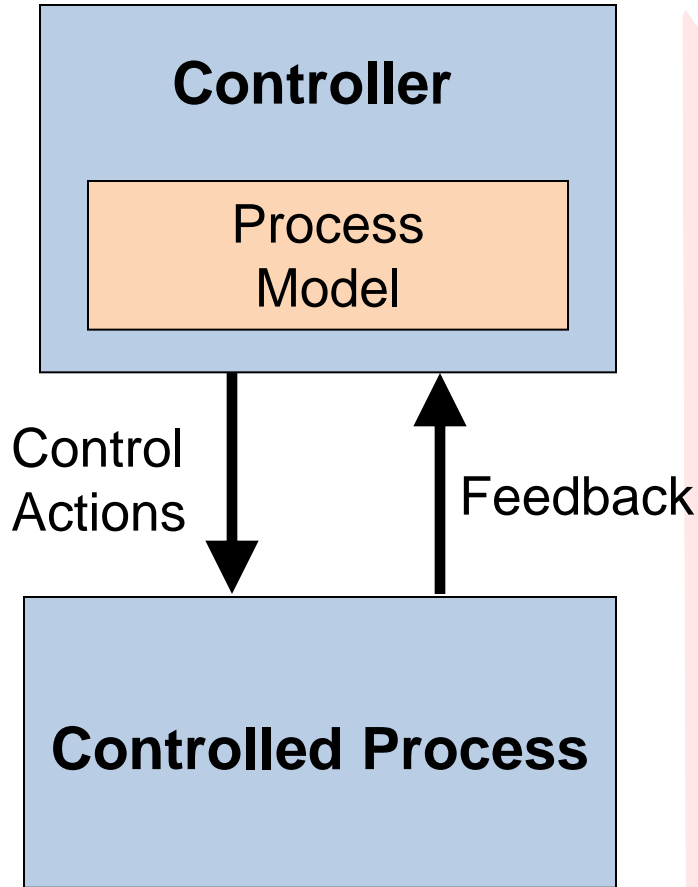
# STAMP



- 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,<sup>51</sup>...**

# STAMP



Operating Assumptions  
Operating Procedures

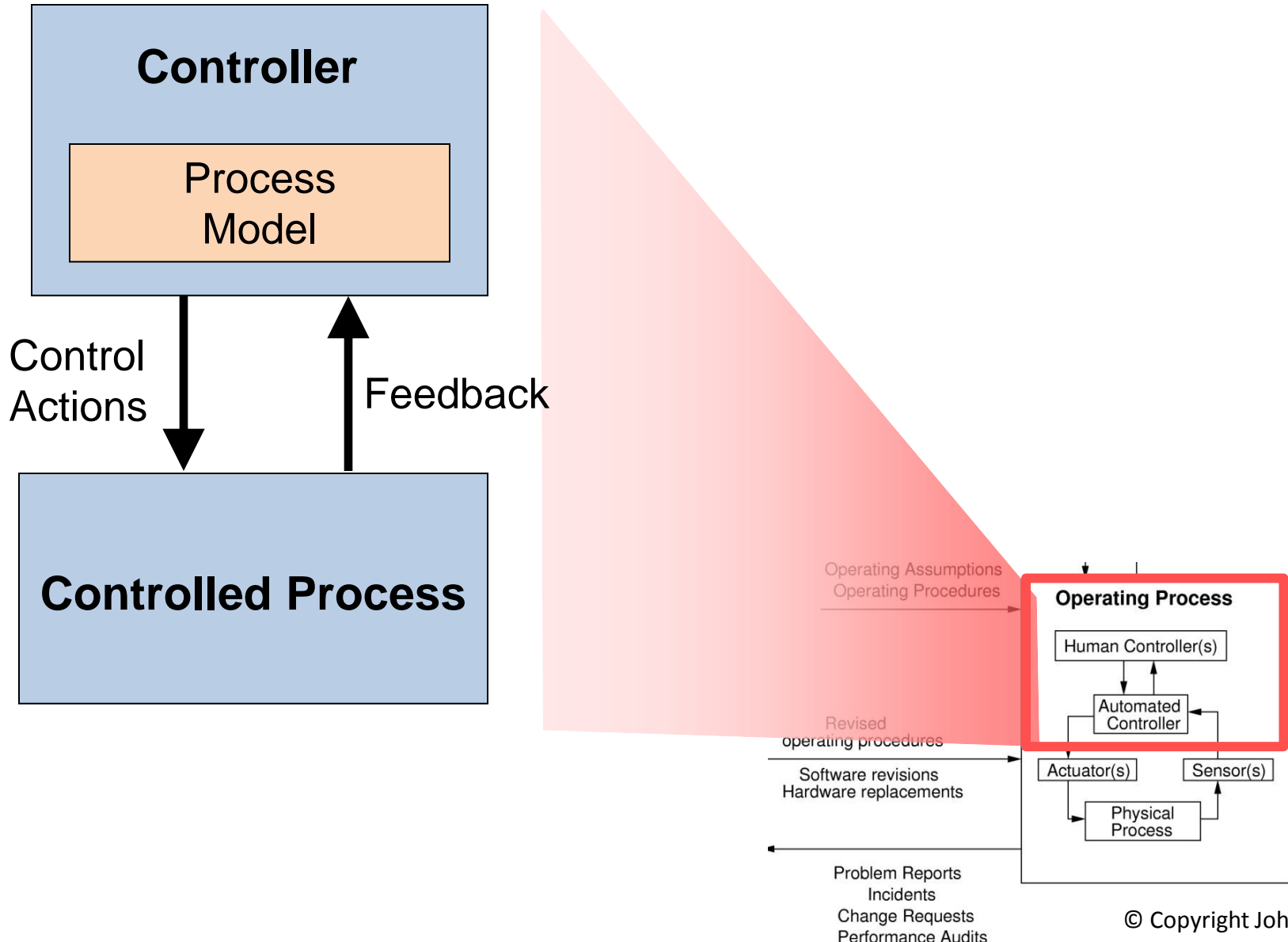
Revised  
operating procedures

Software revisions  
Hardware replacements

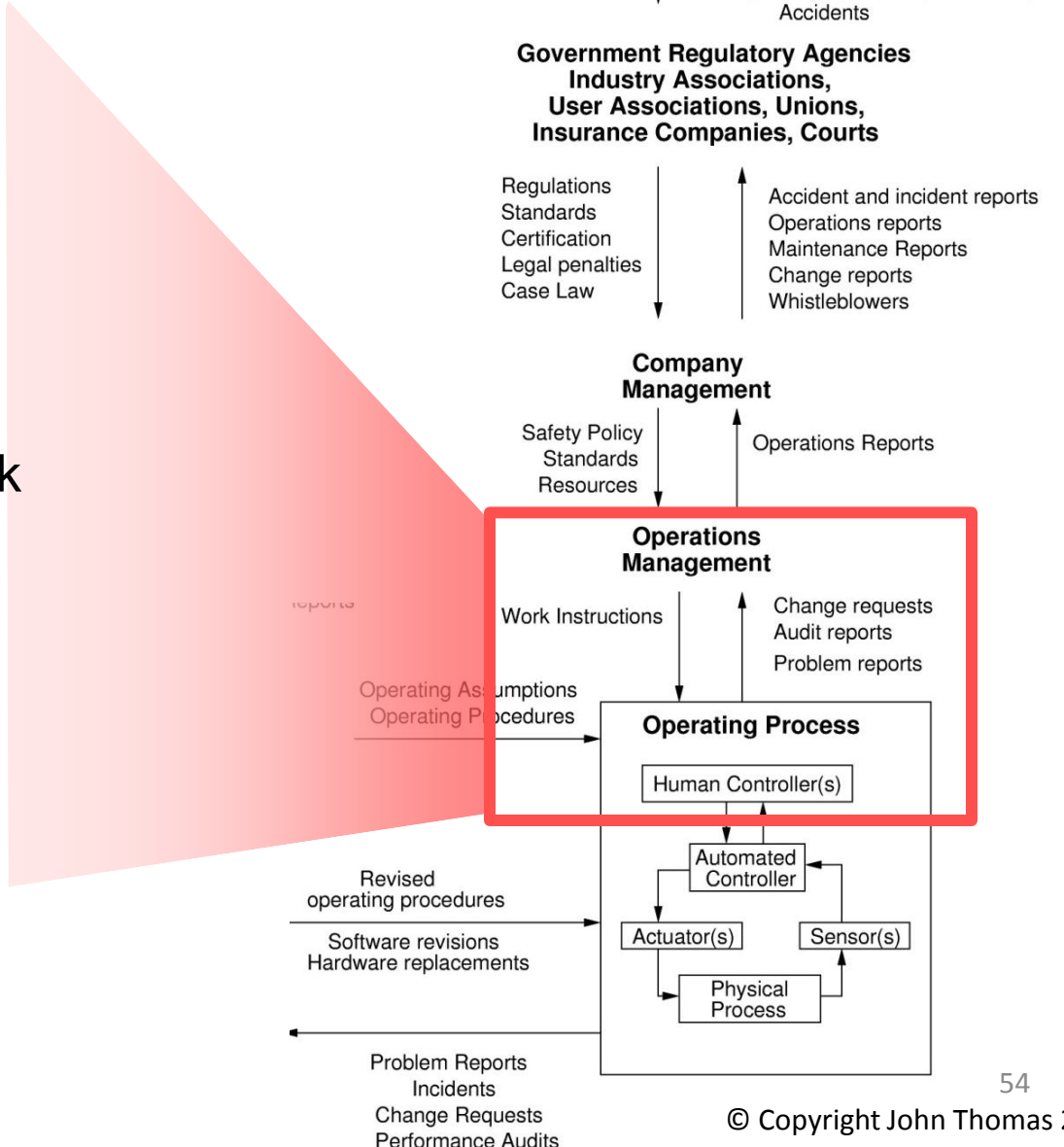
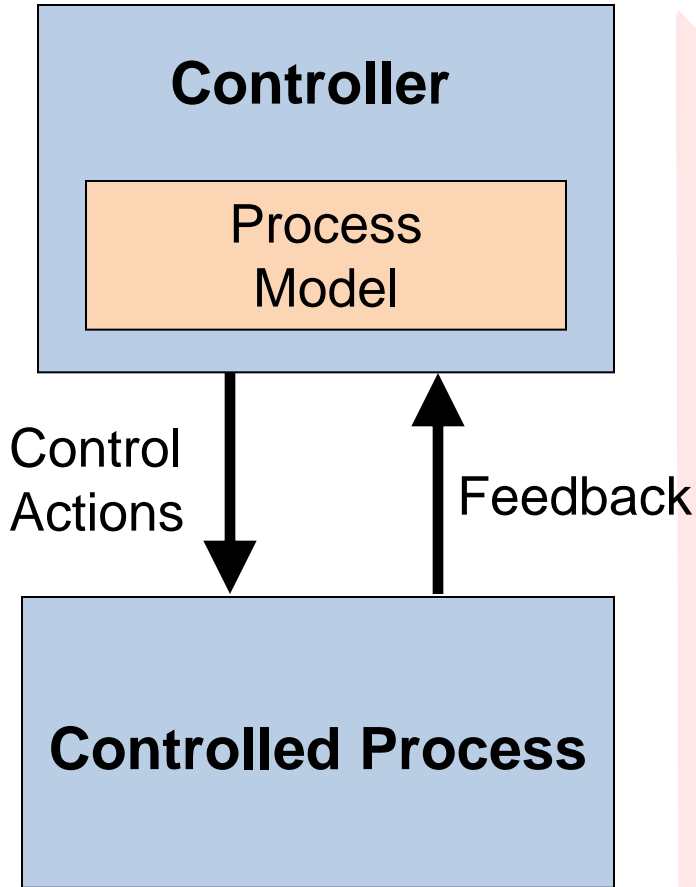
Problem Reports  
Incidents  
Change Requests  
Performance Audits



