STPA Tutorial

MedTech Workshop
Proactive System Safety for Medical Devices Using STPA

STPA for Knee Replacement Robot

John Thomas MIT
Brent Bailey 8iRobotics



STPA: System-Theoretic Process Analysis

FMEA asks: What can fail?

- Focuses on component failures as the root cause of accidents
- Works well for hardware with discrete, known failure modes
- Each component analyzed independently

STPA asks: How can controllers get confused?

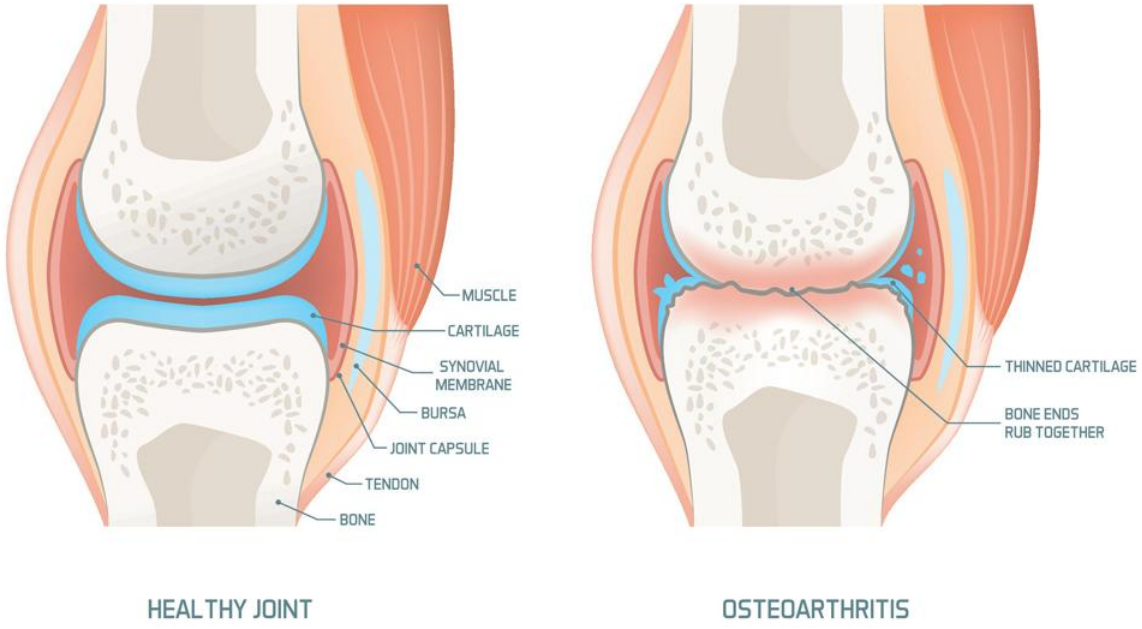
- Most accidents occur when components work as designed, but interact in unsafe ways
- Software, humans, and automation can violate safety constraints without "failing"
- Safety is a control problem, not just a reliability problem

When STPA adds the most value

Software & automation in safety-critical loops	Human judgment is part of the system	Hazards arise from interactions, not just defects	Requirements need to be derived upfront

STPA and FMEA answer different questions. FMEA: what can fail? STPA: what unsafe control behavior — from software, automation, or humans — can lead to harm, even when nothing has failed?

Osteoarthritis



- Protective cartilage at end joints wears over time
- Primarily 65-84 years old
- 55-60% female

irobotics

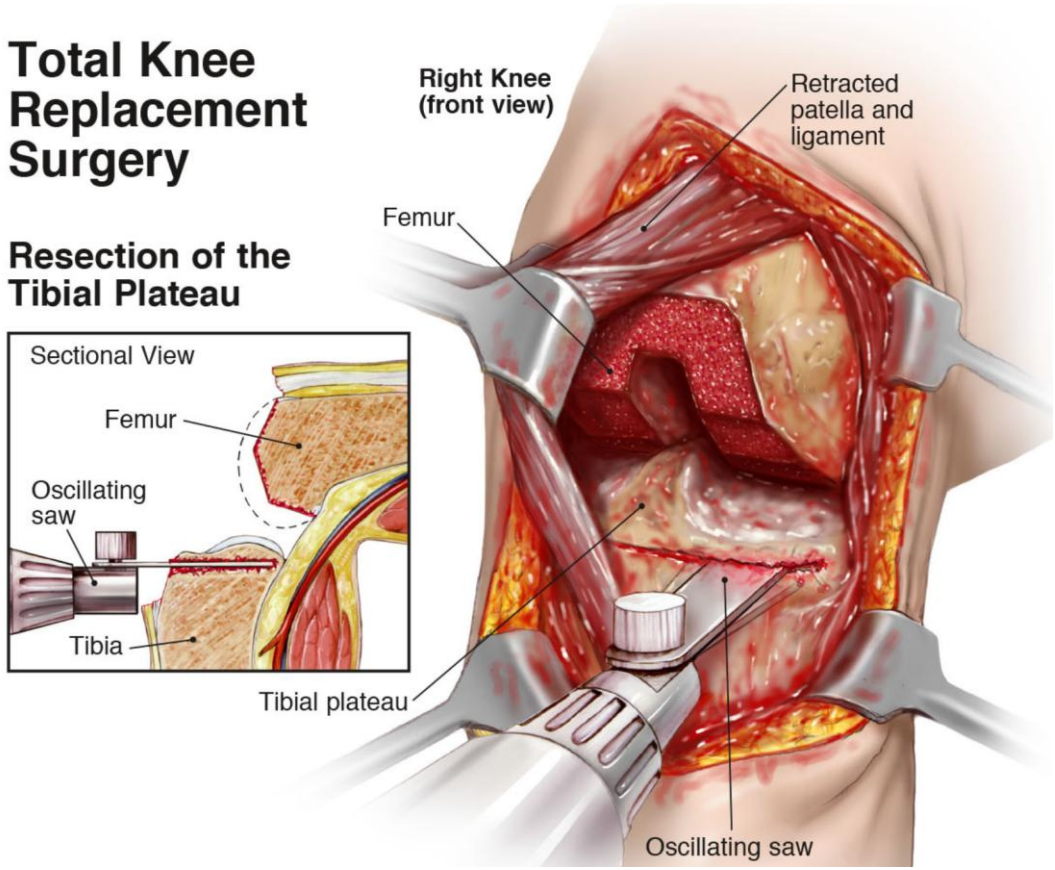
Knee Replacement Implants



- Knee surface of the femur and tibia are
  - Removed
  - Replaced
- Implants are off-the-shelf items
  - 150 brands of knee implants
  - Up to 20 sizes and profiles per brand

robotics

Knee Replacement Surgery



- Need to make exact cuts along specific planes
- Possibility to damage or sever
  - Ligaments
  - Muscles and tendons
  - Nerves
  - Blood vessels

airobotics

Preparations for Surgery



- Pre-operative planning
  - Specify the cutting planes in the CT images
  - Specify keep-out zones
- Calibration
  - Location of sawblade with respect to the end of the robot
  - Pointer used in registration
- Registration
  - Location of CT image with respect to tracking arrays

Ai robotics

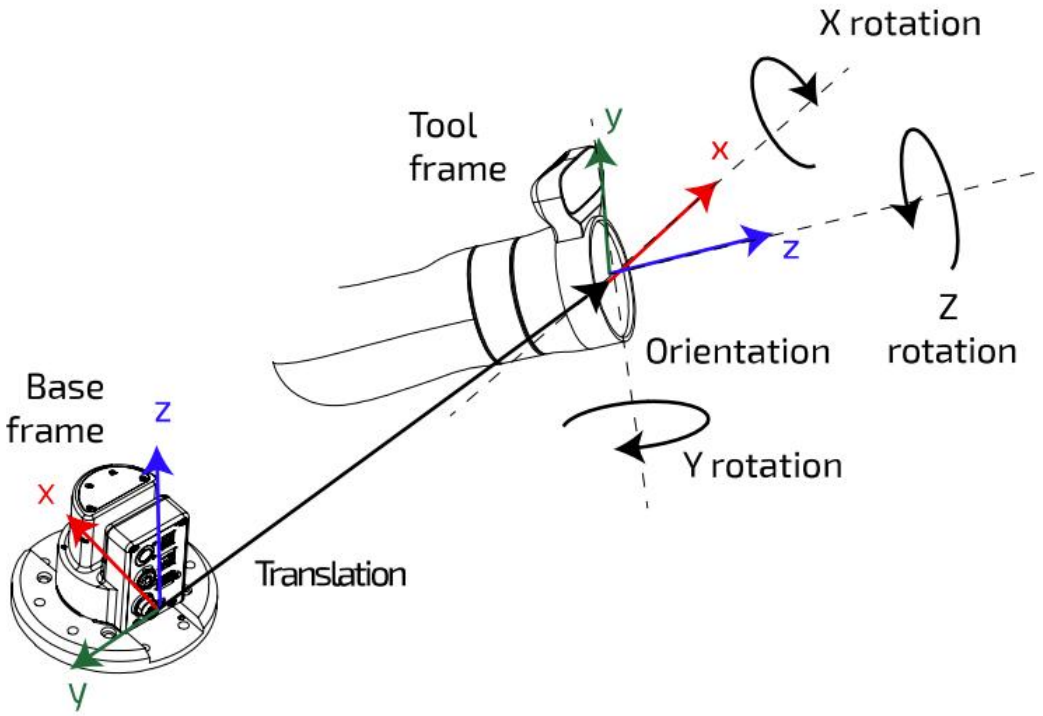
Robotics



- Provide accuracy and repeatability
- Semi-autonomous control
  - Always controlled by Surgeon
  - Provides cut planes
  - Surgeon moves the sawblade in the plane
  - Enforces keep-out zones
  - Surgeon triggers the cutting action

robotics

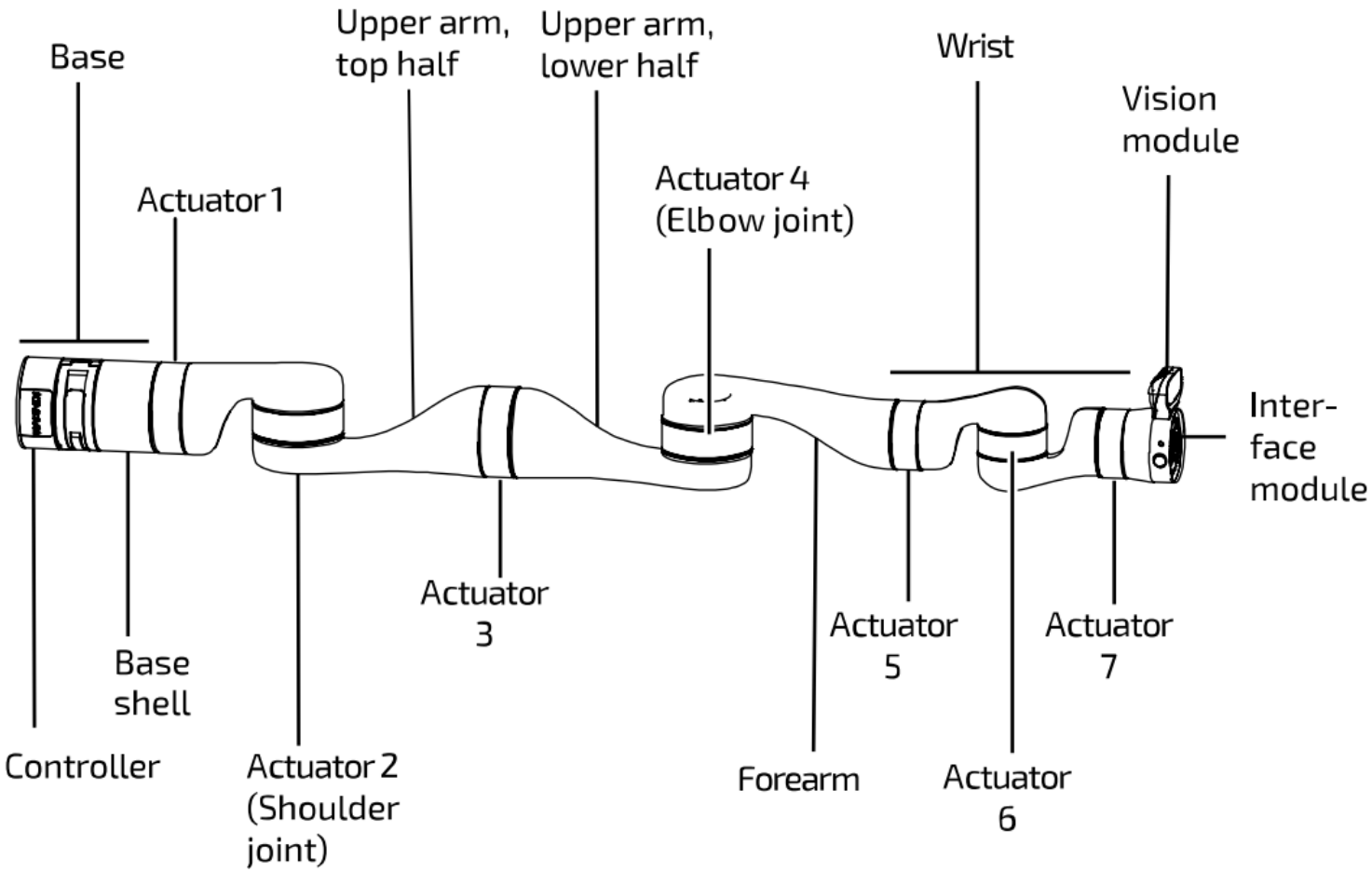
Task-space



- What the robot is doing
  - Pose (position and orientation) of the robot's end-effector in the physical world

irobotics

Joint-space



- How the robot is doing it
  - The position and control of the joints

irobotics

Non-linear Kinematics

- Pose is given in 6 DoFs
- Robot DoFs are equal to the number of movable joints
  - Can have more than, equal to, or less than 6 DoFs
  - Not necessarily a one-to-one transformation
- Transformation matrix contains sine and cosine of joint angles

- Moving from joint-space to task-space is relatively easy
- Moving from task-space to joint-space can be difficult or indeterminate

ai robotics

Singularities

- Specific pose where motion in one or more task-space dimensions is lost
  - Control becomes an issue, as small perturbations or noise cause huge current demands
- Reach singularities
  - At full extension cannot more father out
- Gimbal locking
  - Two or more joint axis align