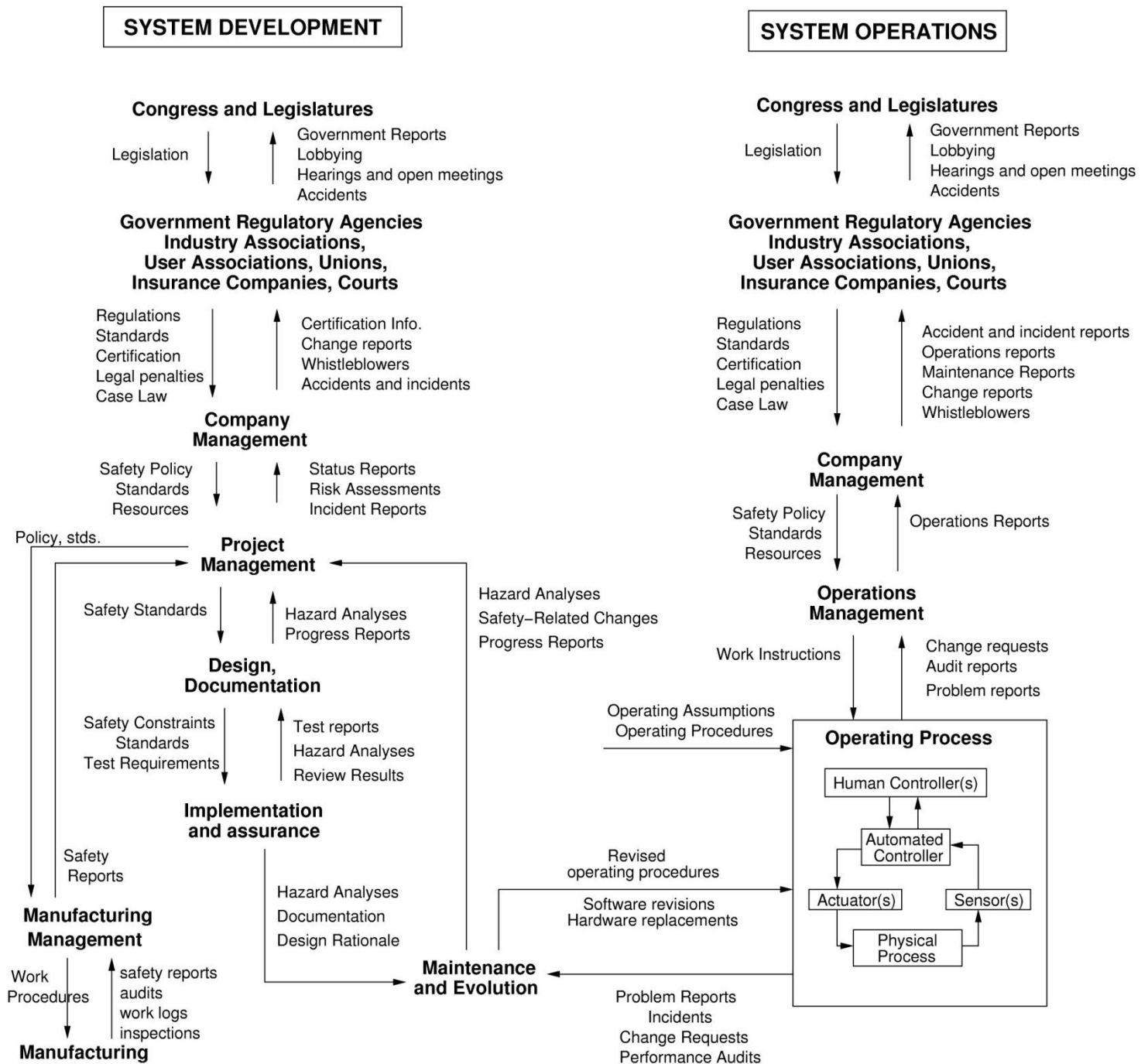
# STAMP



# STAMP




# Example Safety Control Structure



# STAMP and STPA



**STAMP Model**



Accidents are  
caused by  
inadequate control

# STAMP and STPA



STPA  
Hazard Analysis

The diagram consists of two stacked rectangular boxes. The top box is orange and contains the text 'STPA Hazard Analysis'. The bottom box is purple and contains the text 'STAMP Model'. To the right of the orange box is an orange curly bracket pointing to the text 'How do we find inadequate control in a design?'. To the right of the purple box is a purple curly bracket pointing to the text 'Accidents are caused by inadequate control'.

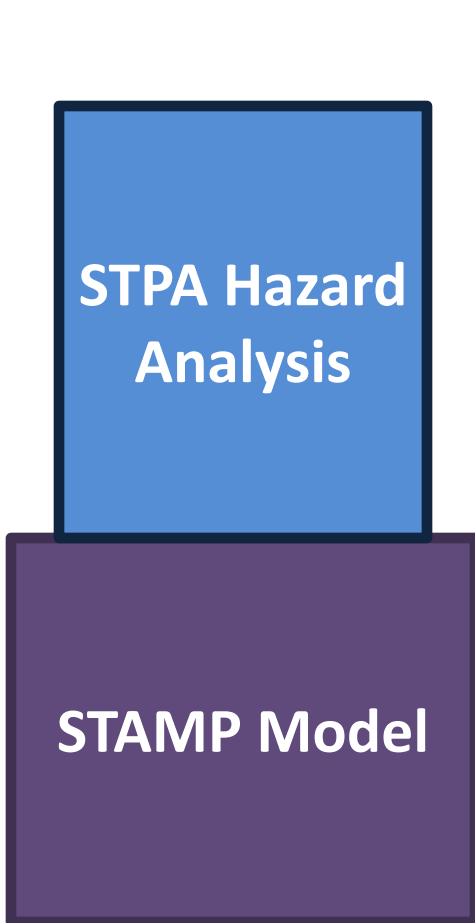
STAMP Model

How do we find  
inadequate control  
in a design?

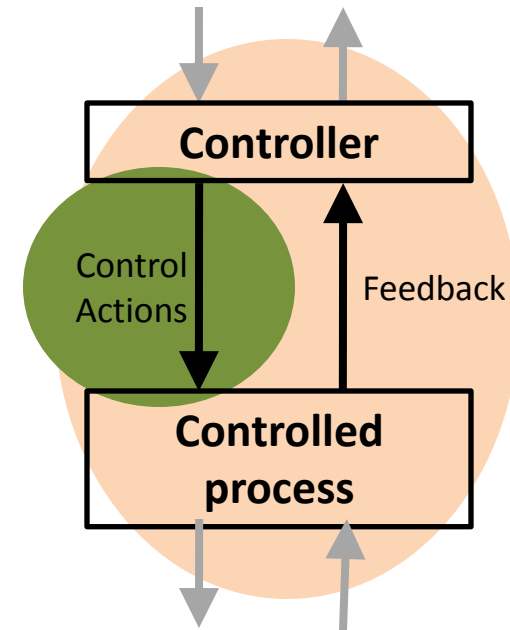
Accidents are  
caused by  
inadequate control

# 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



# 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).

# 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

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



# 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.

## Broad view of safety

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

People die from radiation sickness	Nuclear power plant radioactive materials are not contained
Vehicle collides with another vehicle	Vehicles do not maintain safe distance from each other
People die from food poisoning	Food products for sale contain pathogens

# System Safety Constraints

## System Hazard

---

## System Safety Constraint

---

Toxic chemicals from the plant are in the atmosphere



Toxic plant chemicals must not be released into the atmosphere

---

Nuclear power plant radioactive materials are not contained



Radioactive materials must not be released

---

Vehicles do not maintain safe distance from each other



Vehicles must always maintain safe distances from each other

---

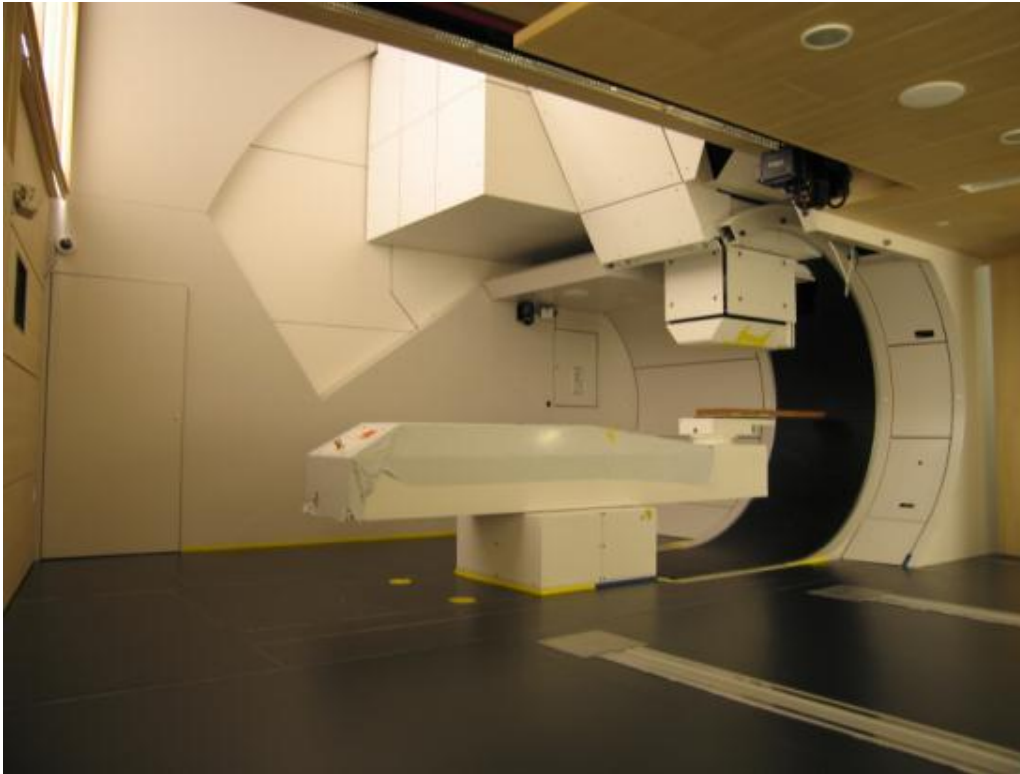
Food products for sale contain pathogens



Food products with pathogens must not be sold

---

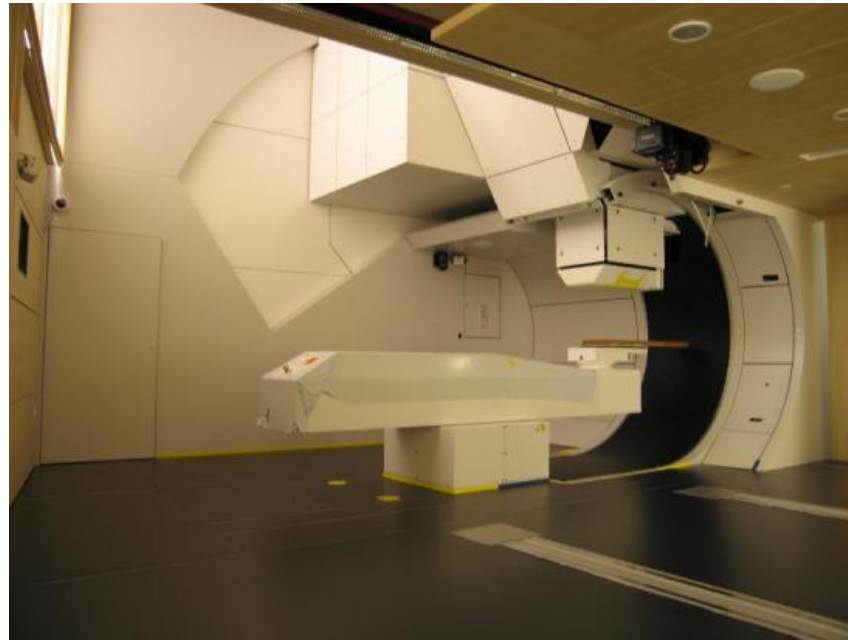
# Proton Radiation Therapy System Paul Scherrer Institute, Switzerland