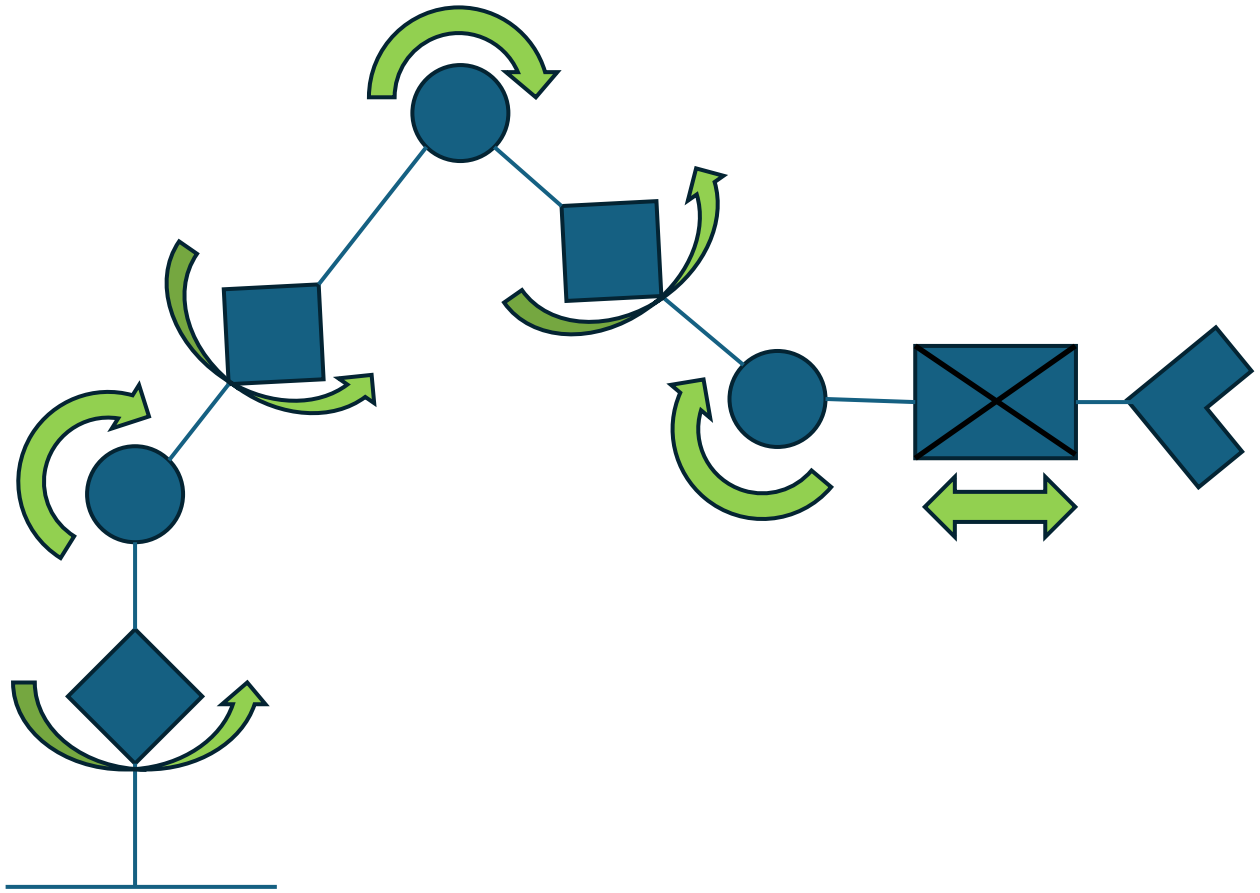
robotics

Sensor Integration

- Navigation
  - Transform tracking arrays to robot coordinates
- Joint
  - Motor-side speed and torque
  - Output-side displacement
- End-effector
  - 6-dof sensor

ai robotics

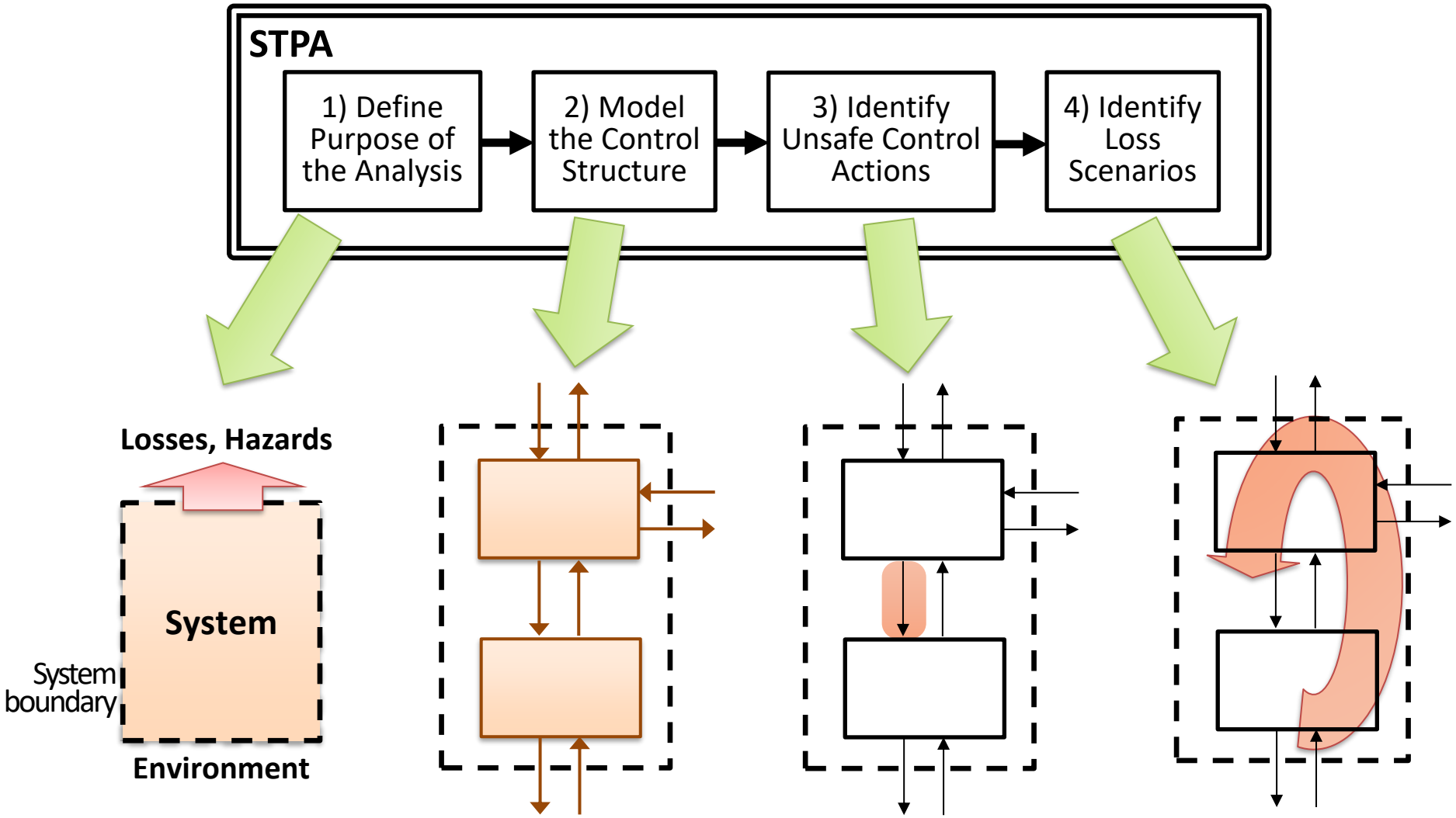
Our Robot



- Plane gets defined by first 5 degrees of freedom
- Last 2 degrees of freedom allow the Surgeon to move the sawblade in the plane

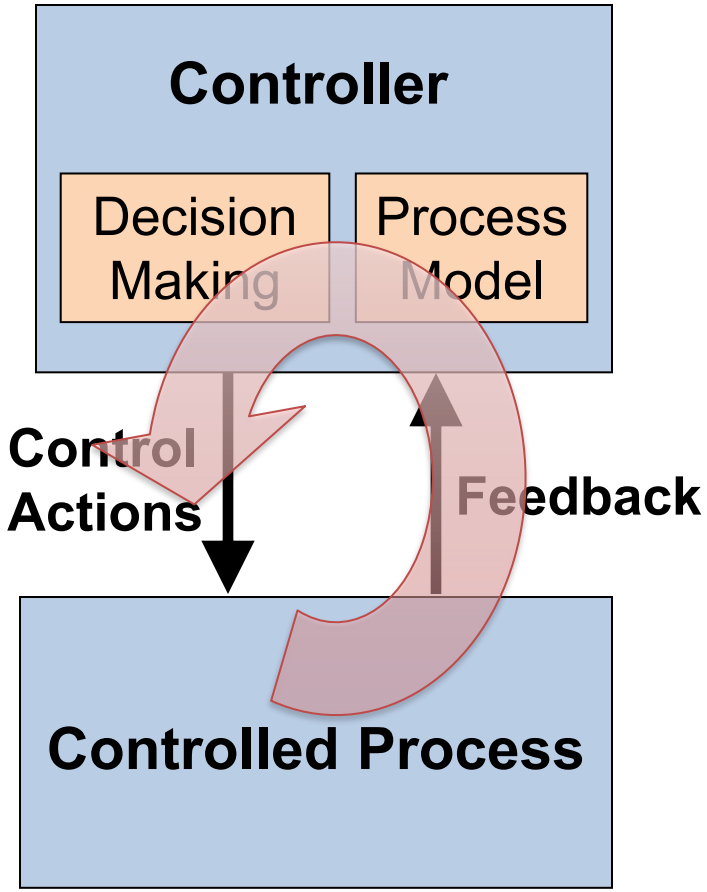
robotics

STPA Overview



(Leveson and Thomas, 2018)

Basic control loop

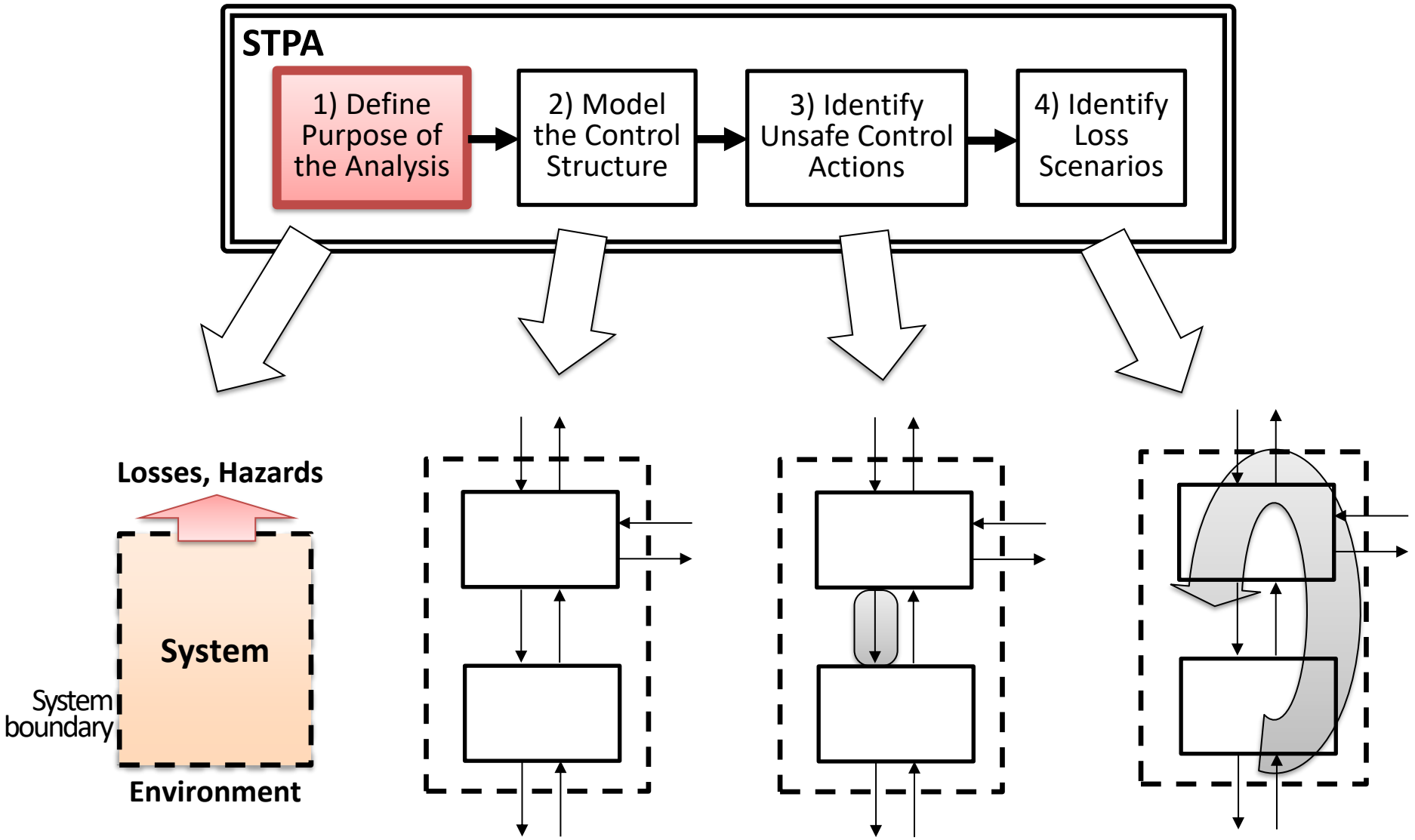


- **Control actions** are provided to affect a controlled process
- **Feedback** may be used to monitor the process
- **Process model** (beliefs) formed based on feedback and other information
- **Decision making** determines appropriate control actions given current beliefs

© Copyright 2026 John Thomas

First: **Without** Robot

STPA



(Leveson and Thomas, 2018)

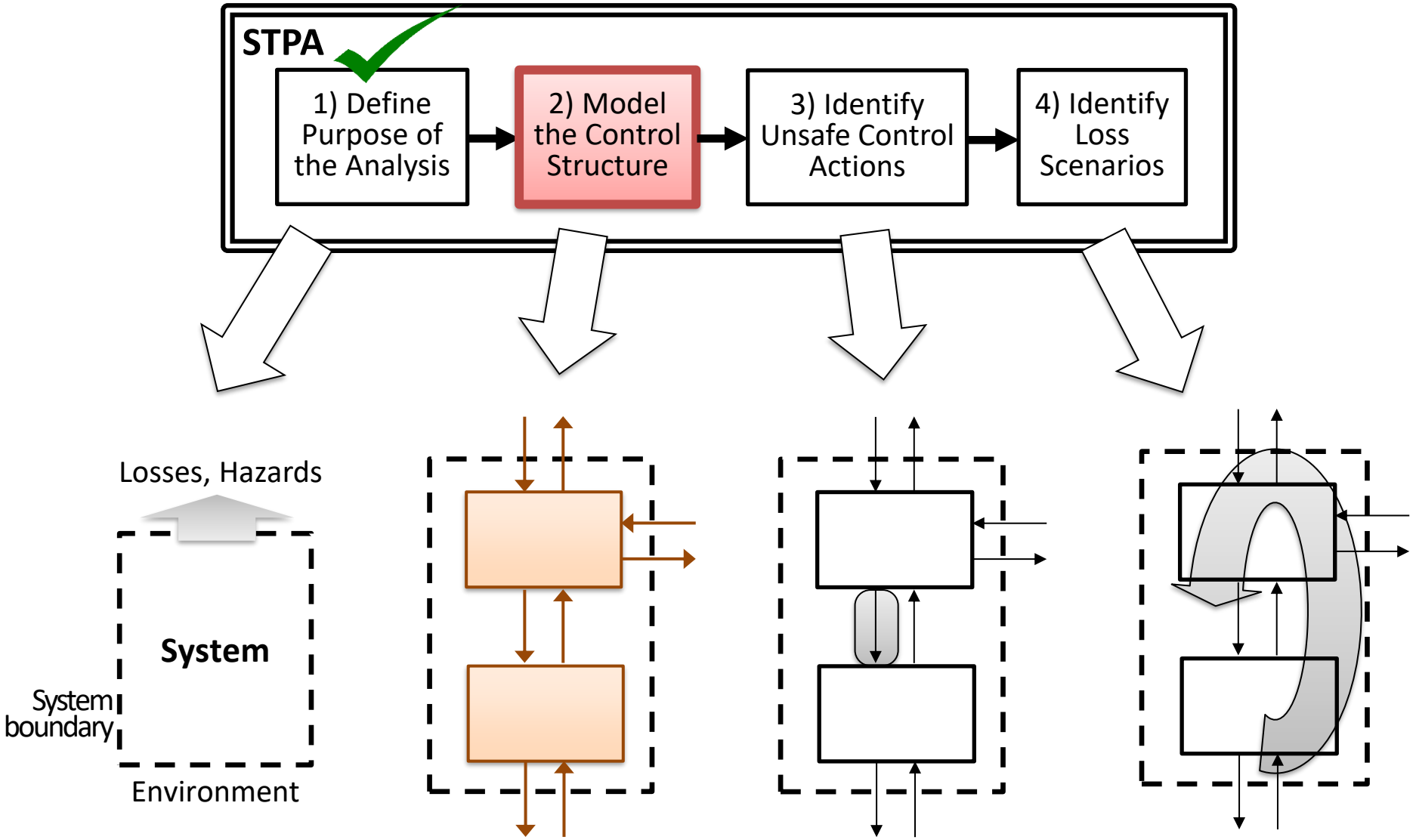
Step 1: Losses and Hazards

Losses

- [L-1] Loss of life or injury to patient or user.
- [L-2] Loss of implanting artificial joint.
- [L-3] Loss of patient information.

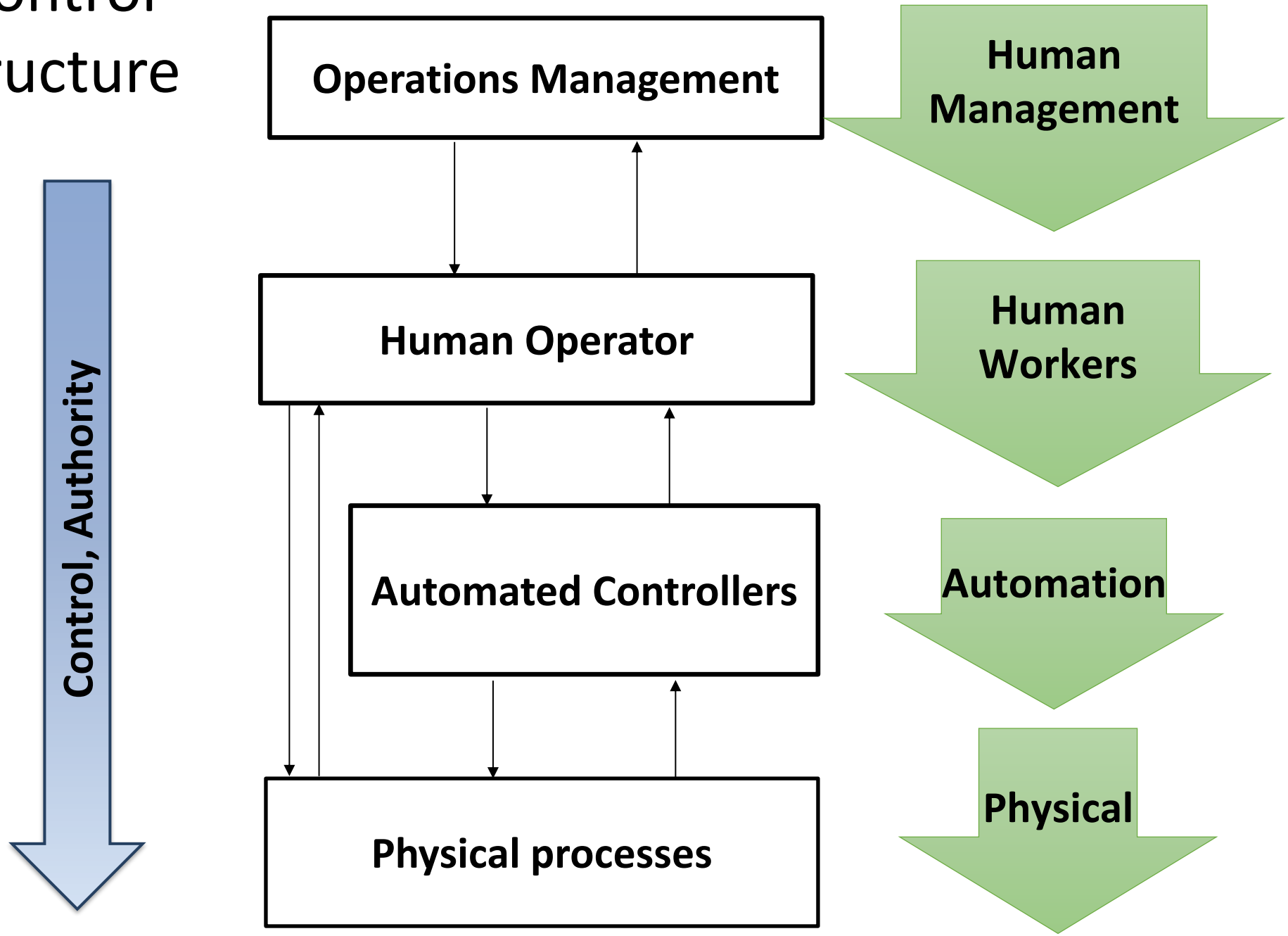
Hazards

- [H-1] The Patient does not receive the planned surgical procedures [L-2]
  - [H-1.1] Surgical procedure delay
  - [H-1.2] More Complex Surgery
  - [H-1.3] Additional Surgery
  - [H-1.4] Cancelled or Aborted Surgical procedure
- [H-2] The Patient's legs result in being of different lengths [L-1].
- [H-3] The Patient's tendons or nerves fibers are damaged [L-1].
- [H-4] The Surgeon cuts herself [L-1].



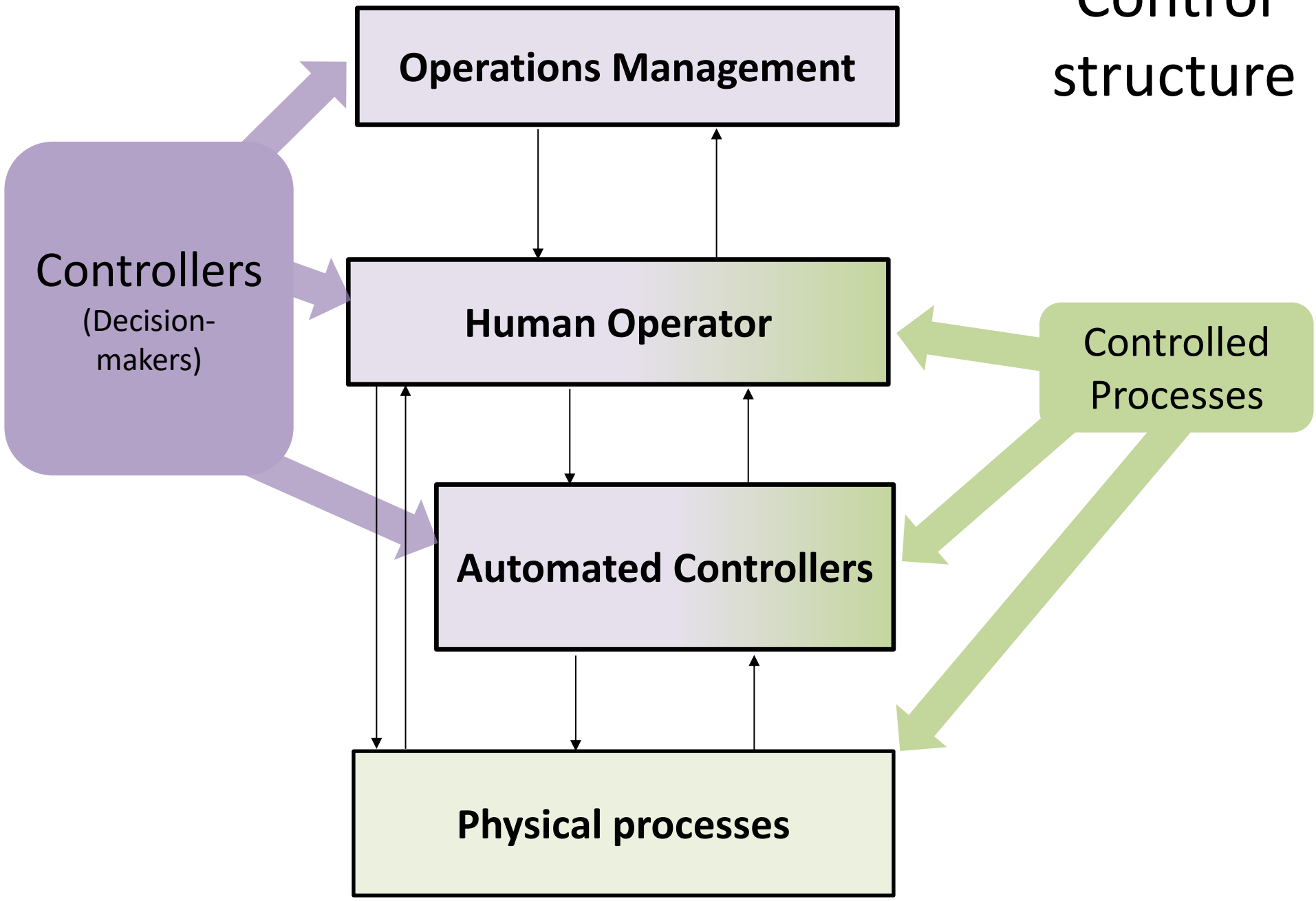
(Leveson and Thomas, 2018)