- 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 Structure

# Chemical Plant



# Chemical Plant

Citicchem Safety Control Structure

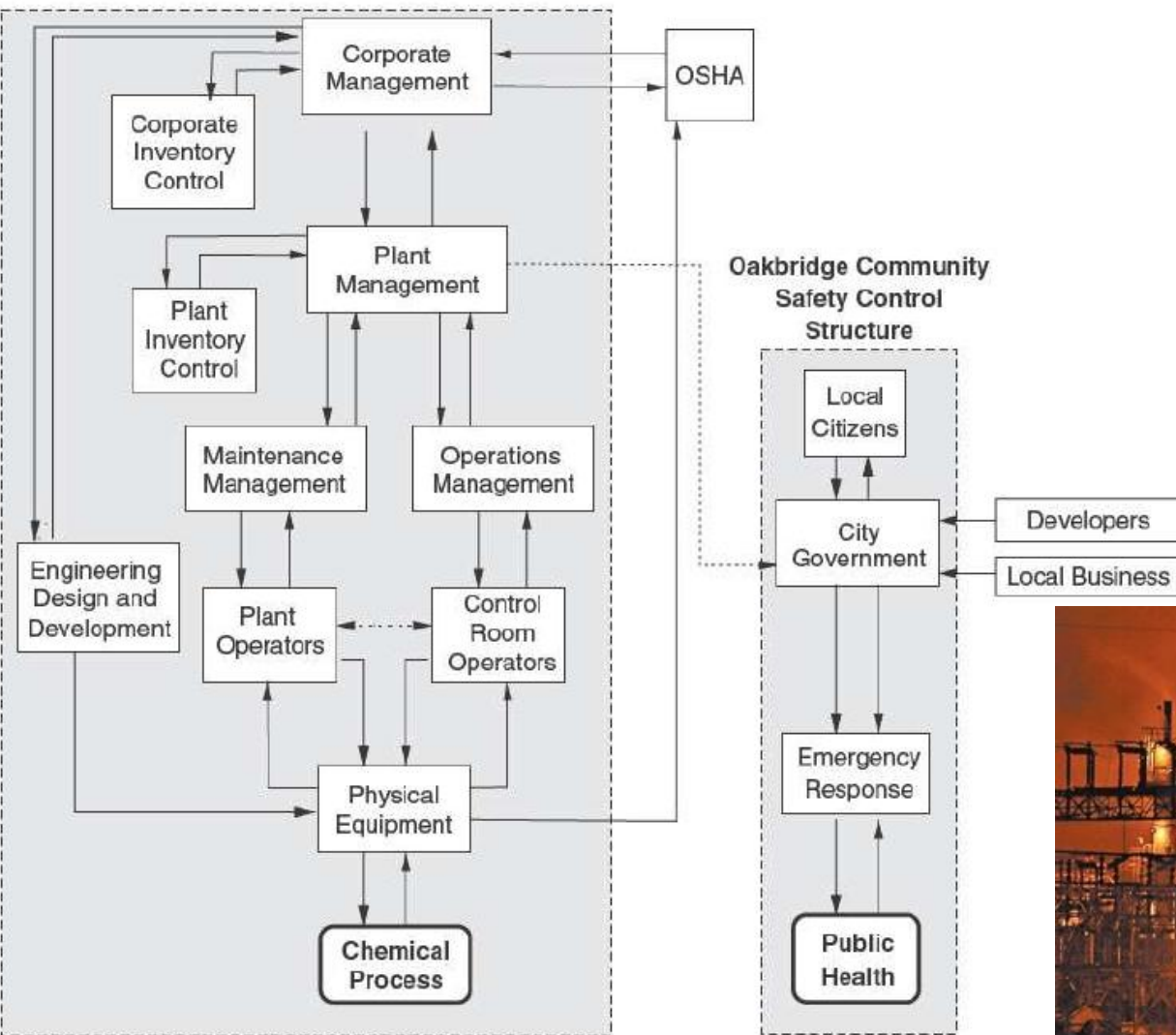
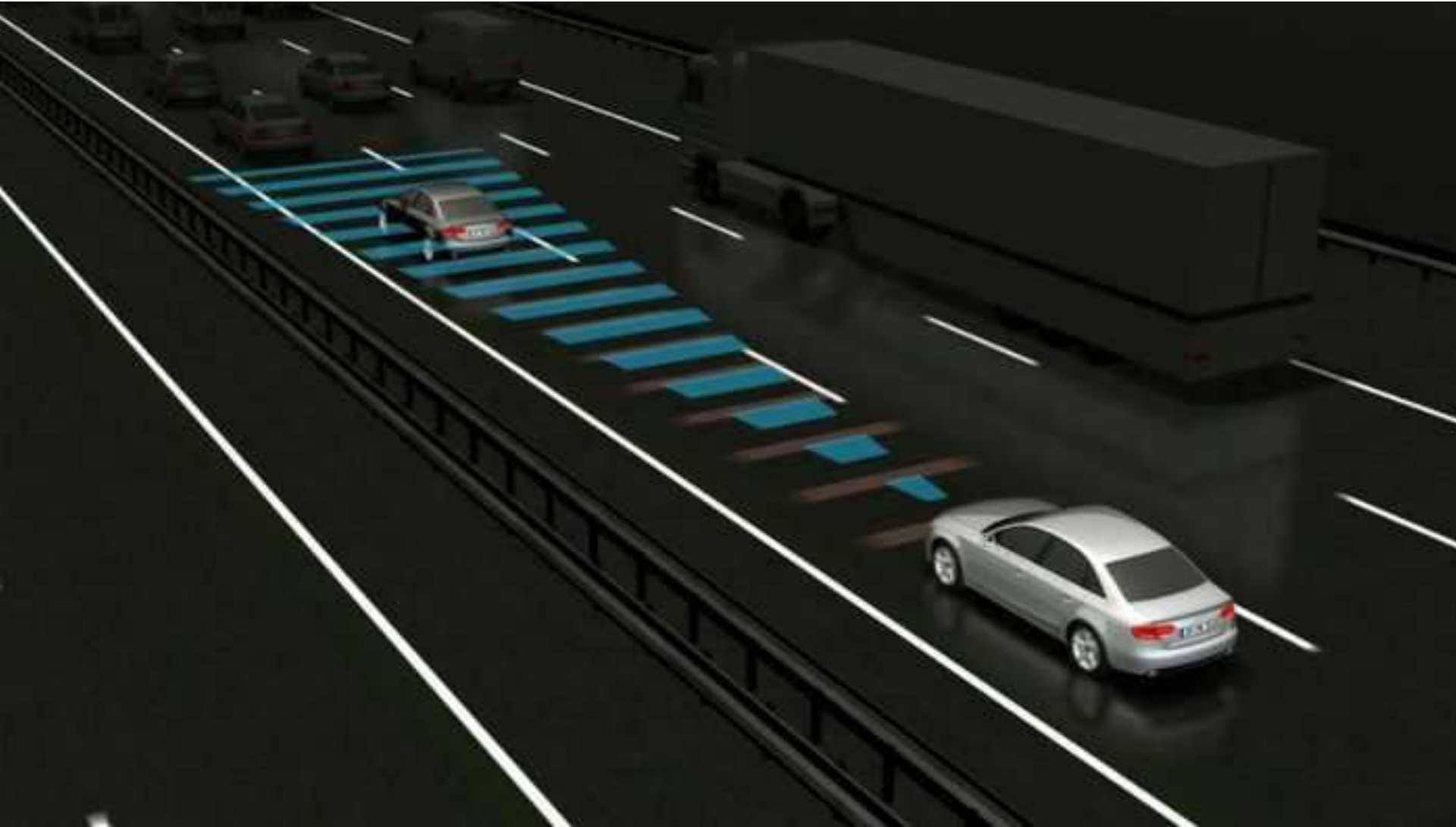


Image from:  
<http://www.cbgnetwork.org/2608.html>

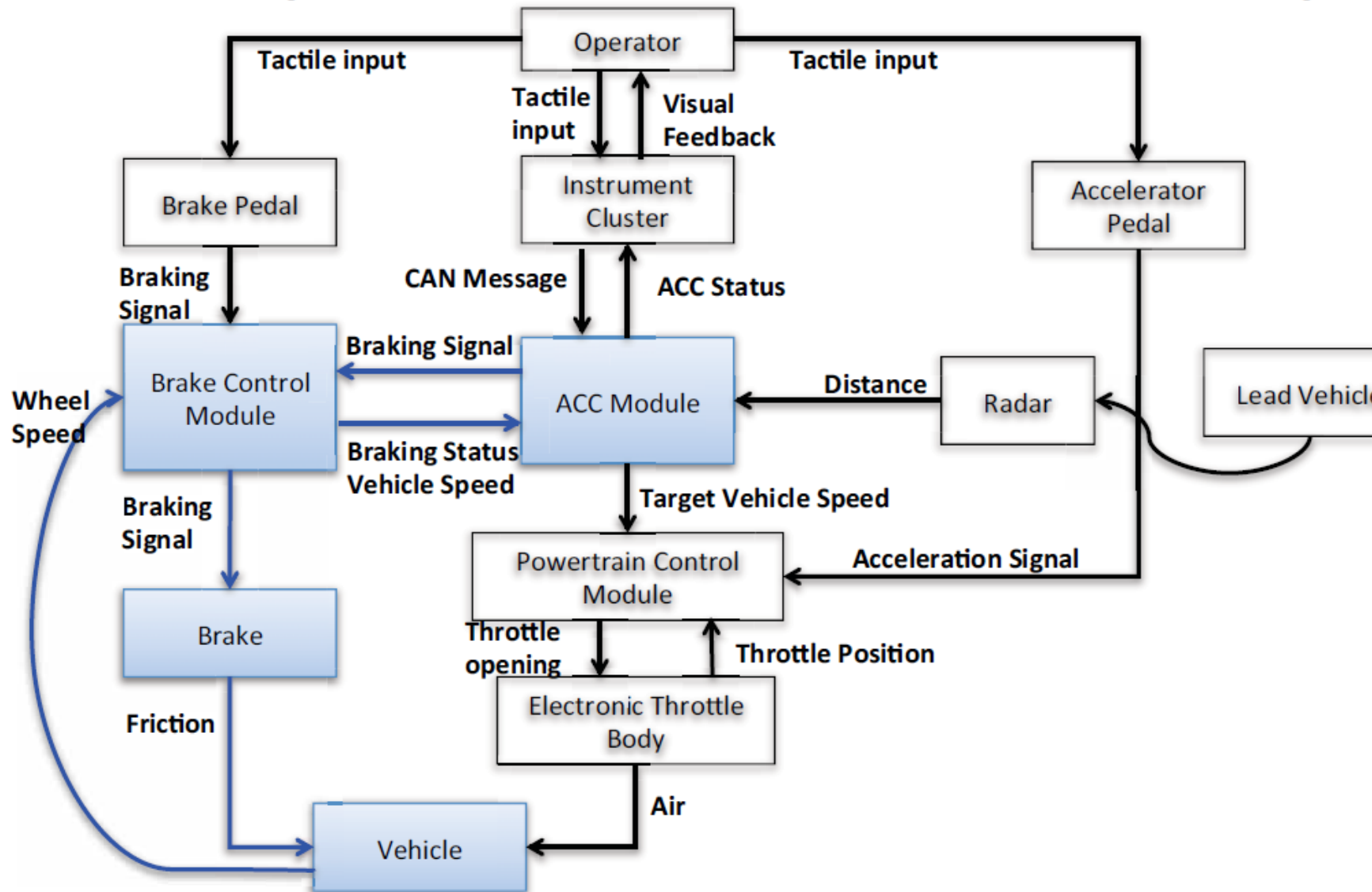


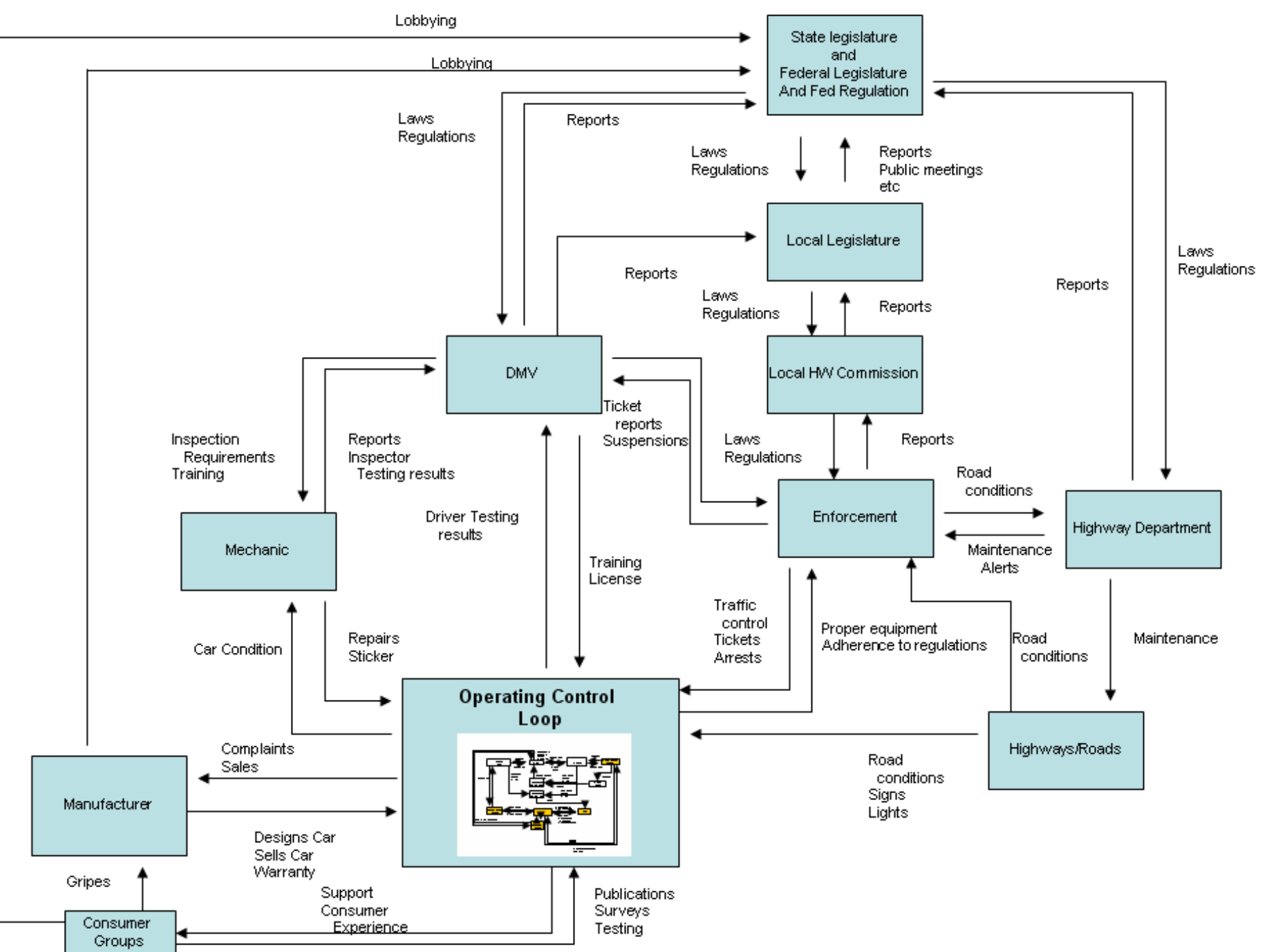


# Adaptive Cruise Control



# Example: ACC – BCM Control Loop





# Ballistic Missile Defense System

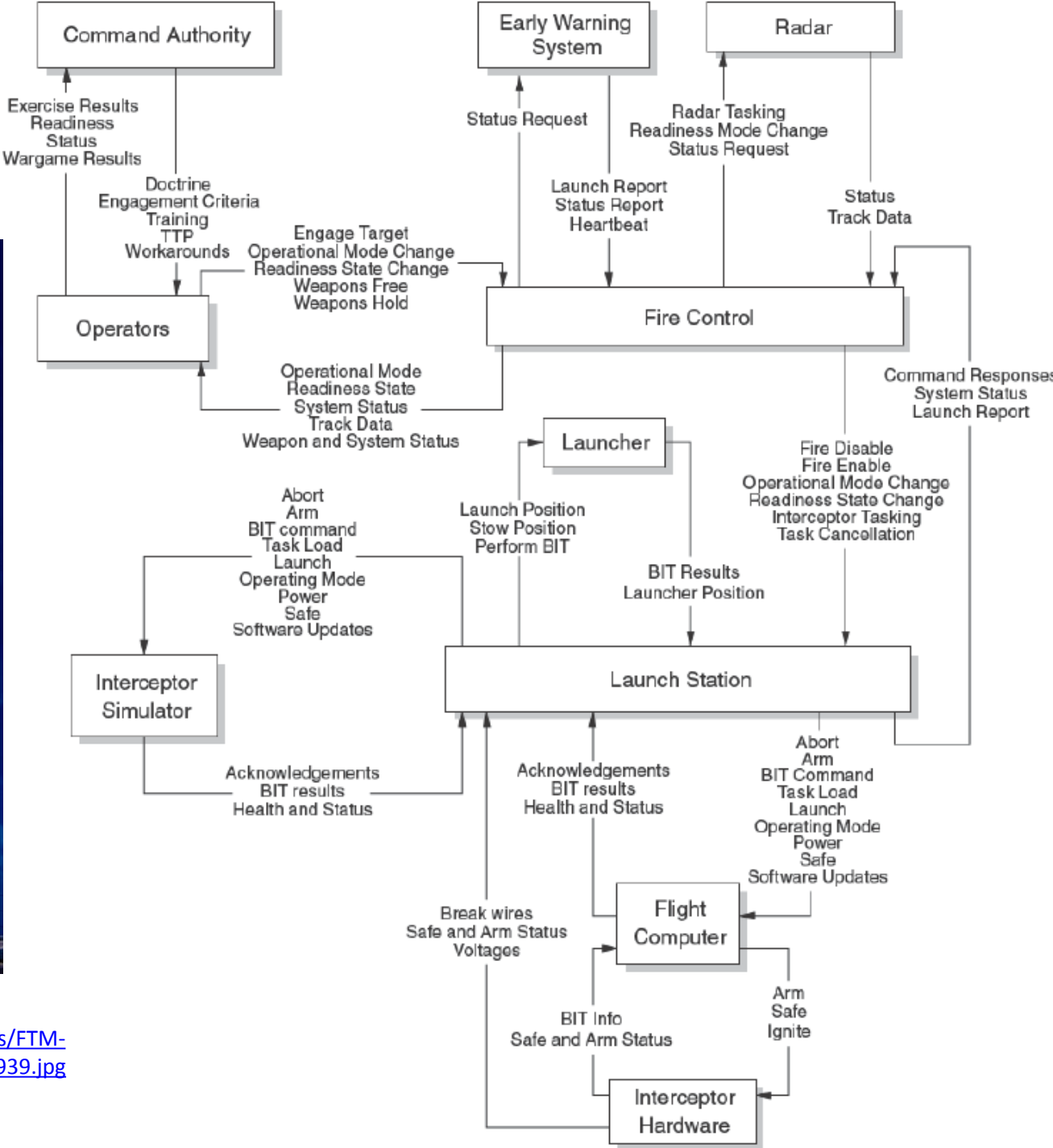


Image from:  
[http://www.mda.mil/global/images/system/aegis/FTM-21\\_Missile%20Bulkhead%20Center14\\_BN4H0939.jpg](http://www.mda.mil/global/images/system/aegis/FTM-21_Missile%20Bulkhead%20Center14_BN4H0939.jpg)

# Congress U.S. pharmaceutical safety control structure

FDA



Pharmaceutical  
Companies

Doctors

Patients

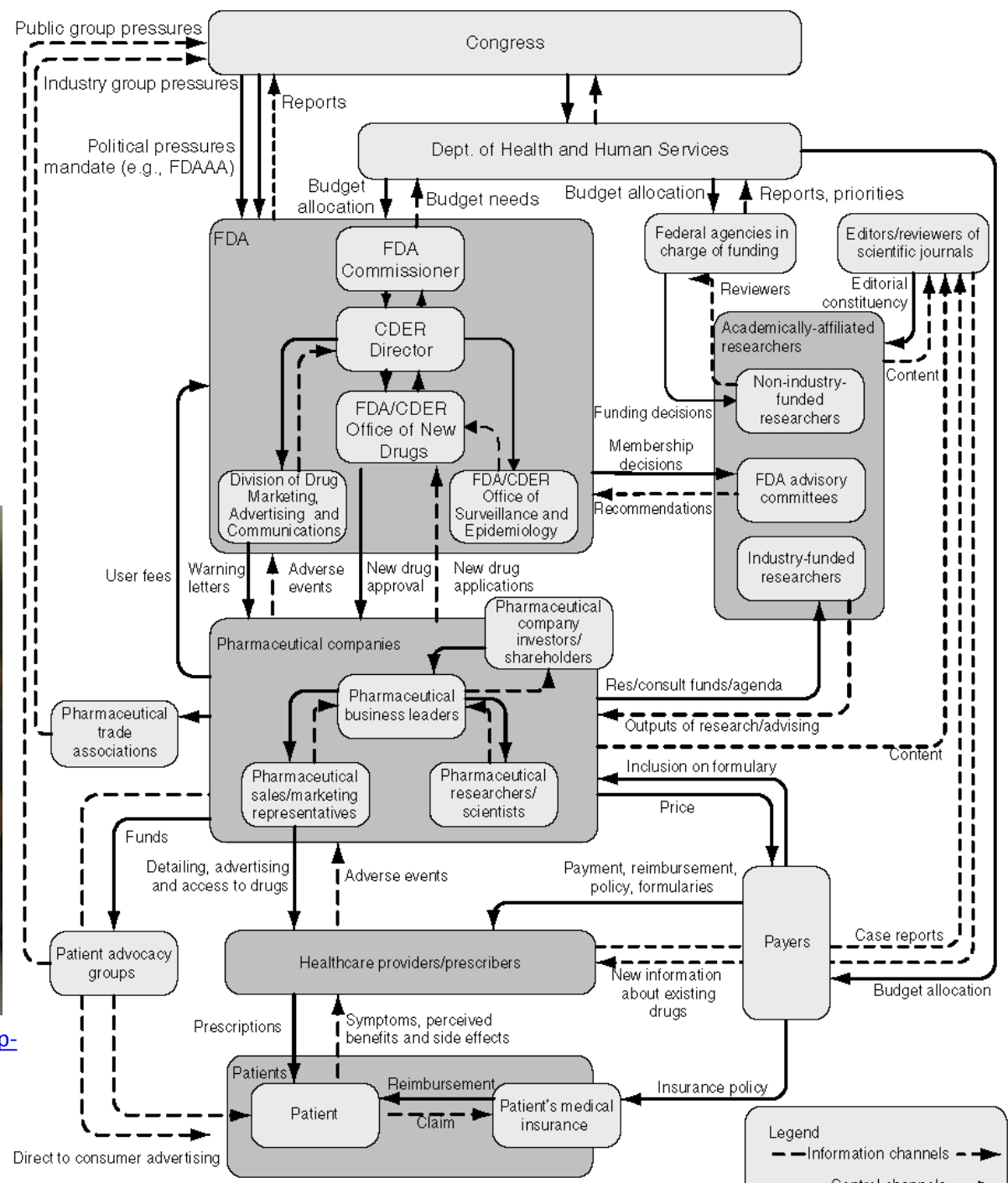
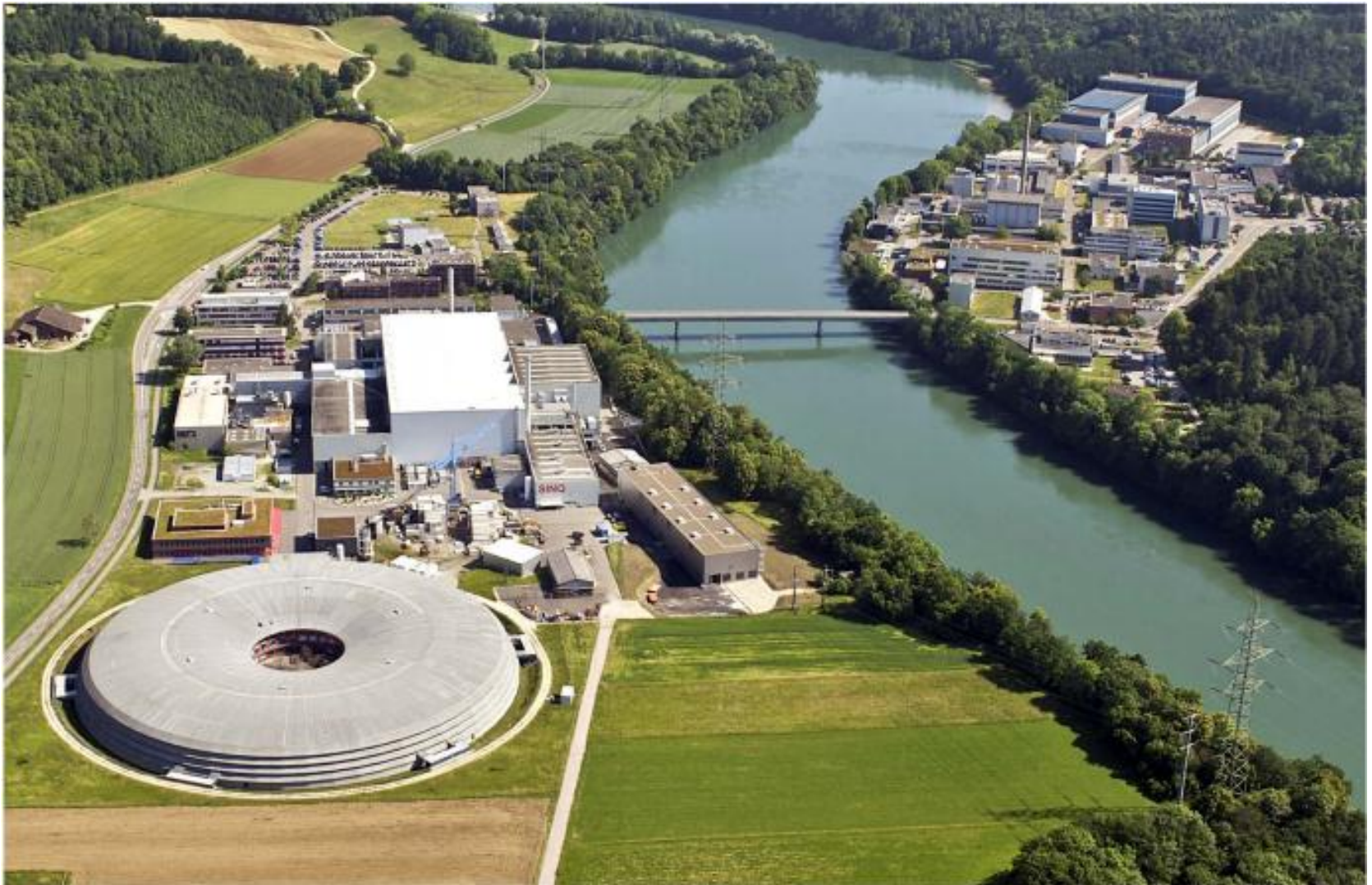


Image from: <http://www.kleantreatmentcenter.com/wp-content/uploads/2012/07/vioxx-pills>