Control structure



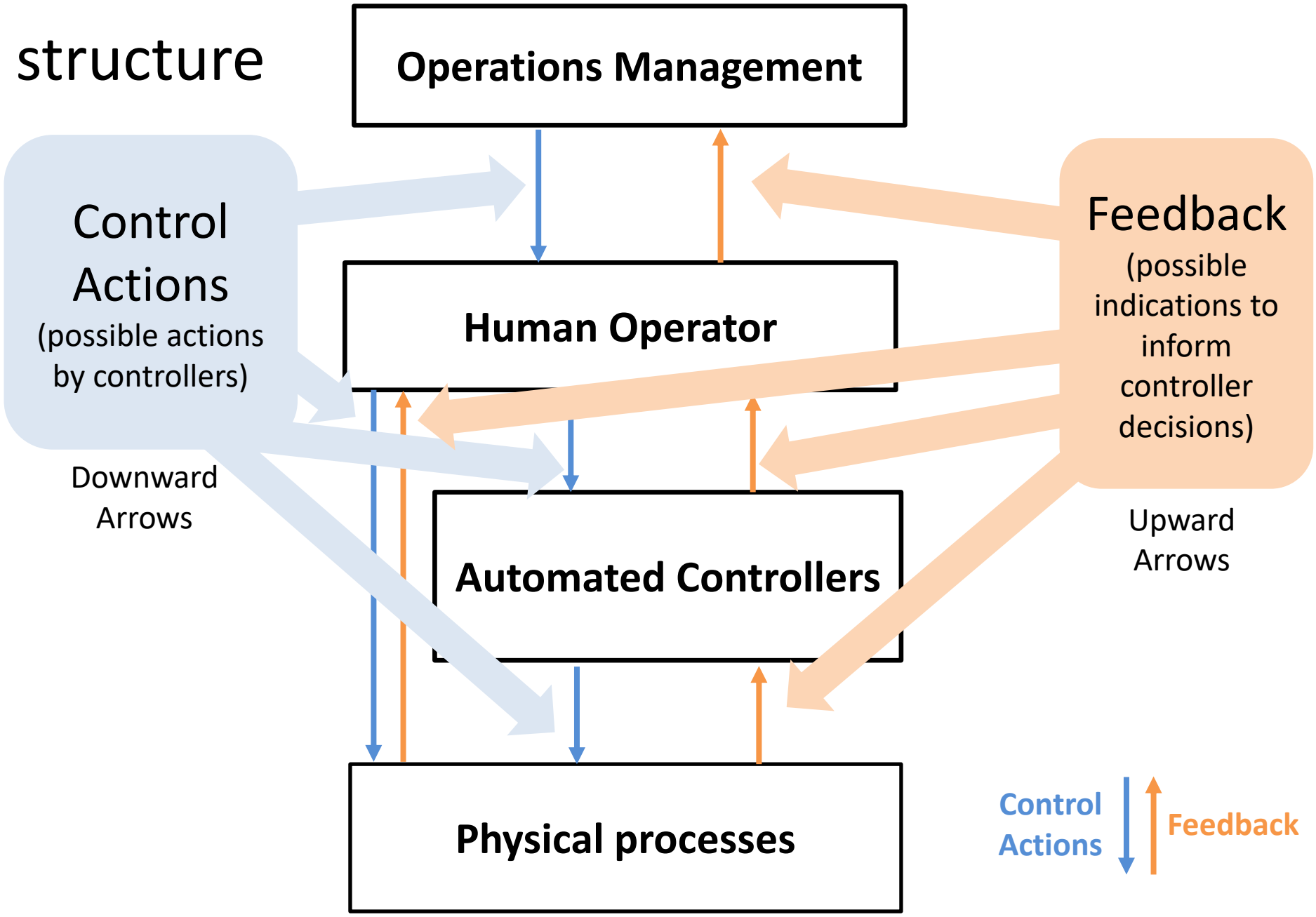
© Copyright 2026 John Thomas

Control structure



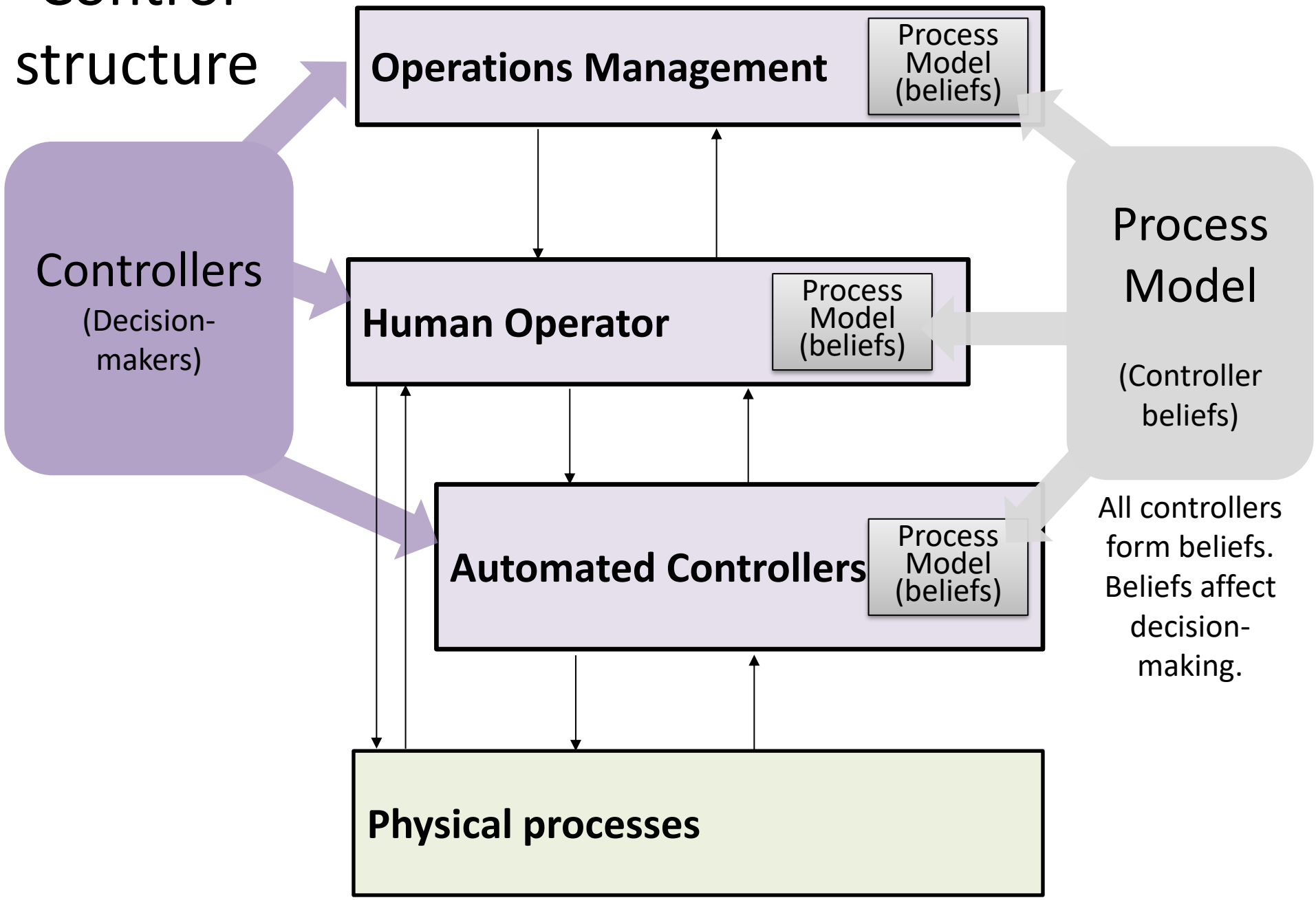
© Copyright 2026 John Thomas

Control structure



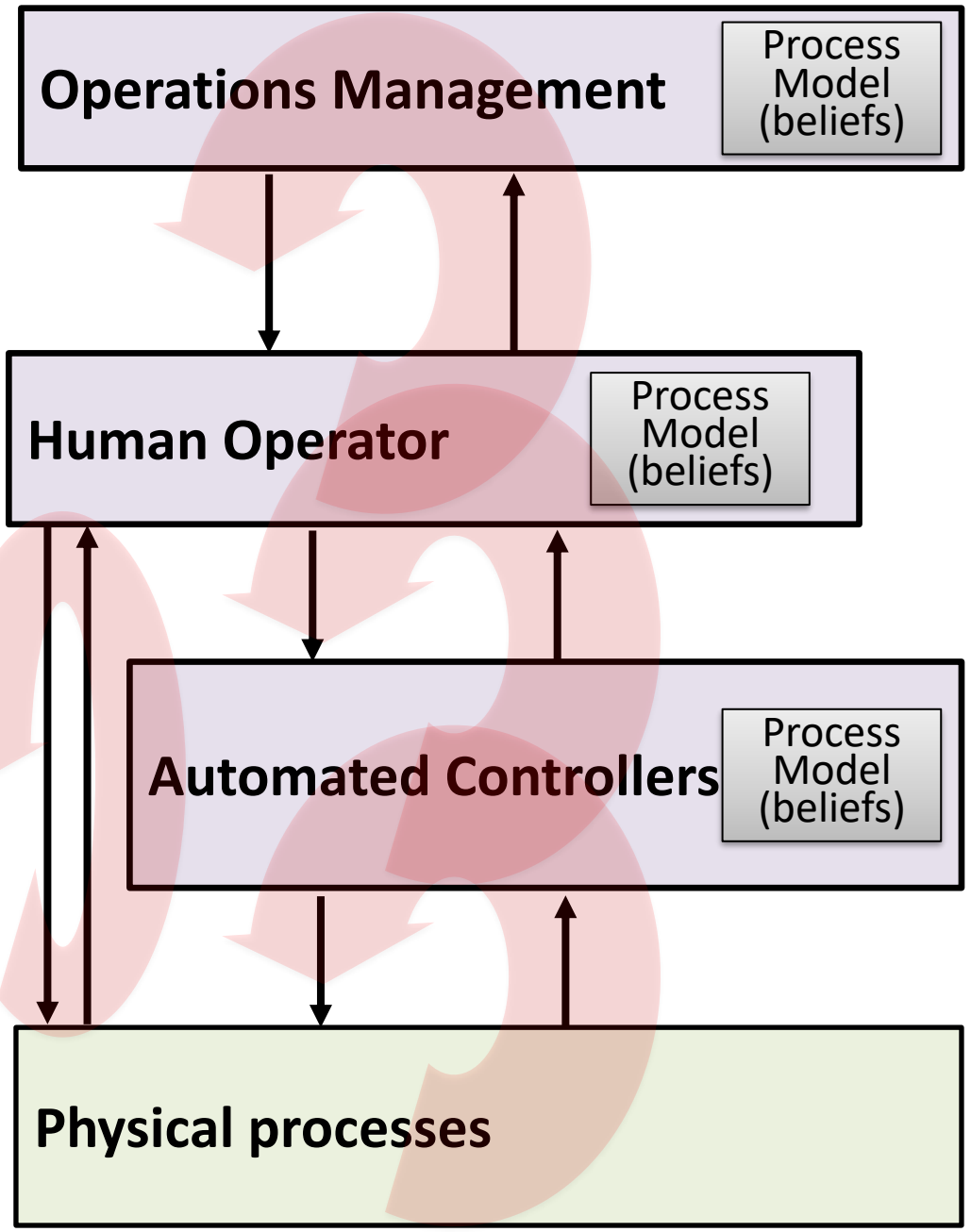
© Copyright 2026 John Thomas

Control structure



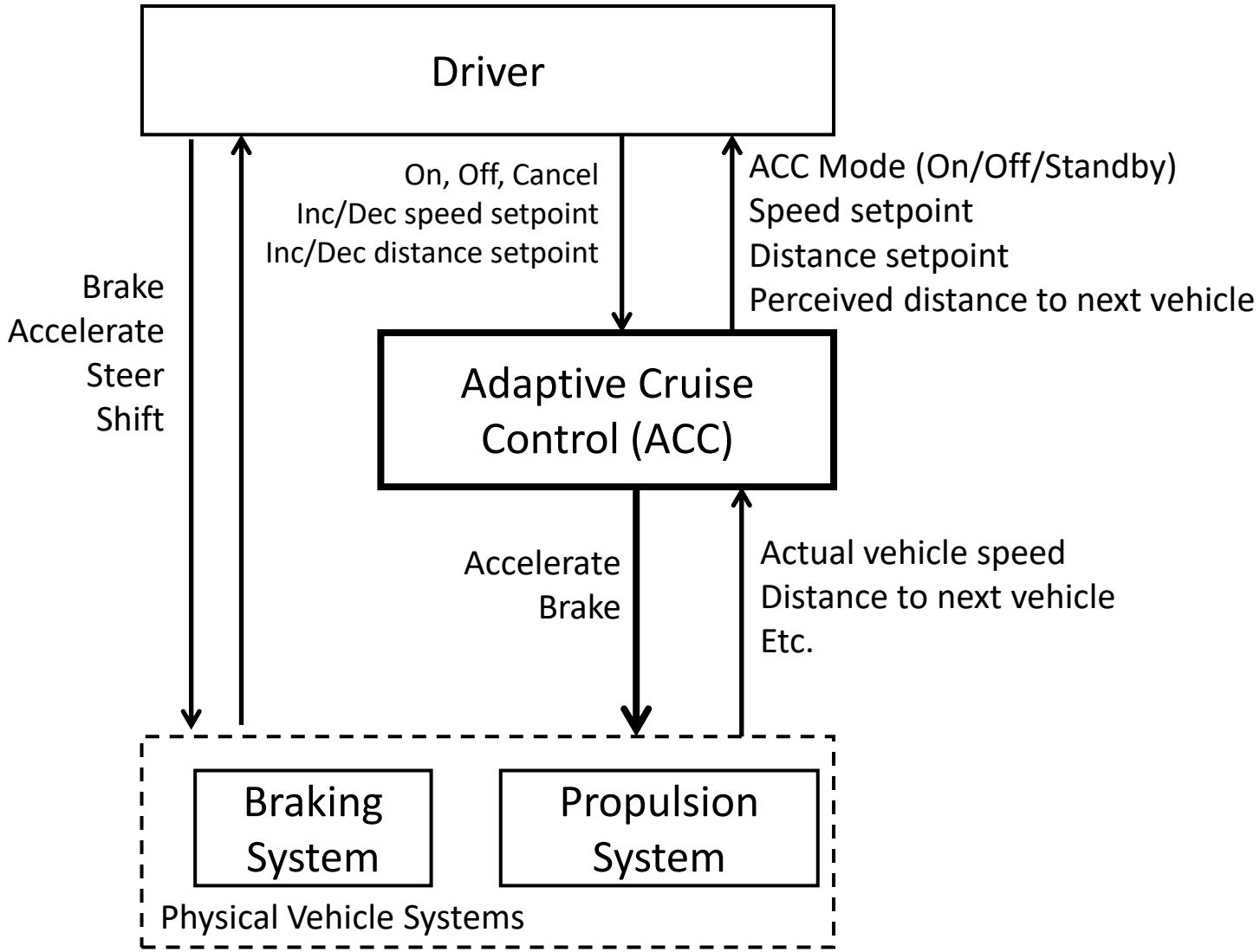
© Copyright 2026 John Thomas

Control structure



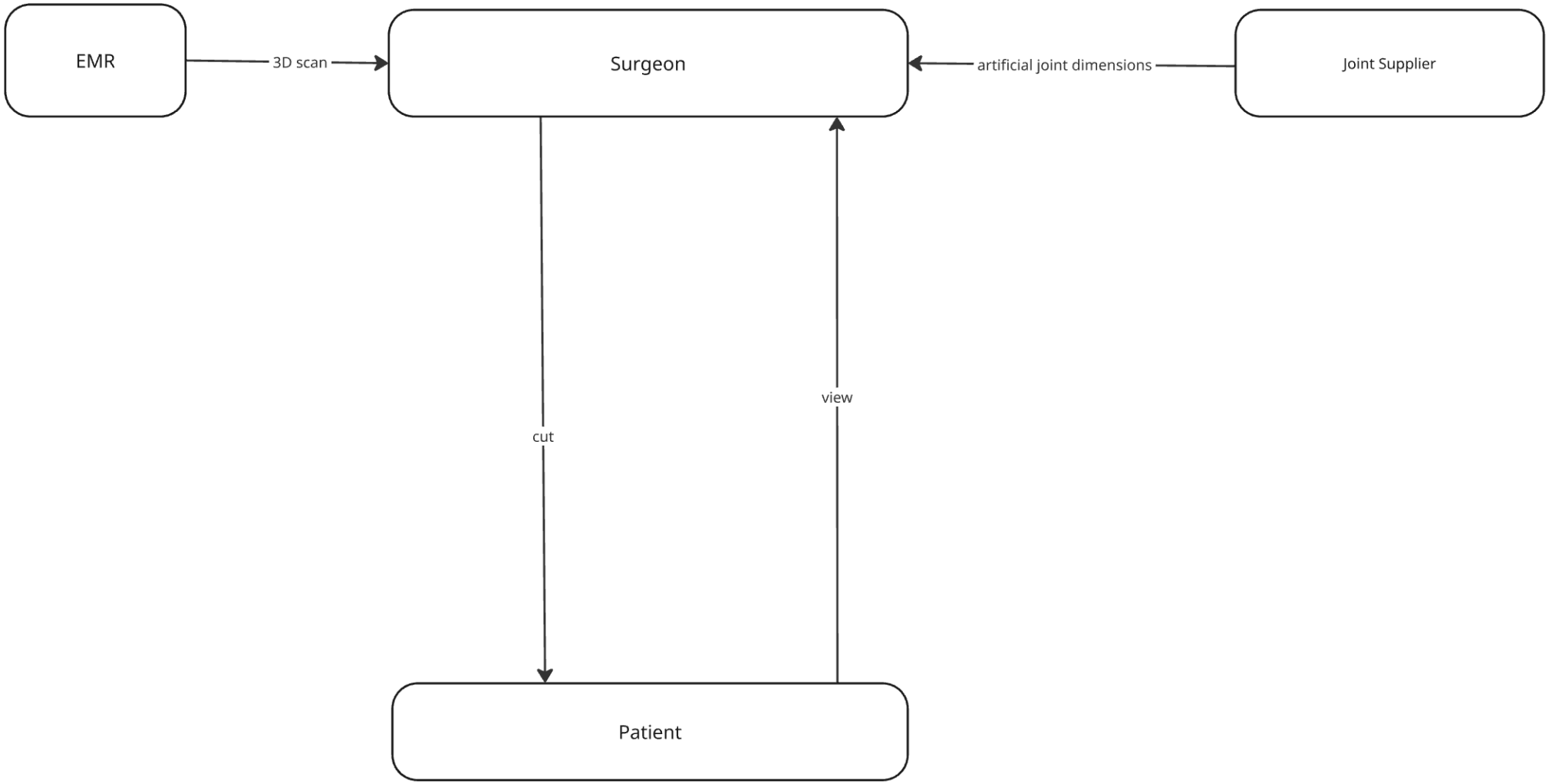
© Copyright 2026 John Thomas

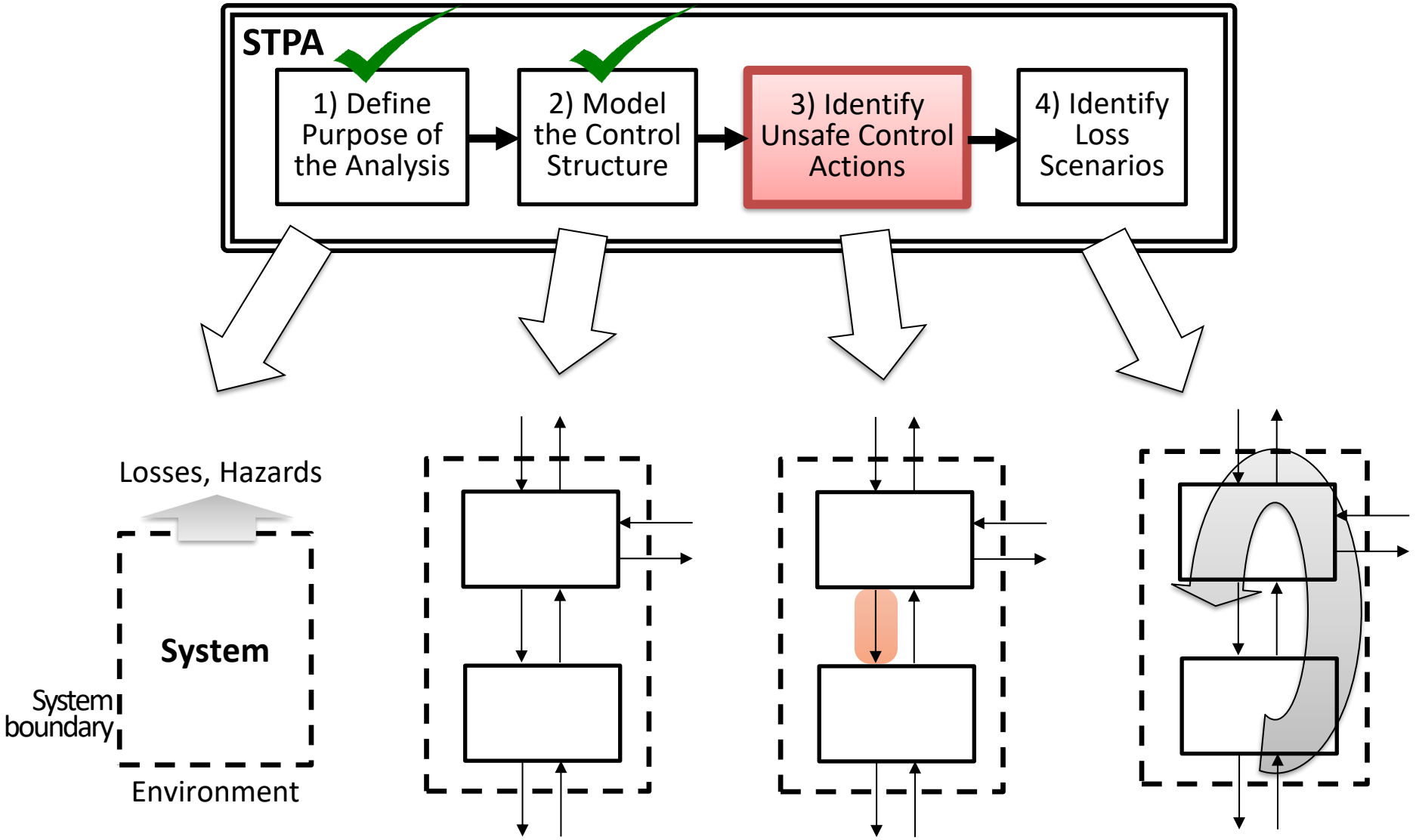
Example Control structure: Adaptive Cruise Control (ACC)



Thomas, 2017

High-level control structure: Without Robot

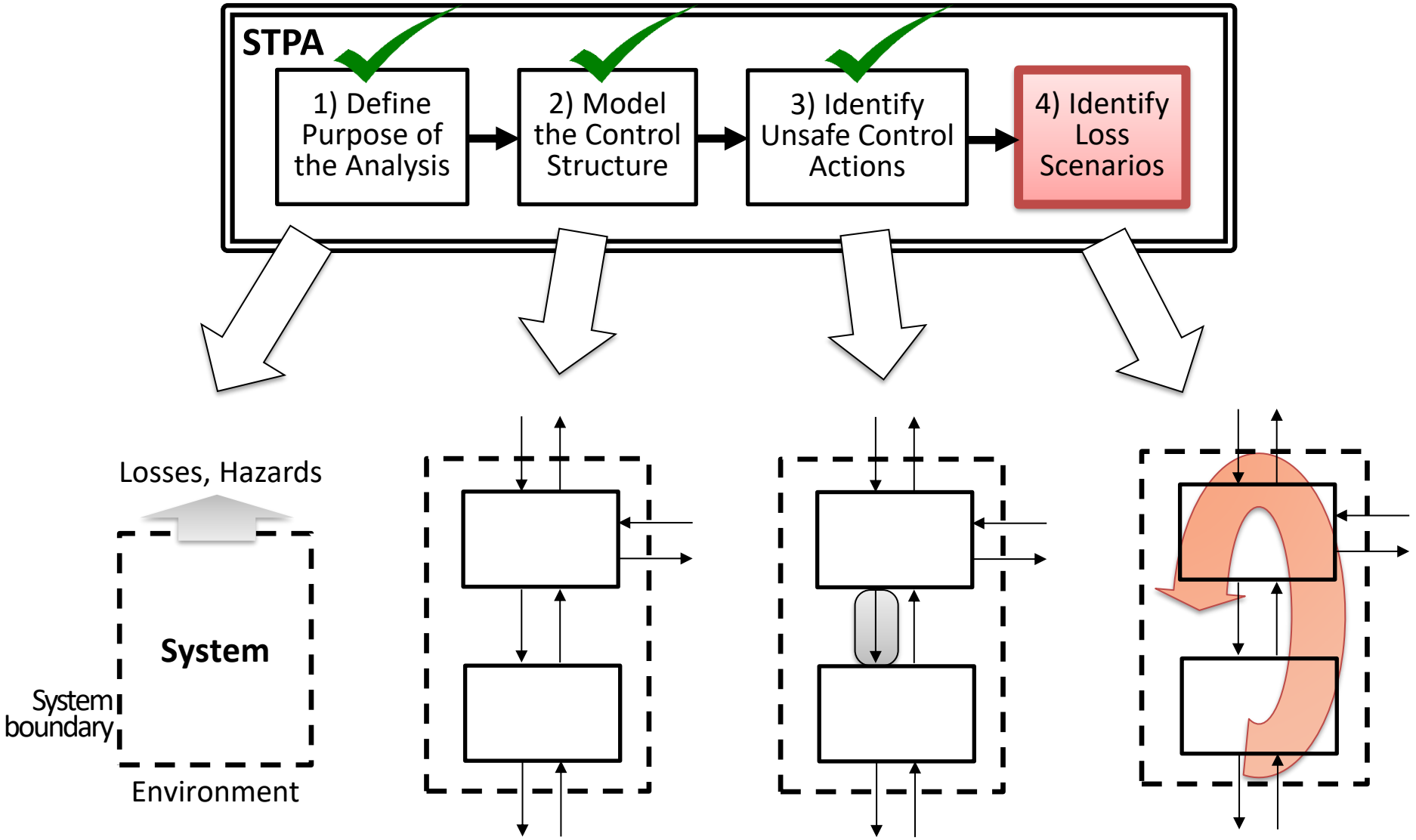




(Leveson and Thomas, 2018)

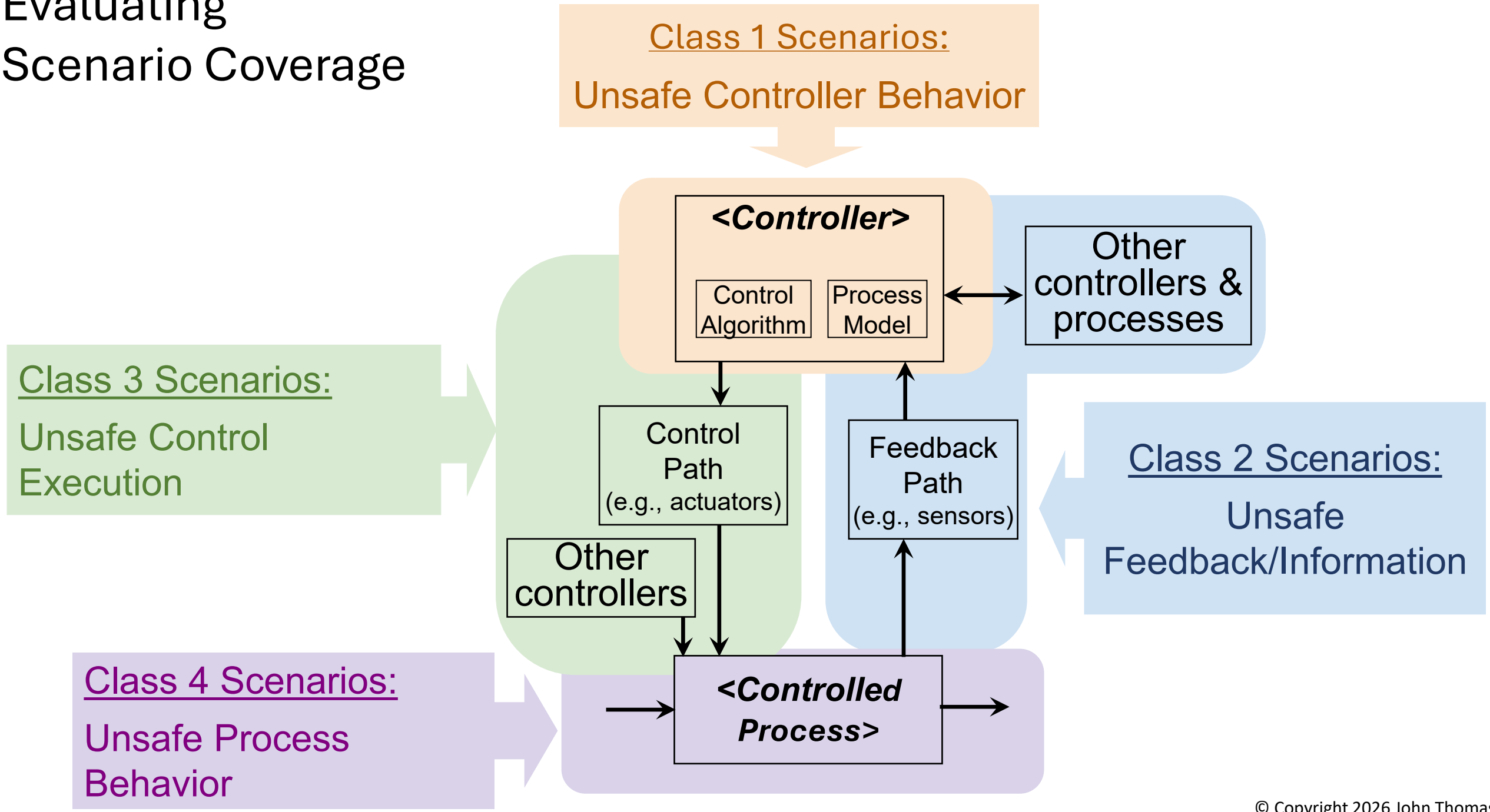
Step 3: Unsafe Control Actions (UCAs)

Control Action	Not Providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, Applied to long
cut	[UCA1] The Surgeon does not provide the cut command when Patient's knee is exposed. [H1]	[UCA2] The Surgeon provides cut command when scan data is misconfigured. [H1,H2] [UCA3] The Surgeon provides cut command when artificial joint dimensions are incorrect. [H1,H2] [UCA4] The Surgeon provides cut command when at incorrect cut plane. [H1,H2,H3]	[UCA5] The Surgeon provides the cut command too early, before arriving at the bone surface.[H2, H3, H4] [UCA6] The Surgeon provides cut command too late, after arriving at the bone surface. [H1,H2]	[UCA7] The Surgeon stops providing cut command before the bone slice is removed. [H1] [UCA8] The Surgeon continues providing cut command after the slice has been removed. [H3,H4]



(Leveson and Thomas, 2018)

Evaluating Scenario Coverage



© Copyright 2026 John Thomas

Scenario Coverage

Class 1 Scenario Archetype:
Unsafe Controller Behavior

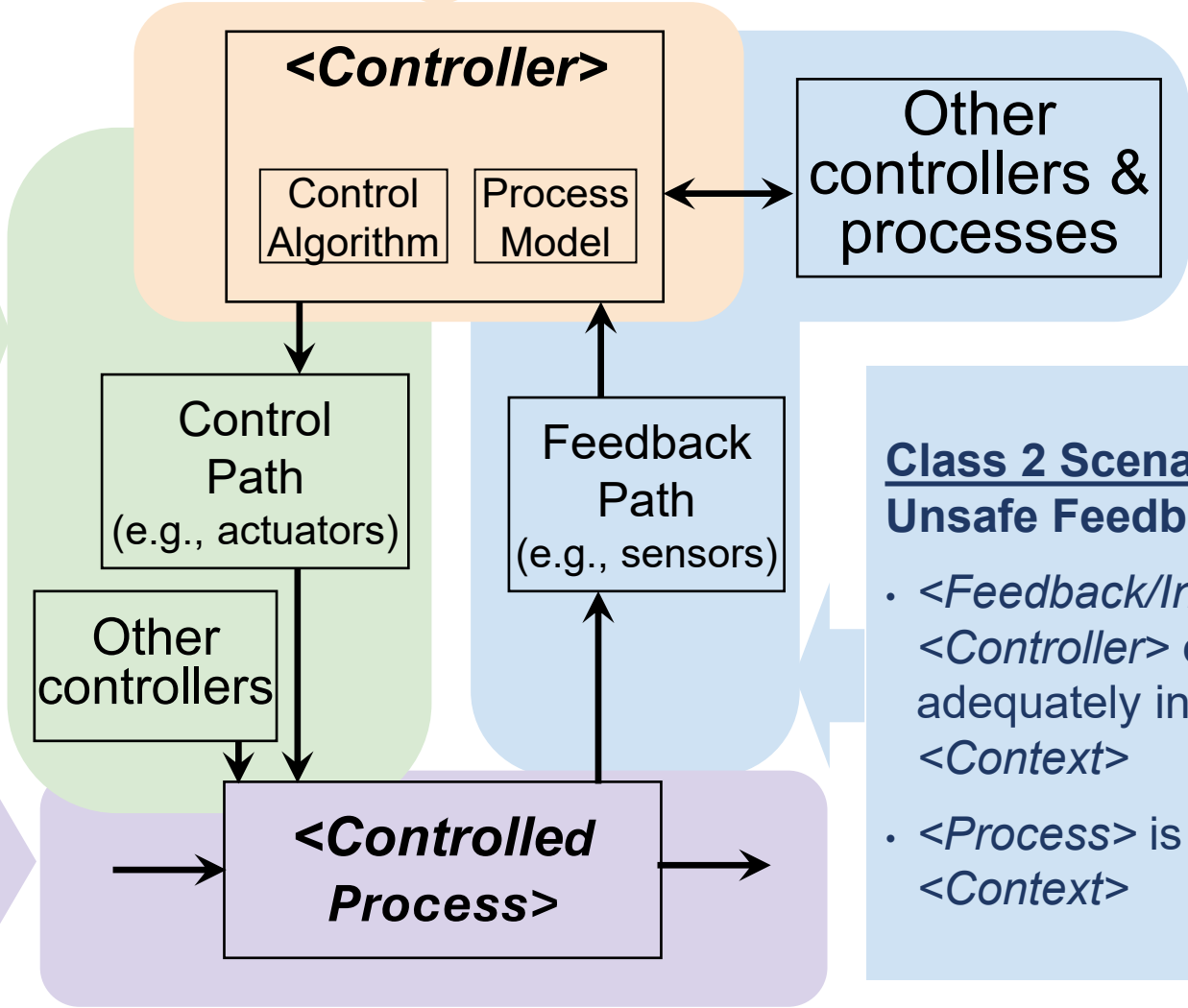
- UCA: *<Controller>* provides *<Control Action>* when *<Context>*
- *<Feedback/Input>* to *<Controller>* correctly indicates *<Context>*

Class 3 Scenario Archetype:
Unsafe Control Execution

- *<Controller>* does not provide *<Control Action>* when *<Context>*
- *<Process>* receives *<Control Action>* when *<Context>*

Class 4 Scenario Archetype:
Unsafe Process Behavior

- *<Process>* does not receive *<Control Action>* when *<Context>*
- *<Process>* effectively applies *<Control Action>* when *<Context>*



Class 2 Scenario Archetype:
Unsafe Feedback/Information

- *<Feedback/Input>* to *<Controller>* does not adequately indicate *<Context>*
- *<Process>* is actually *<Context>*

© Copyright 2026 John Thomas

Step 4: Scenarios

Scenario Template:

Unsafe Control Action	Scenario Class 1: Unsafe Controller Behavior	Scenario Class 2: Unsafe Feedback Path	Scenario Class 3: Unsafe Control Path	Scenario Class 4: Unsafe Controlled Process Behavior
UCA: <controller> does not provide <<cmd> when <context>	1) <controller> doesn't provide <cmd> when <context>, and 2) <controller> received feedback (or other input) that indicates <context>	1) feedback (or other input) received by <controller> does not adequately indicate <context>, and 2) <context> is true	1) <controller> does provide <cmd> when <context>, and 2) <cmd> is not received by <controlled process> when <context>	1) <cmd> is received by <controlled process> when <context>, and 2) <controlled process> does not respond by <...>

Robot Example:

Unsafe Control Action	Scenario 1: Unsafe Controller Behavior	Scenario 2: Unsafe Feedback Path	Scenario 3: Unsafe Control Path	Scenario 4: Unsafe Controlled Process Behavior
UCA1: The Surgeon does not provide the cut command when Patient knee is exposed.	The Surgeon doesn't provide cut when Patient's knee is exposed, and The Surgeon received feedback that indicates Patient's knee is exposed.	Feedback received by Surgeon does not adequately indicate that the Patient knee is exposed. Patient knee is exposed.	The Surgeon does provide cut when Patient's knee is exposed. Cut is not received by Patient when Patient's knee is exposed.	Cut is received by Patient when Patient's knee is exposed. Patient does not respond by ....

Other medical issues become apparent

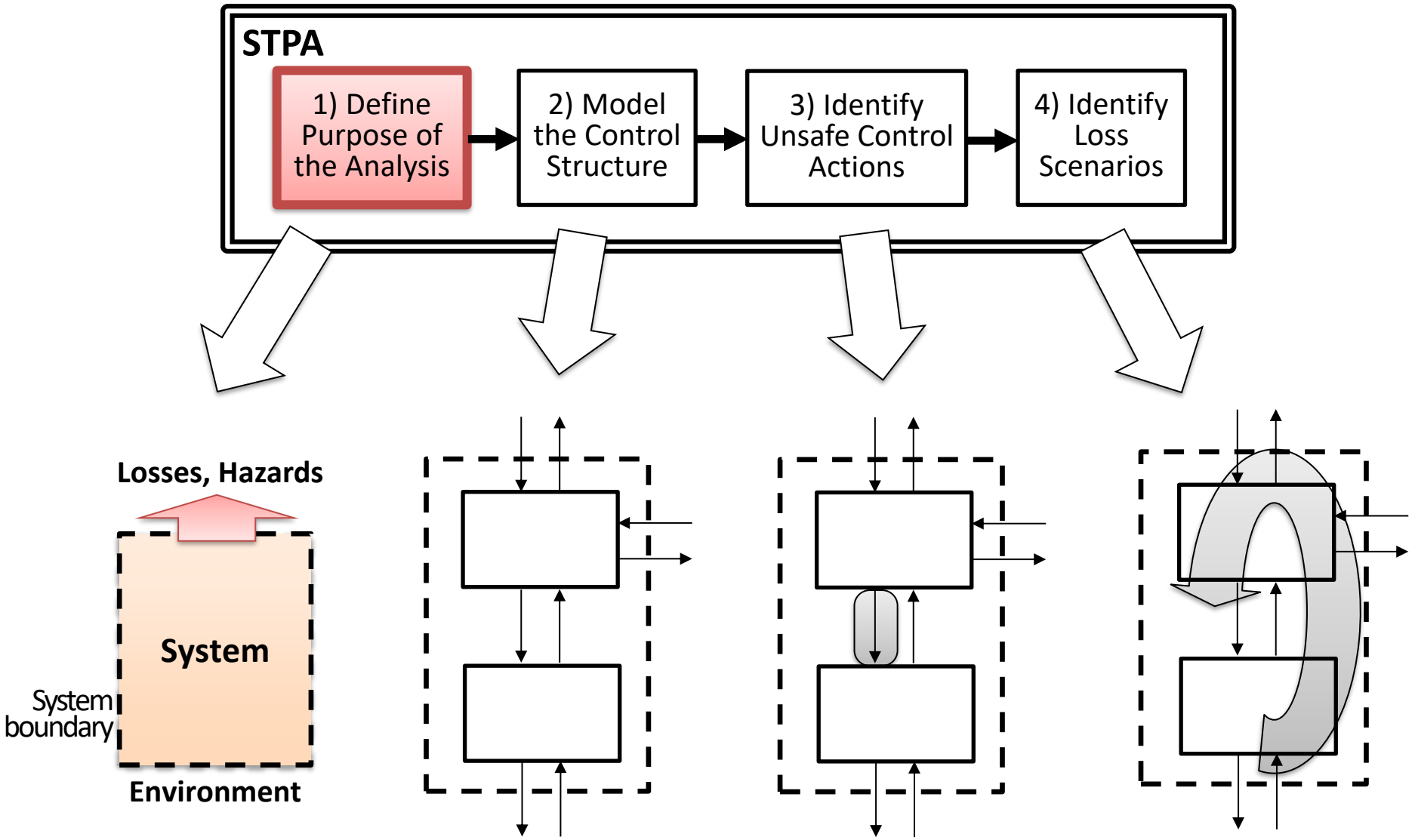
Possible cause: Patient's knee has moved

We'll handle the causes next!

Unsafe Control Action	Scenario 1: Unsafe Controller Behavior	Scenario 2: Unsafe Feedback Path	Scenario 3: Unsafe Control Path	Scenario 4: Unsafe Controlled Process Behavior	System Requirement
[UCA1] The Surgeon does not provide the cut command when Patient knee is exposed.	The Surgeon doesn't provide cut when Patient's knee is exposed. The Surgeon received feedback that indicates Patient's knee is exposed. 1.Other medical issues become apparent	Feedback received by Surgeon does not adequately indicate that the Patient knee is exposed. Patient knee is exposed.	The Surgeon does provide cut when Patient's knee is exposed. Cut is not received by Patient when Patient's knee is exposed. 1.Patients knee has moved	Cut is received by Patient when Patient's knee is exposed. Patient does not respond by	The System shall only cut if Surgeon allows it. The System shall track the Patient's tibia and femur positions.
[UCA2] The Surgeon provides cut command when scan data is incorrect.	The Surgeon provides cut when scan data is incorrect. The Surgeon received feedback (or other inputs) that indicates scan data is incorrect.	Feedback (or other inputs) received by the Surgeon does not adequately indicate scan data is incorrect. Scan data is incorrect. 1.old scan chosen 2.incorrect patient choosen	The Surgeon does not provide cut when scan data is incorrect. Patient receives cut when scan data is incorrect.	Cut is not received by Patient when scan data is incorrect. Patient responds by ...	The System shall require Surgeon confirmation if a scan older than x months is loaded. The System shall prohibit cutting if scan data does not match with Patient.
[UCA3] The Surgeon provides cut command when artificial joint dimensions are incorrect.	The Surgeon provides cut when artificial joint dimensions are incorrect. The Surgeon received feedback (or other inputs) that indicates artificial joint dimensions are incorrect. 1.input wrong artificial joint dimensions	Feedback (or other inputs) received by the Surgeon does not adequately indicate artificial joint dimensions are incorrect. Artificial joint dimensions are incorrect. 1.Incorrect data sheet used	The Surgeon does not provide cut when artificial joint dimensions are incorrect. Patient receives cut when artificial joint dimensions are incorrect.	Cut is not received by Patient when artificial joint dimensions are incorrect. Patient responds by ...	The System shall provide a means to load 3d models of artificial joints. The System shall overlay artificial joint models with 3D scans.
[UCA4] The Surgeon provides cut command when at incorrect cut plane.	The Surgeon provides cut when at incorrect cut plane. The Surgeon received feedback (or other inputs) that indicates cut plane is incorrect	Feedback (or other inputs) received by the Surgeon does not adequately indicate when at incorrect cut plane. Cut plane is incorrect.	The Surgeon does not provide cut when at incorrect cut plane. Patient receives cut when at incorrect cut plane.	Cut is not received by Patient when at incorrect cut plane. Patient responds by ...	The System shall constrain movement to cut plane. The System shall determine the cut plane based on

Now: **With** Robot

STPA



(Leveson and Thomas, 2018)

Step 1: Losses and Hazards

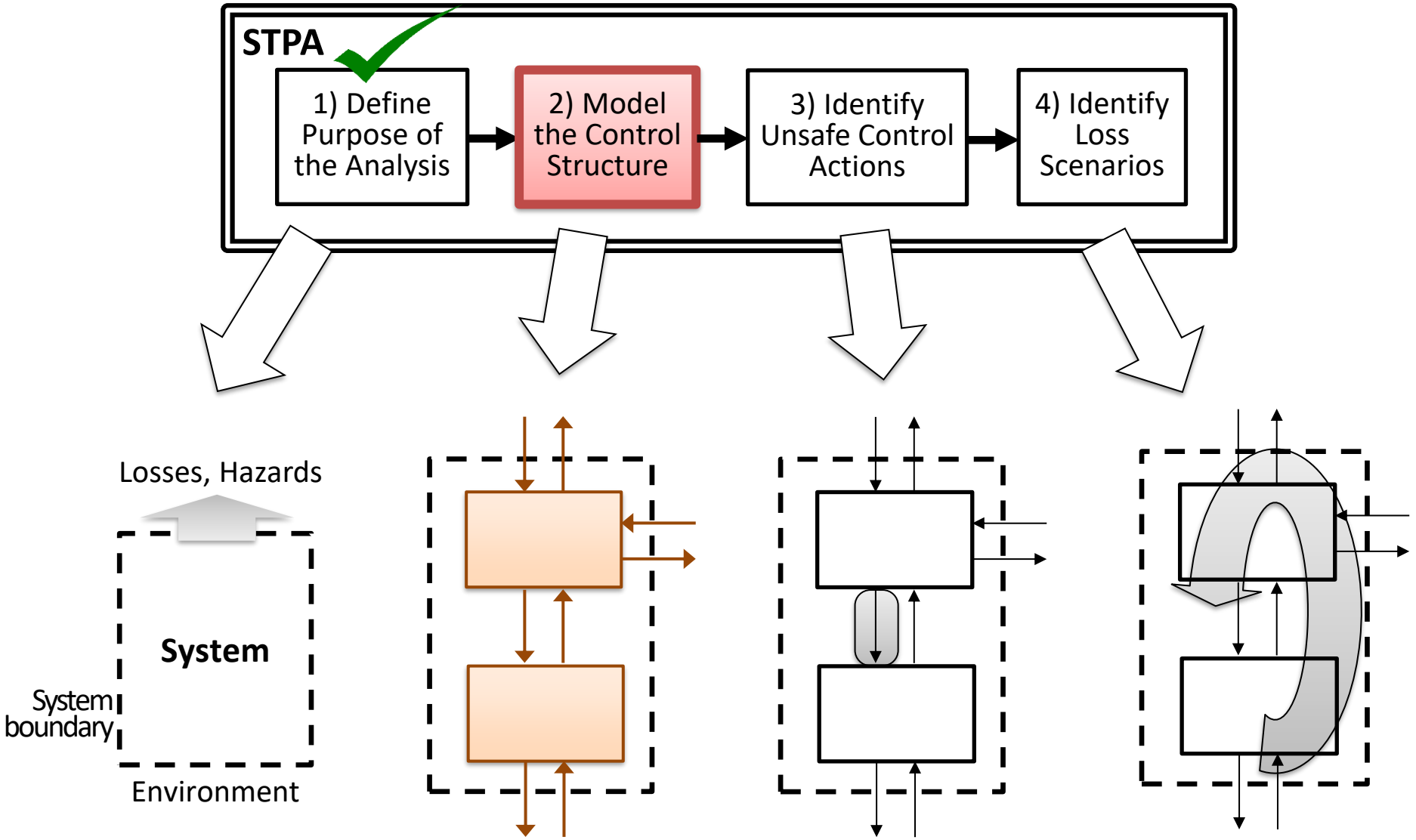
Losses

- L-1: Loss of life or injury to patient or user.
- L-2: Loss of implanting artificial joint.
- L-3: Loss of patient information.