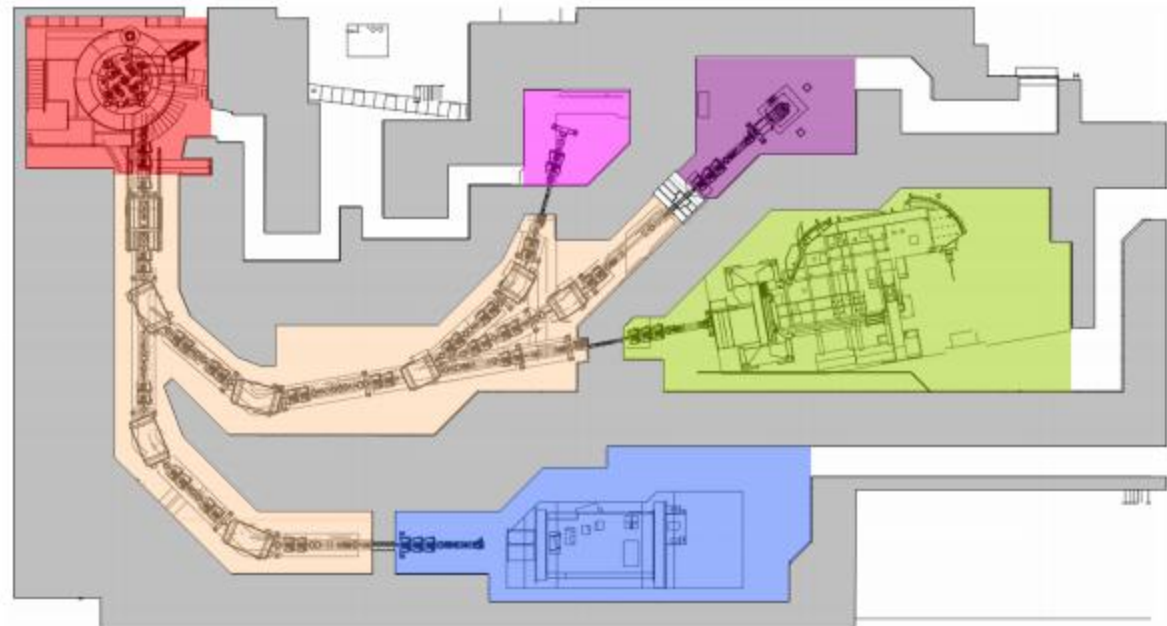


# Proton Radiation Therapy System Paul Scherrer Institute, Switzerland



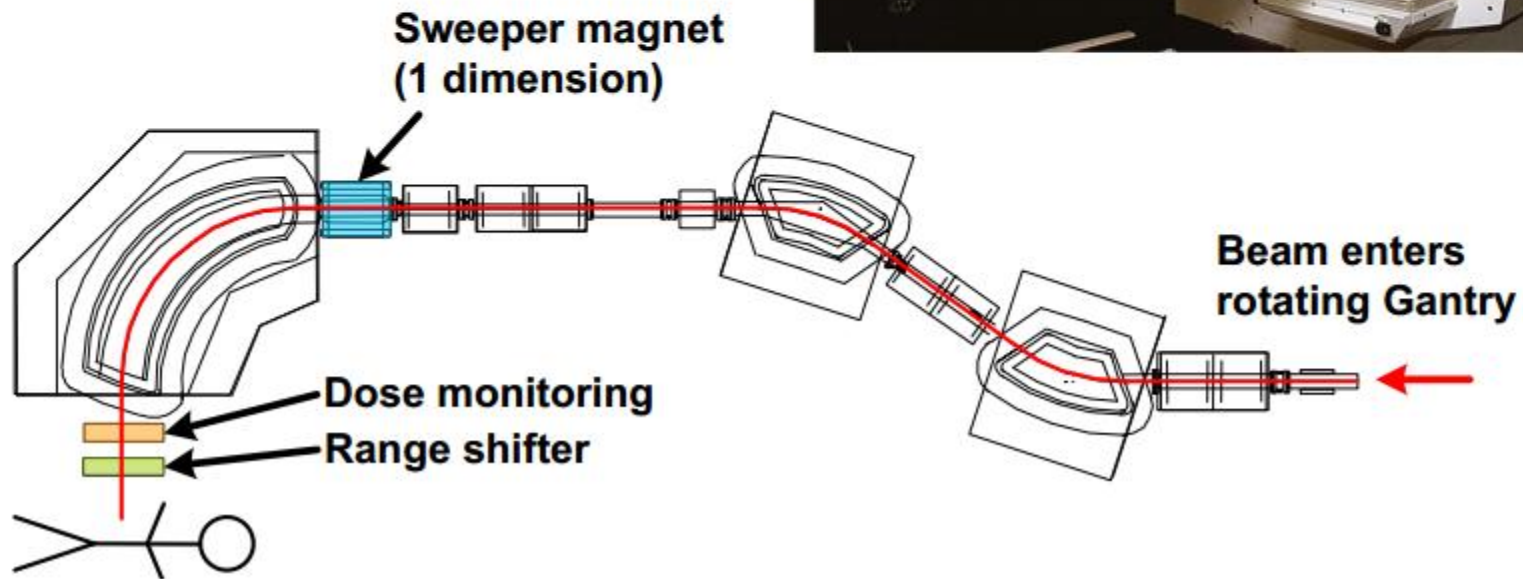
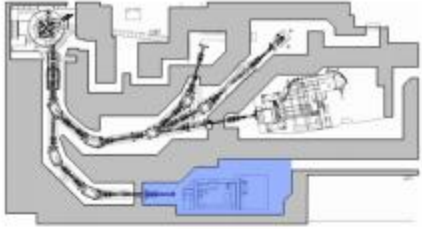
# Proton Radiation Therapy System Paul Scherrer Institute, Switzerland

- 250 MeV Proton accelerator (superconducting cyclotron)
- Beamlines to 4 user areas
- OPTIS
- Gantry 1
- Gantry 2
- Experimental area



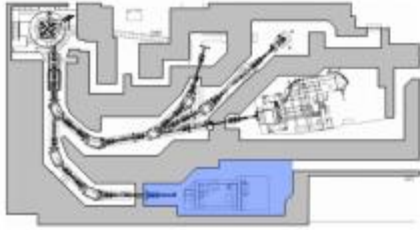


# Proton Radiation Therapy System Gantry 1



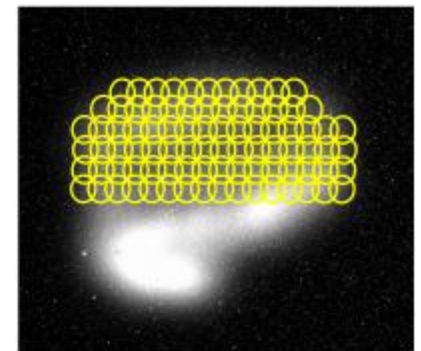


# Proton Radiation Therapy System Spot Scanning Technique

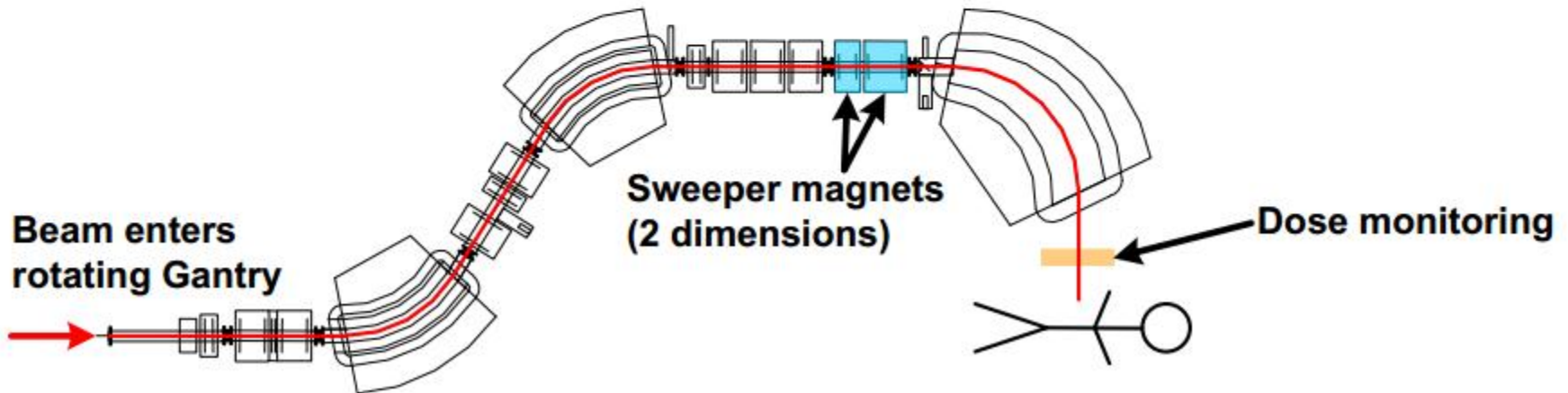


Elements of spot scanning:

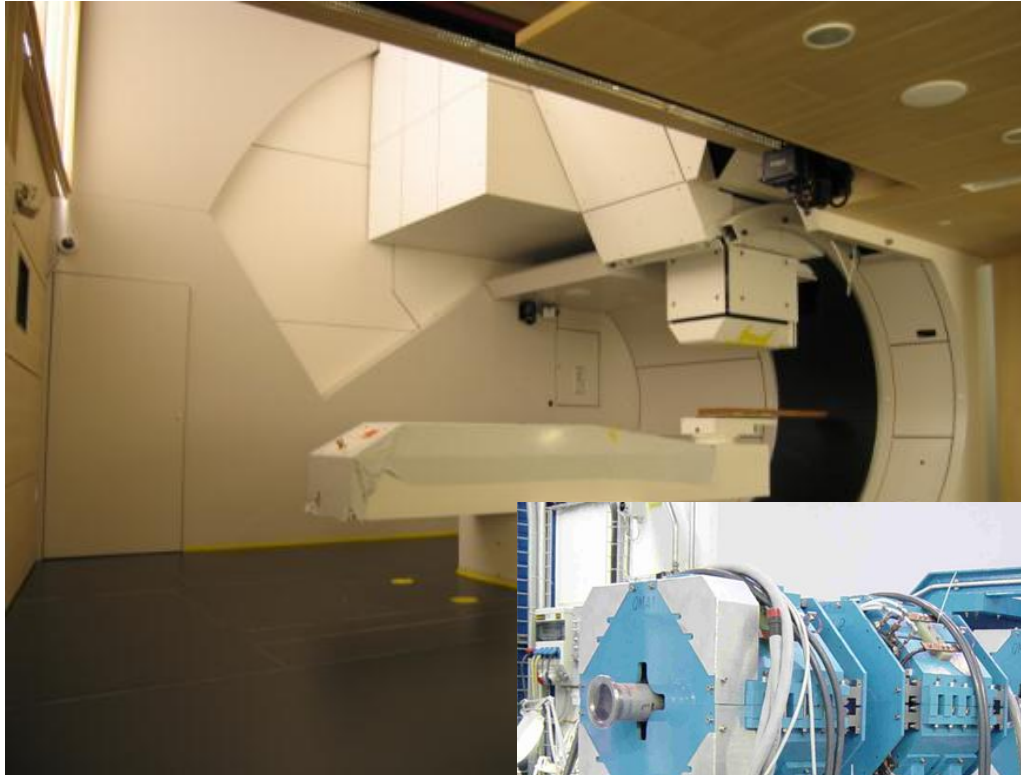
- Beam on/off            50  $\mu$ s
  - Sweeper magnet      5 ms/step
  - Range shifter         30 ms
  - Patient table          1 cm/s
- 
- 10'000 spots to treat 1 liter volume



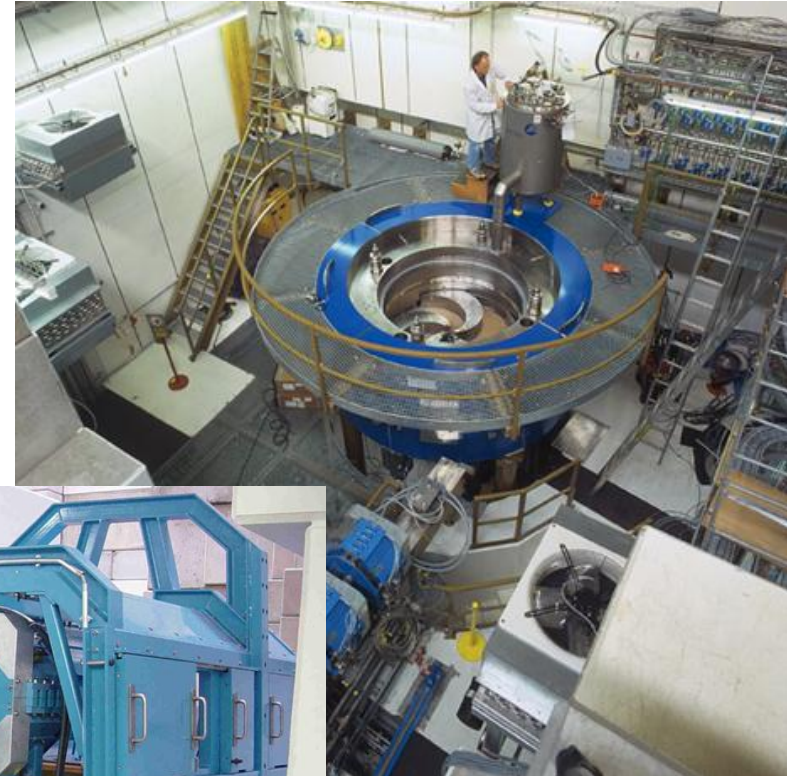
# Proton Radiation Therapy System Gantry 2