Hazards

- H-1: The Robot injures the patient [L-1]
  - H-1.1: The Patient's legs result in being of different lengths [L-1]
  - H-1.2: The Patient's tendons or nerves fibers are damaged [L-1]
- H-2 The Robot injures the Surgeon or staff [L-1]
- H-3 The Robot is unable to perform surgical procedures [L-2]
  - [H-3.1] Surgical procedure delay
  - [H-3.2] More Complex Surgery
  - [H-3.3] Additional Surgery
  - [H-3.4] Cancelled or Aborted Surgical procedure

© Copyright 2026 John Thomas

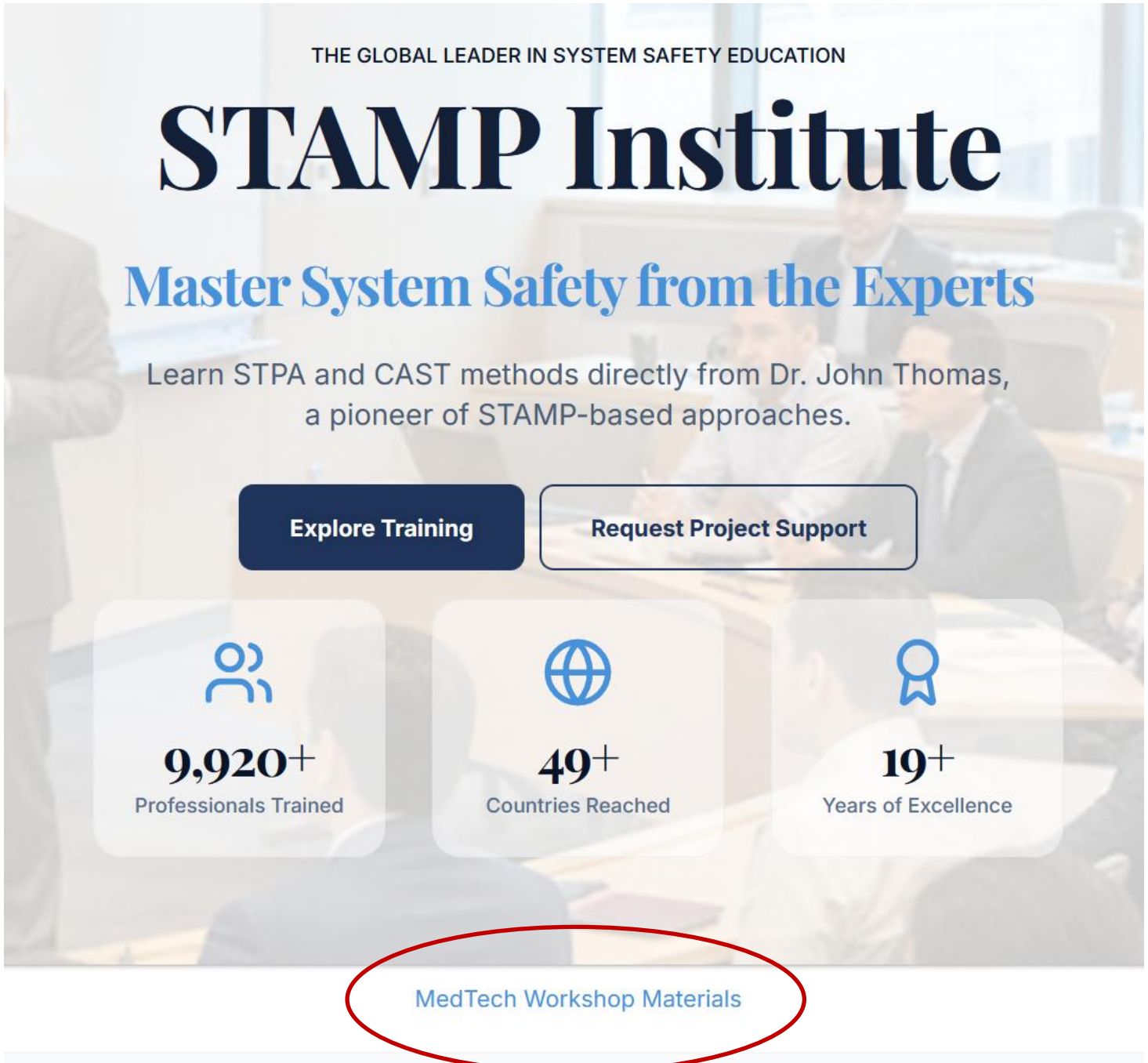


(Leveson and Thomas, 2018)

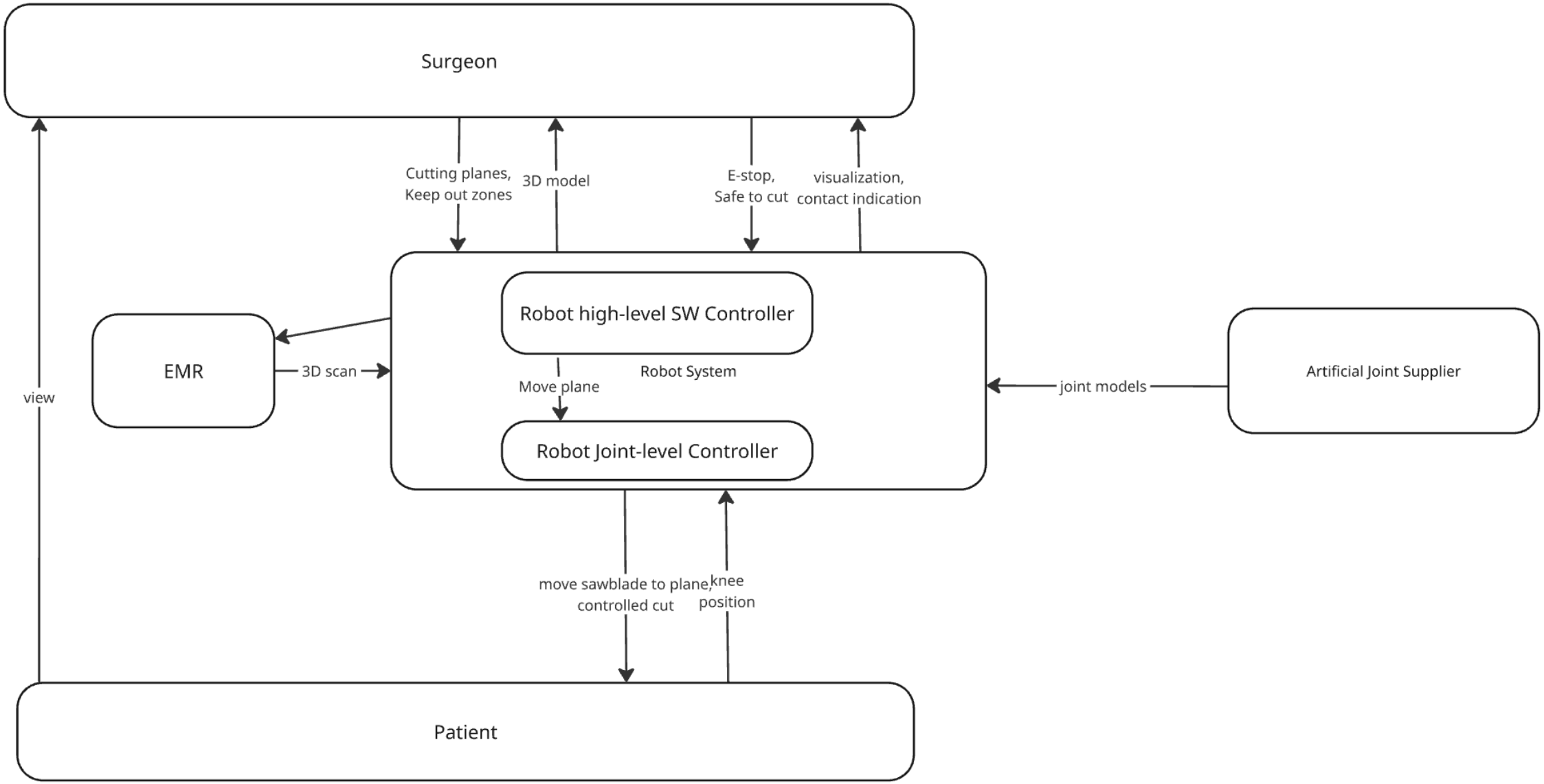
Open the Breakout Session Files

Start “Breakout 1”

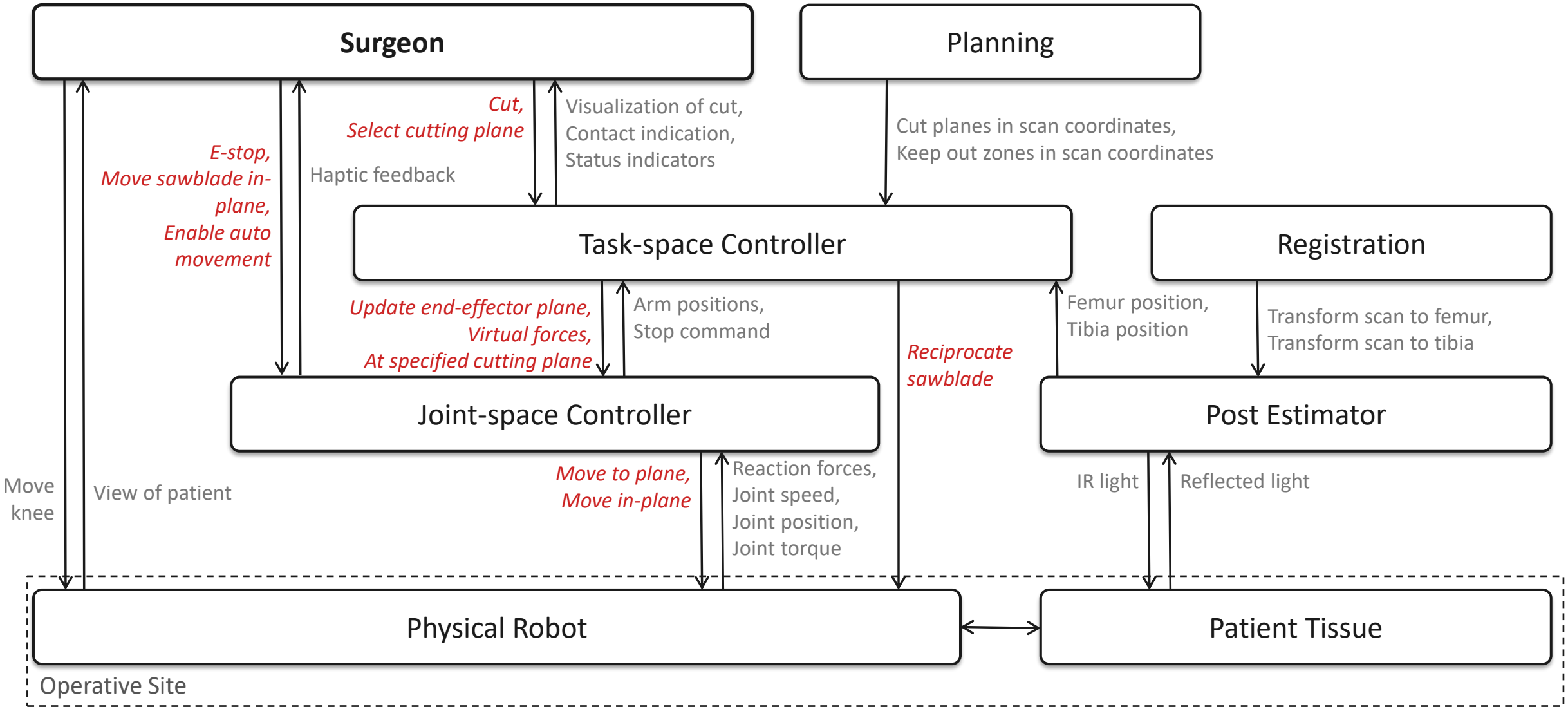
Go to: **STAMP-institute.com**



High-level control structure: With robot

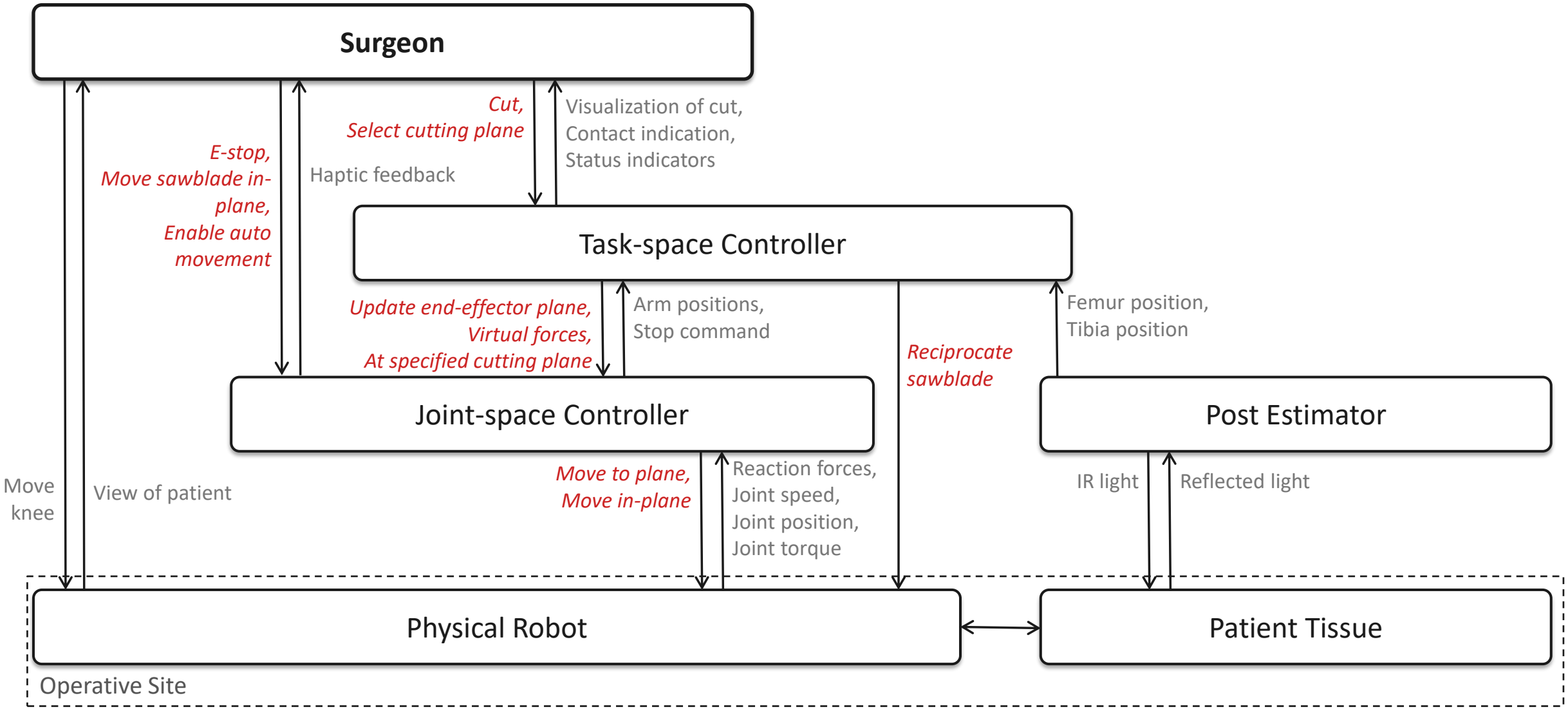


Robot Control Structure (with planning)