# Proton Therapy Machine Overview



Gantry



Cyclotron

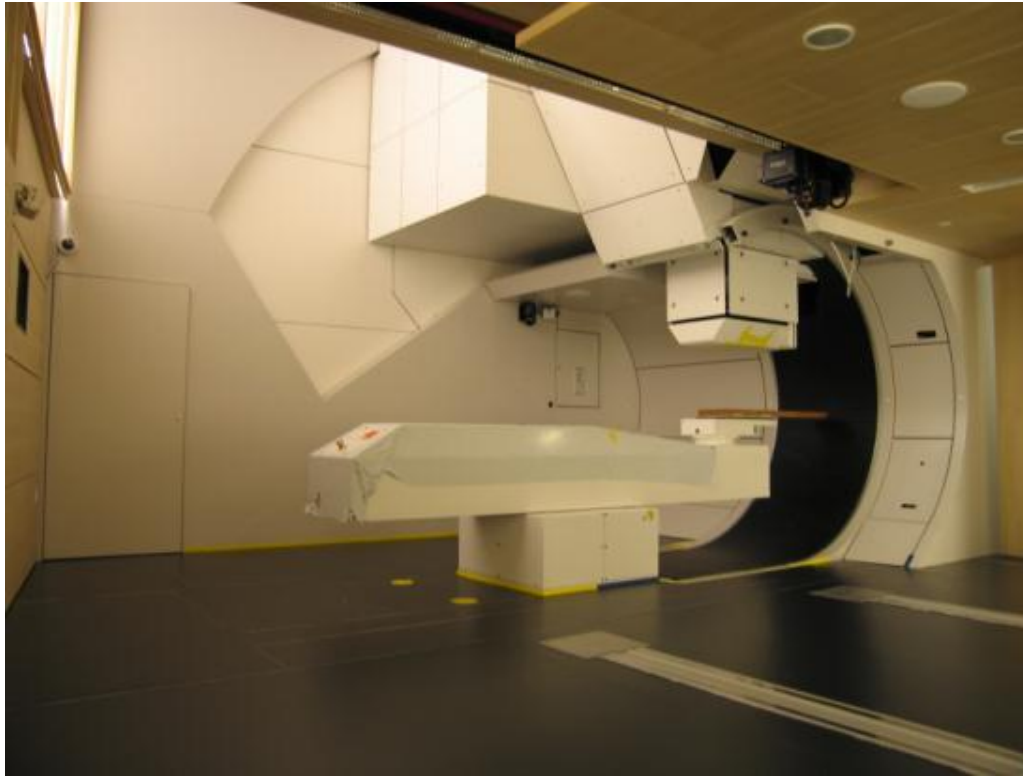


Beam path and  
control elements



# 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

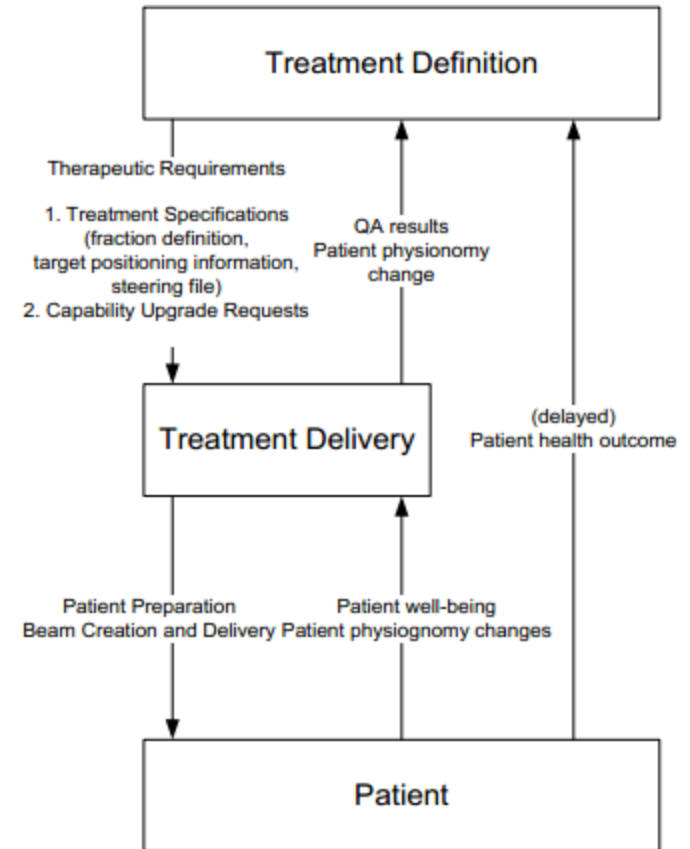
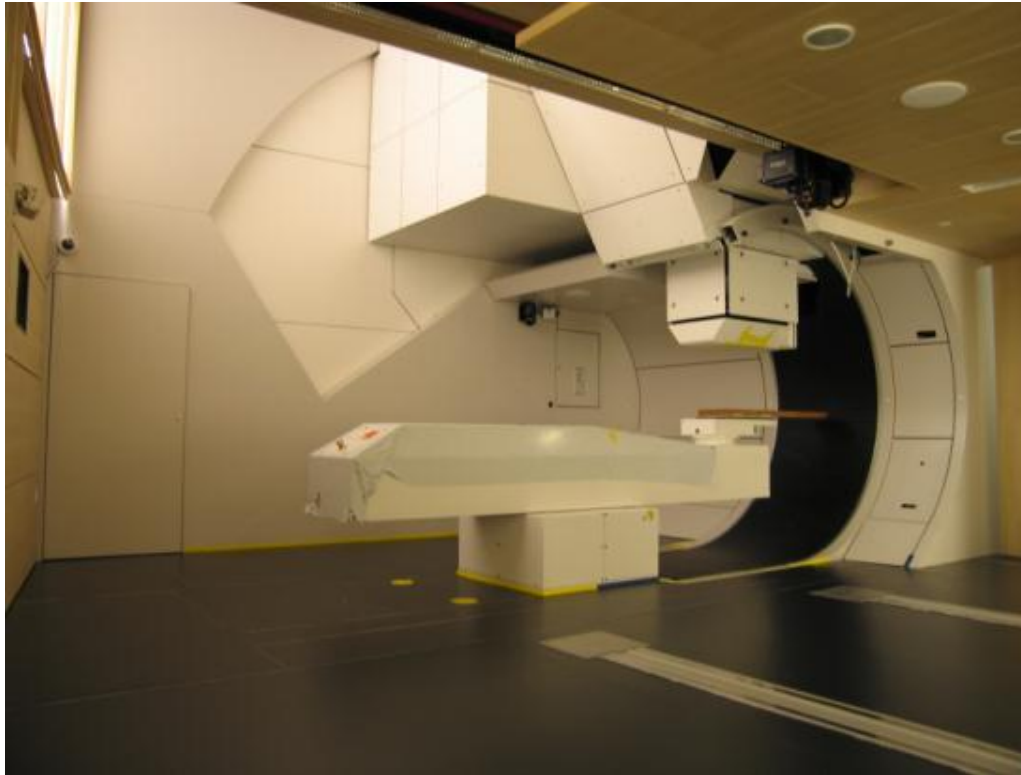


Figure 11 - High-level functional description of the PROSCAN facility (D0)

# Proton Therapy Machine Control Structure

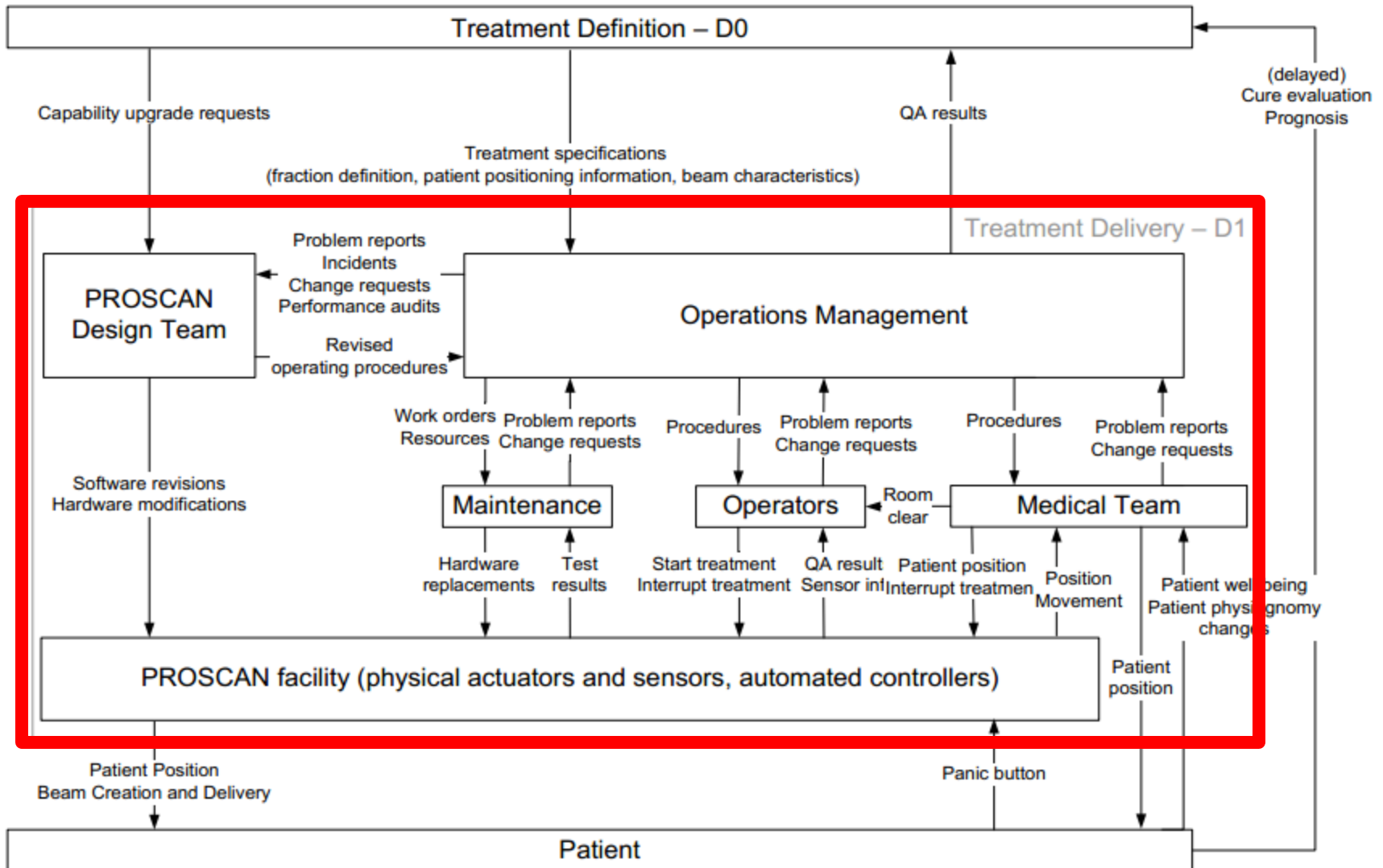
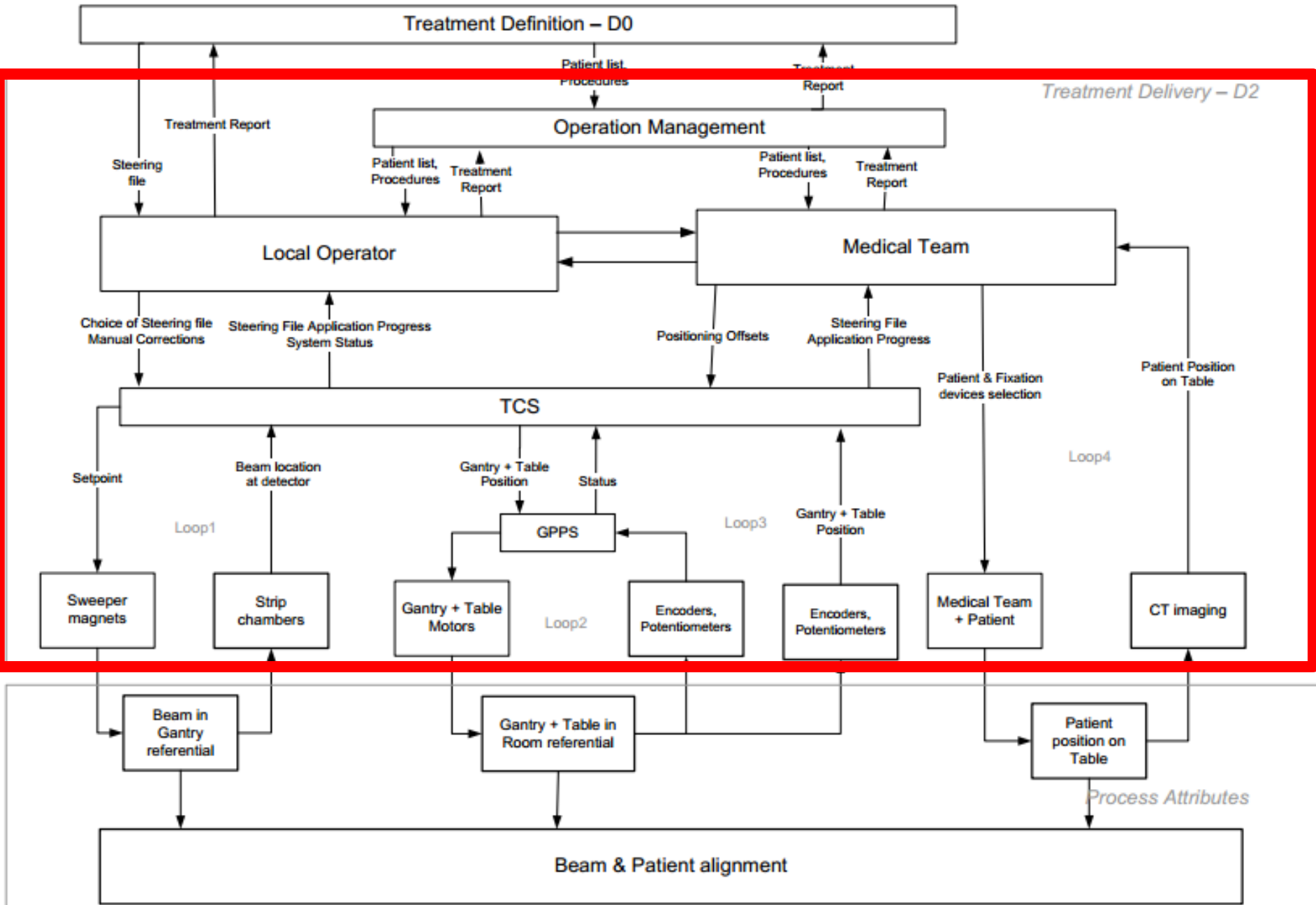


Figure 13 - Zooming into the Treatment Delivery group (D1)

# Proton Therapy Machine Detailed Control Structure



# STPA

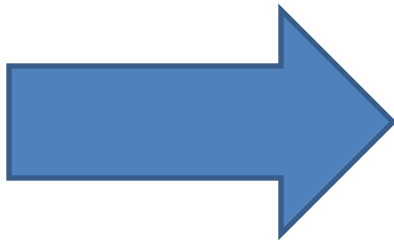
## (System-Theoretic Process Analysis)



- Identify accidents and hazards

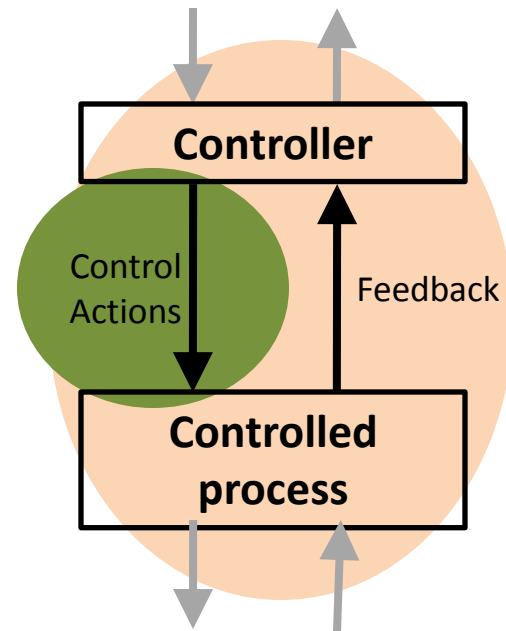


- Construct the control structure



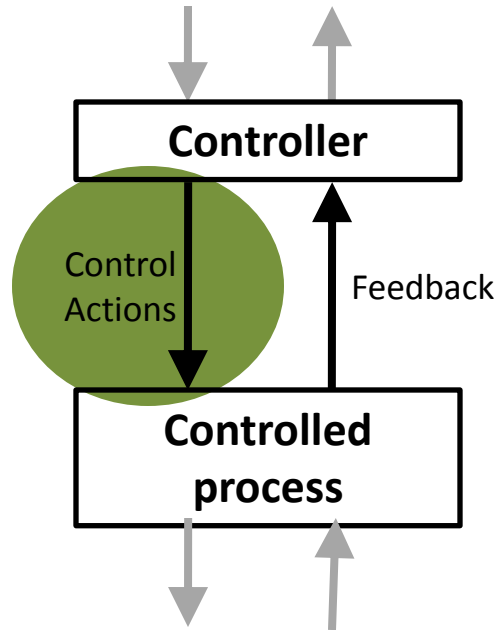
- Step 1: Identify unsafe control actions

- Step 2: Identify causal factors and control flaws





# 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)

	Not providing causes hazard	Providing causes hazard	Incorrect Timing/ Order	Stopped Too Soon / Applied too long
Control Action				

# Proton Therapy Machine Control Structure

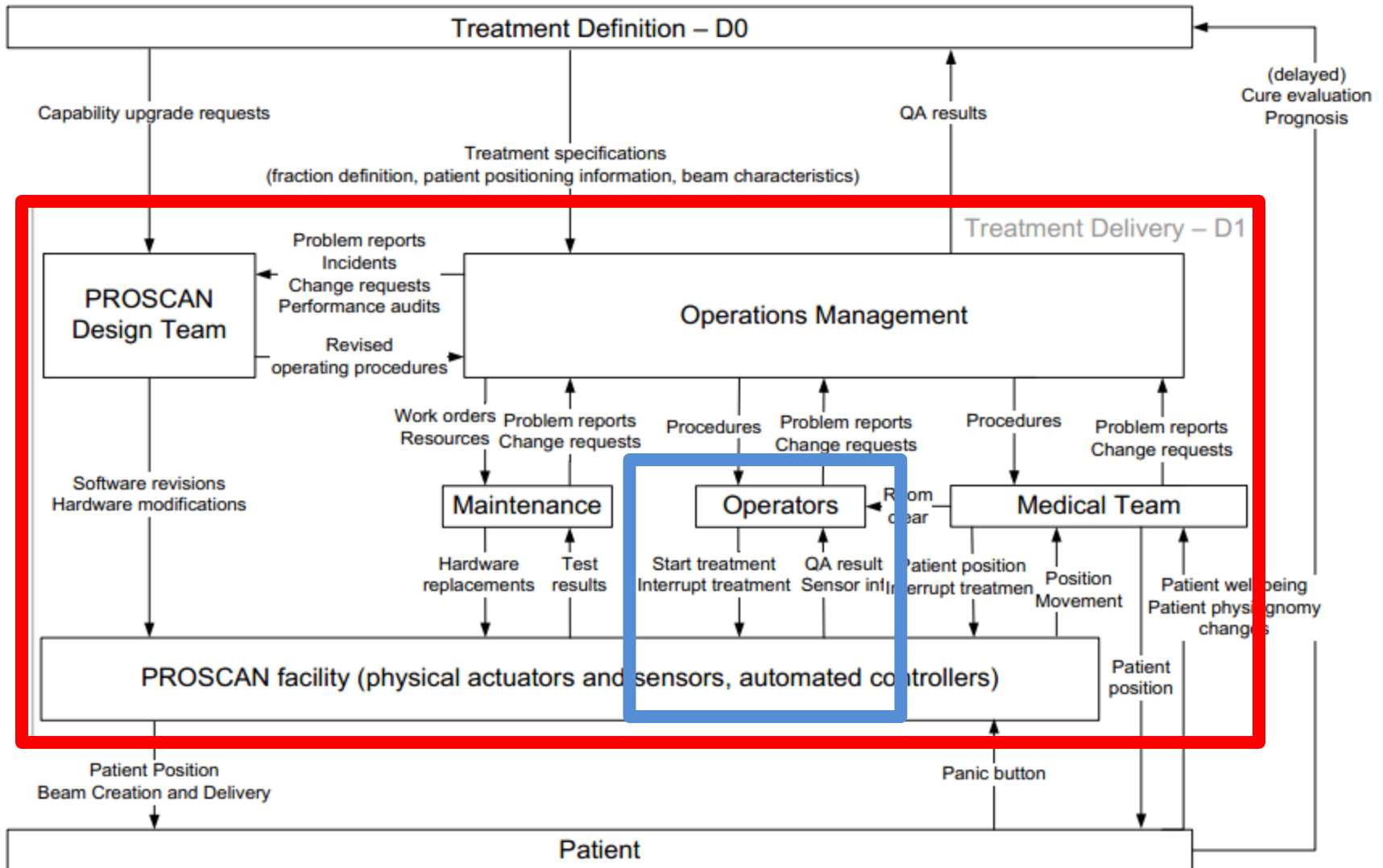
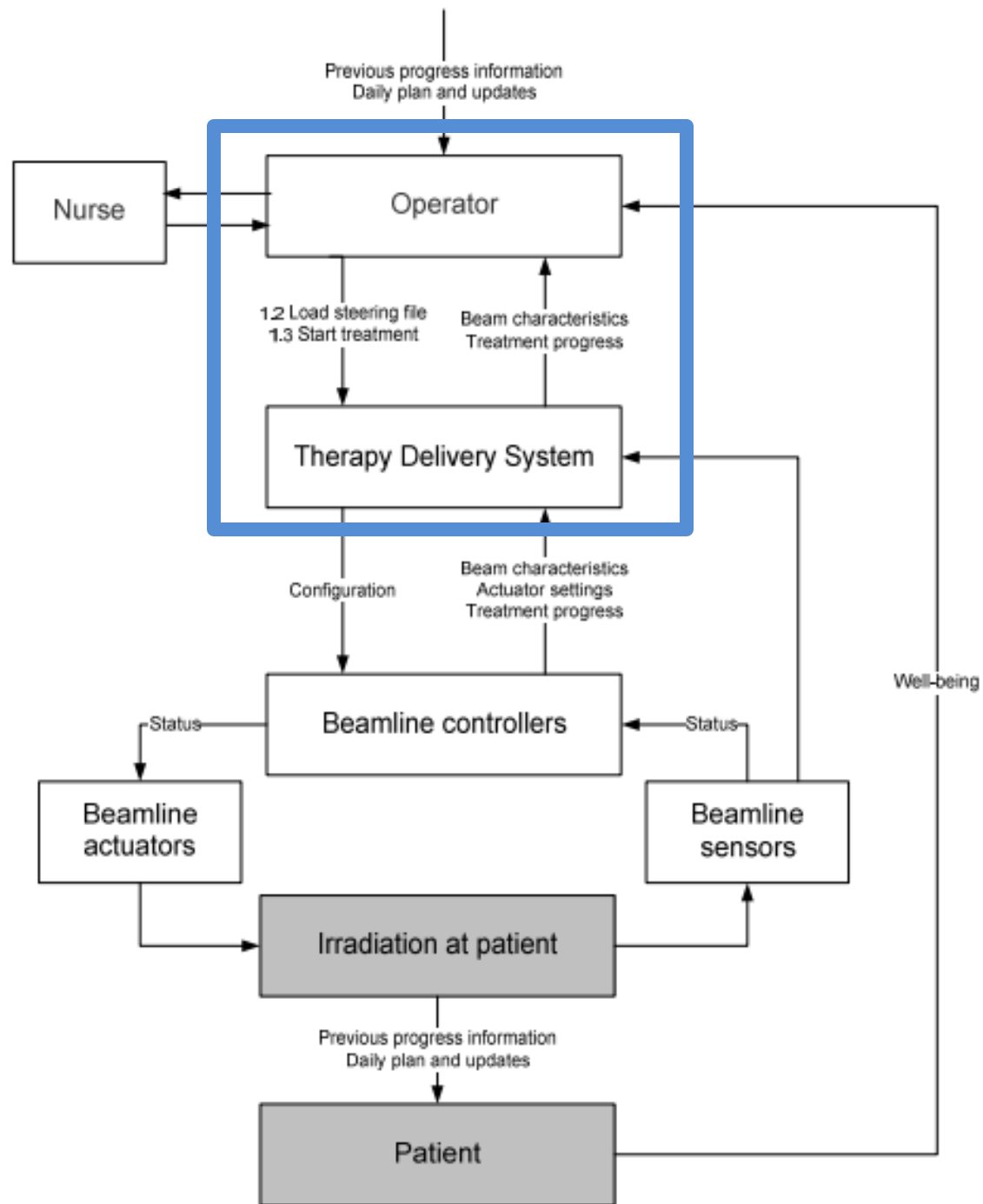


Figure 13 - Zooming into the Treatment Delivery group (D1)

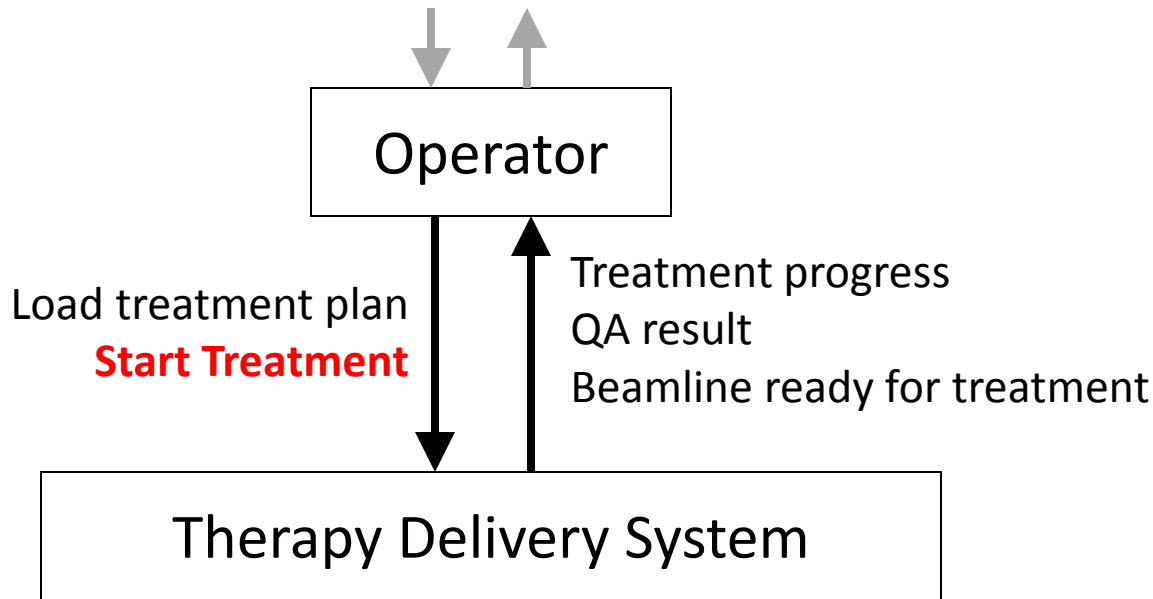
# Unsafe Control Actions

## 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



## 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

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

# Structure of an Unsafe Control Action

Example:

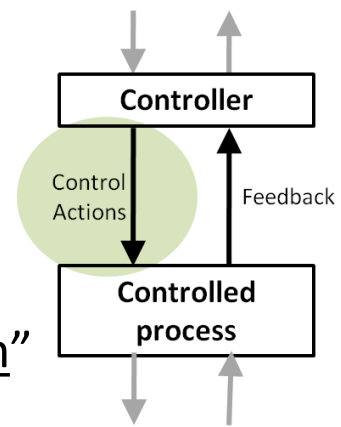
“Operator provides start treatment cmd while personnel is in room”

Type

Control Action

Context

Source Controller



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

## Unsafe Control Action

## Component Safety Constraint

Treatment is started while personnel is in room



Treatment must not be started while personnel are in the room

Treatment is started while the beamline is not ready to receive the beam



Treatment must not start before beamline is fully configured

Treatment is started when there is no patient at the treatment point



Treatment must not start until when patient is at the treatment point

Treatment is started without a treatment plan having been loaded



Treatment must not start until a new treatment plan has been loaded

# STPA

## (System-Theoretic Process Analysis)



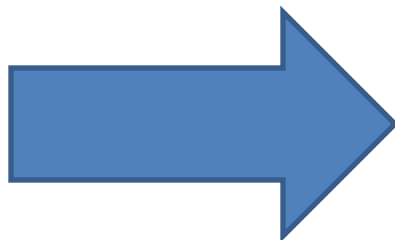
- Identify accidents and hazards



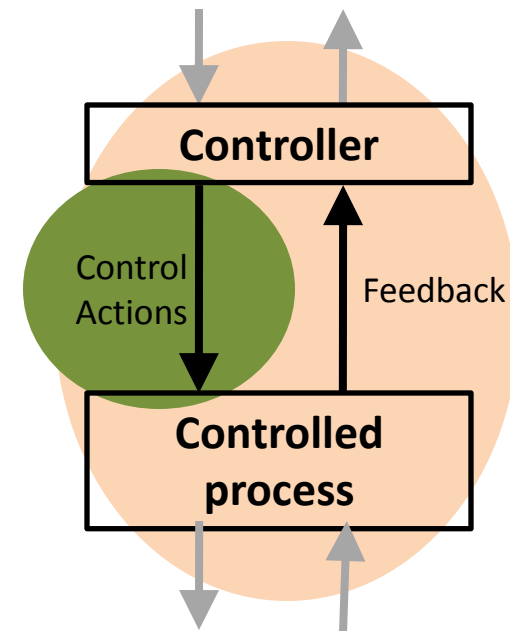
- Construct the control structure



- Step 1: Identify unsafe control actions

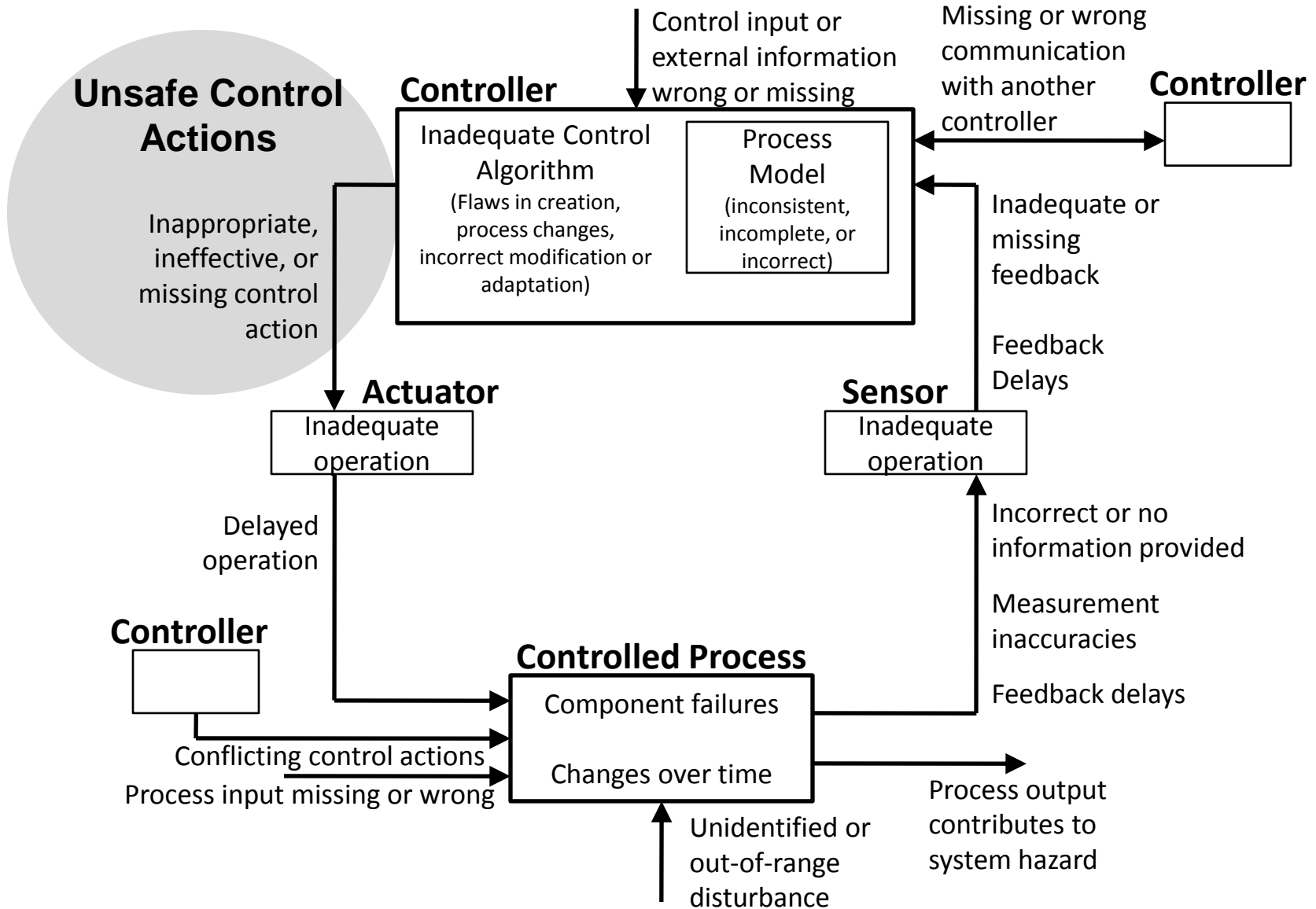


- Step 2: Identify causal factors and control flaws

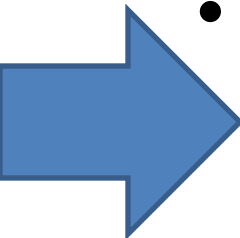




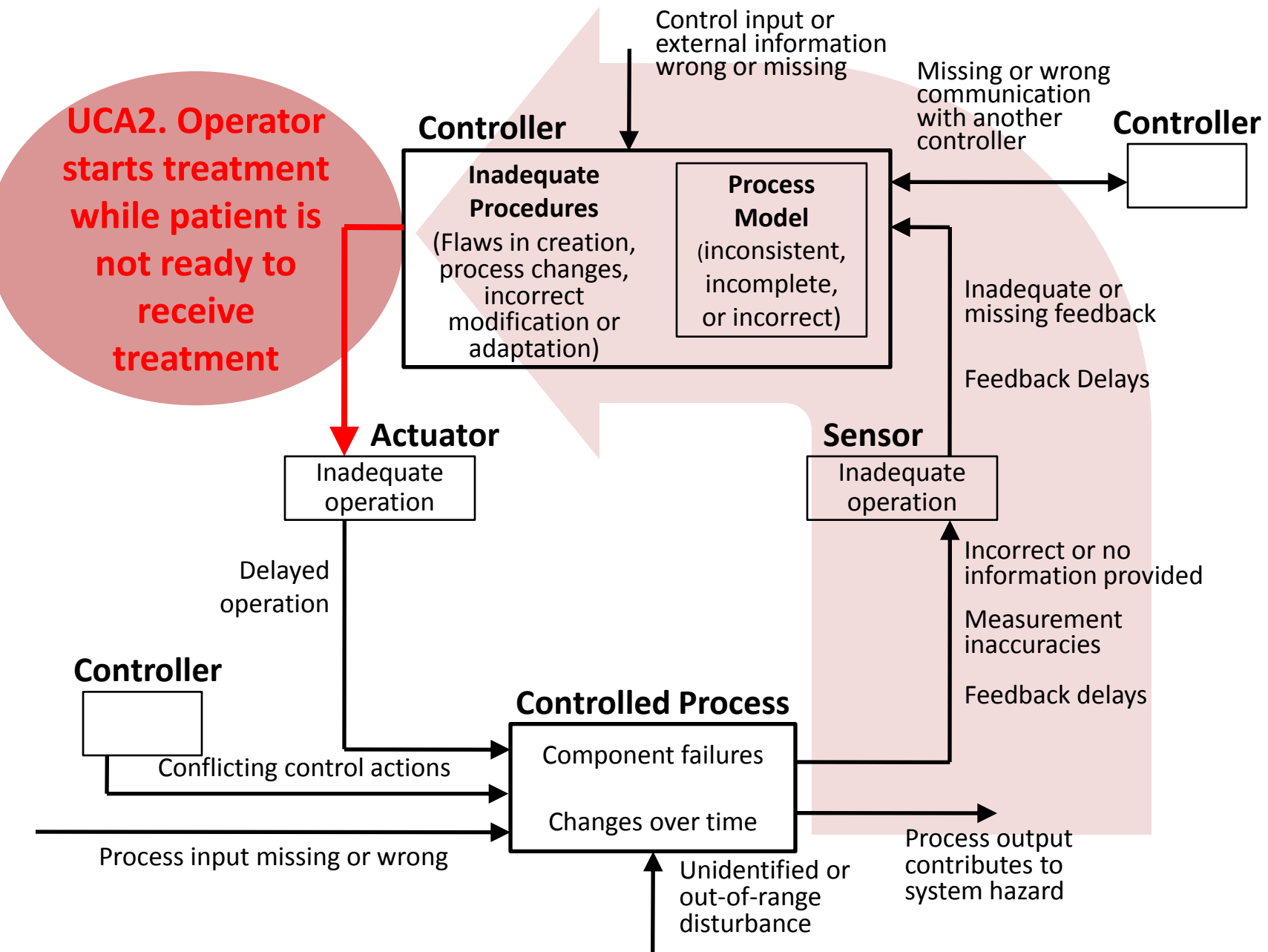
# STPA Step 2: Identify Control Flaws



# STPA Step 2: Identify Causal Factors

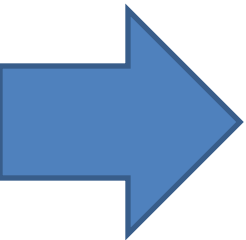
- 
- Select an Unsafe Control Action
    - A. Identify causal factors that explain how it could happen
      - Develop causal accident scenarios
    - B. Identify causal factors that explain how control actions may not be followed or executed properly
      - Develop causal accident scenarios
  - Identify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

# Step 2A: Potential causes of UCAs

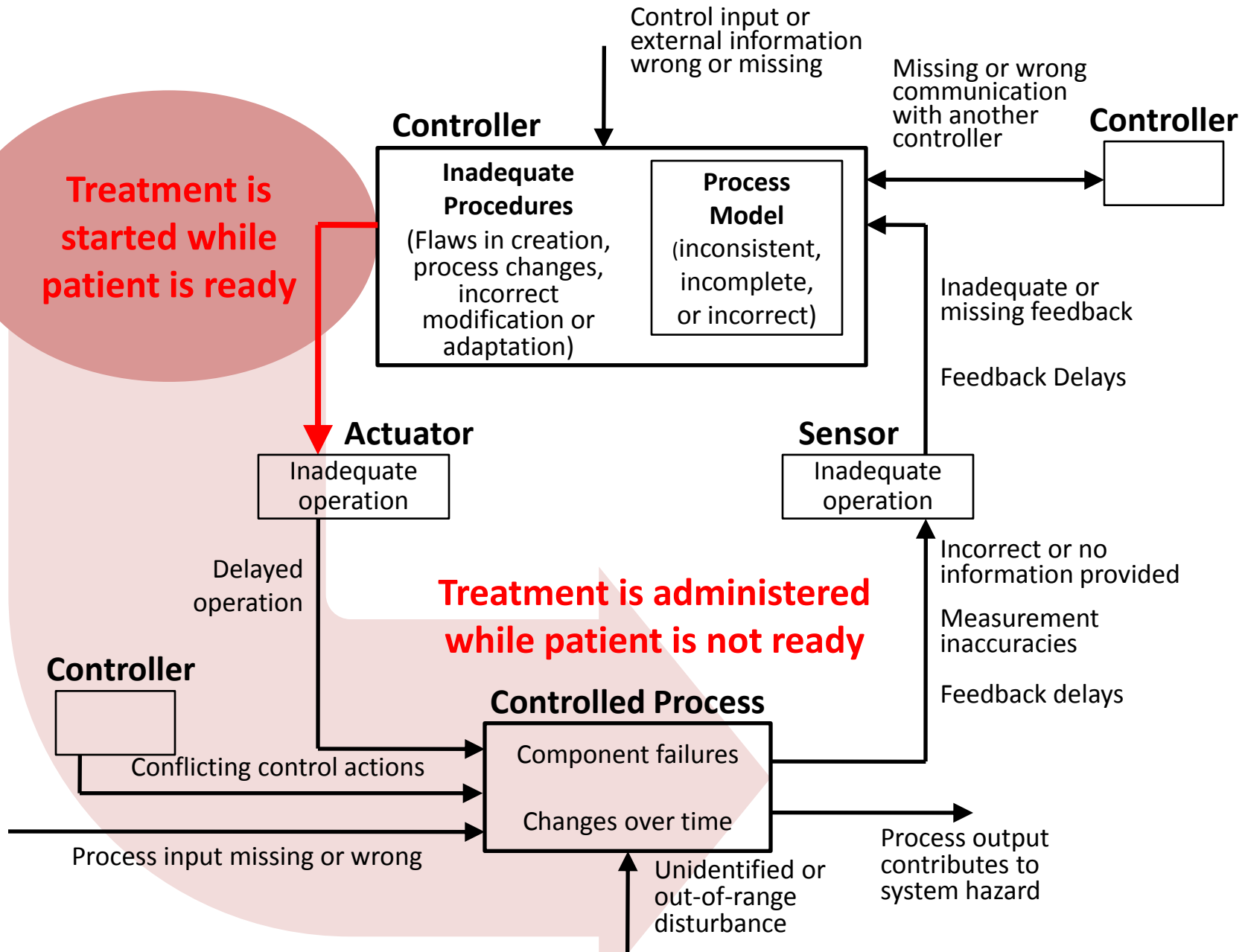


# STPA Step 2: Identify Causal Factors

- 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



# 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.)

# For more information...

- Email: [jthomas4@mit.edu](mailto:jthomas4@mit.edu)
- STPA Primer
  - Not a book or academic paper
  - Written for industry to provide guidance in learning STPA
  - “living” document
  - Google “STPA Primer”
- Website
  - <http://psas.scripts.mit.edu/home>
  - Annual MIT conference in March
  - Presentations with examples in every industry available
- Book
  - “Engineering a Safer World”, 2012
  - Free PDF download at MIT Press website
- Dissertation
  - “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