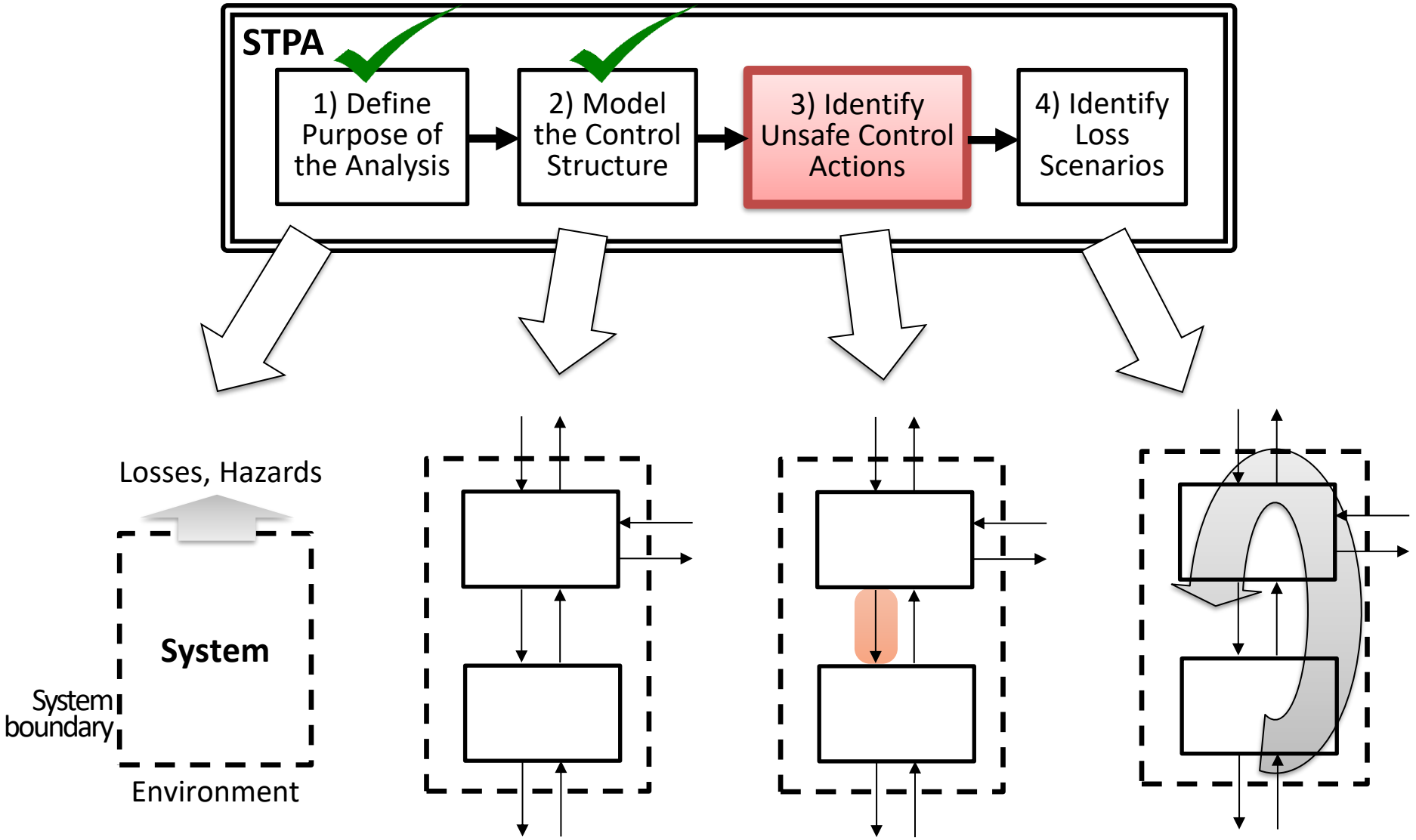


© Copyright 2026 John Thomas

Robot Control Structure

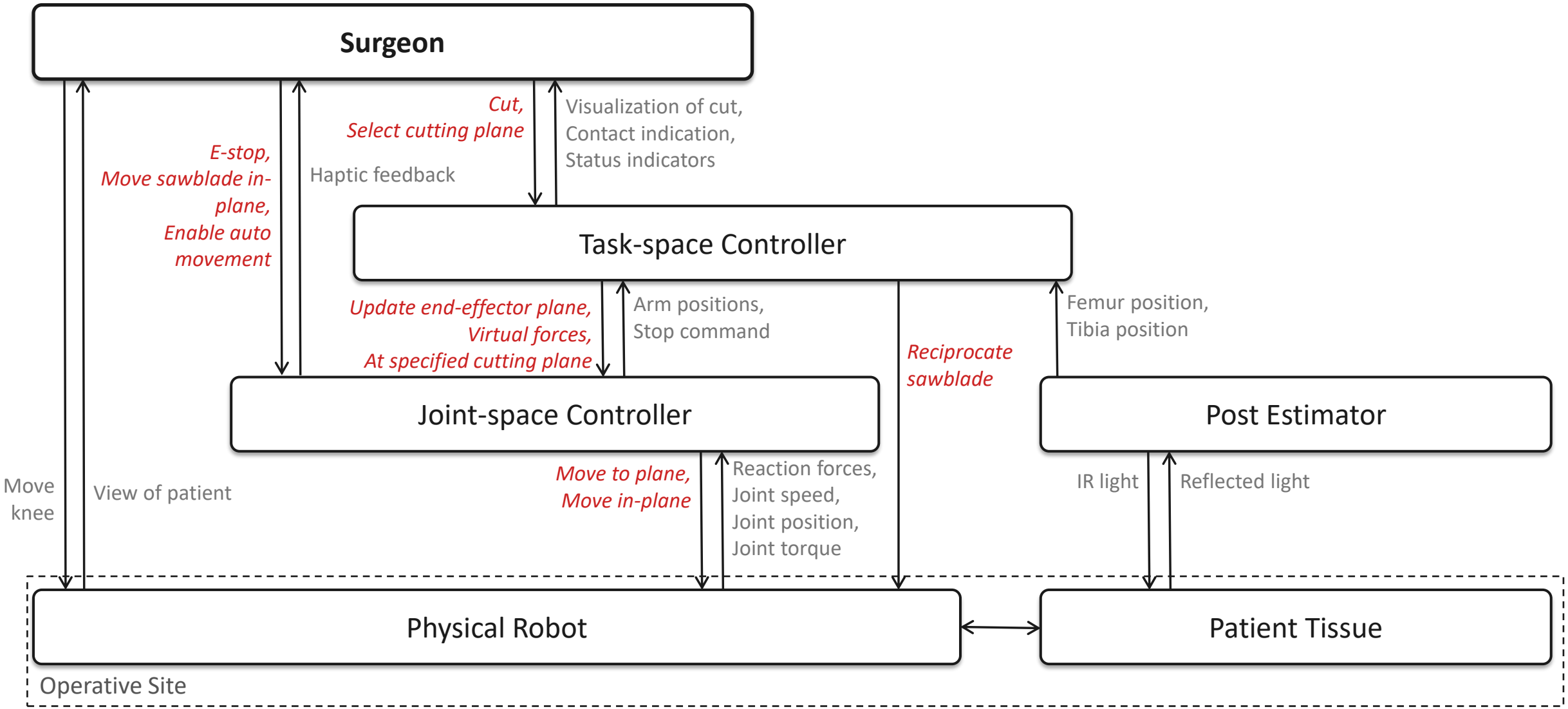


© Copyright 2026 John Thomas



(Leveson and Thomas, 2018)

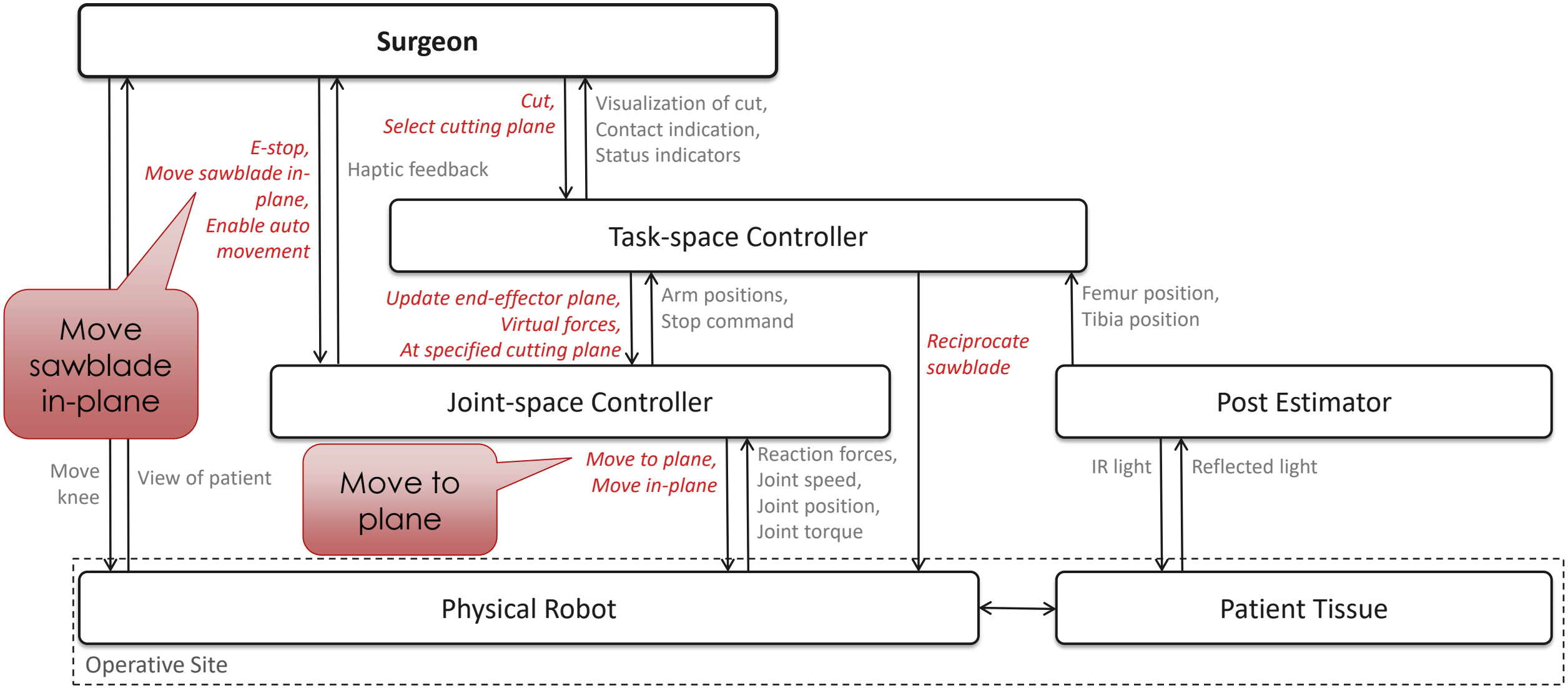
Robot Control Structure



© Copyright 2026 John Thomas

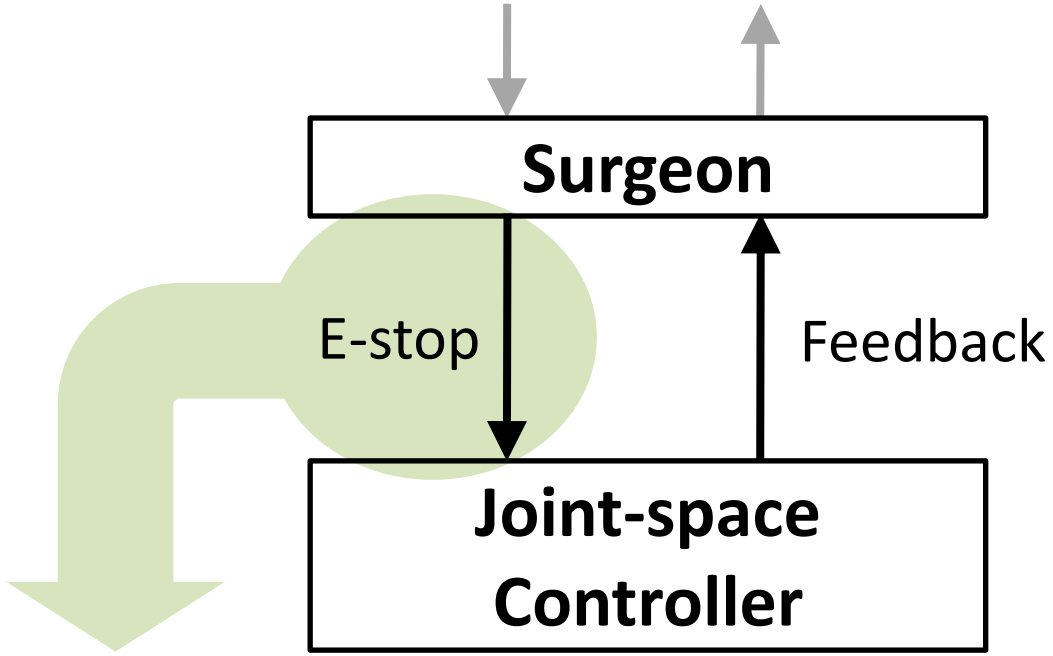
Robot Control Structure

Today's focus areas



© Copyright 2026 John Thomas

Structure for Unsafe Control Actions



UCA Example:

"Surgeon provides *E-stop* command when there is no urgent emergency condition

- Surgeon: Source Controller
- provides: Type
- *E-stop* command: Control Action
- there is no urgent emergency condition: Context

© Copyright 2026 John Thomas

Step 3: Unsafe Control Actions (UCAs)

Control Action	Not Providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, Applied to long
e-stop	The Surgeon does not provide *e-stop* command when robot performs uncommanded motion. [H-1.1, H-1.2]	The Surgeon provides *e-stop* command when sawblade is inside bone. [H-3] The Surgeon provides *e-stop* command when there is no urgent emergency condition [H-3]	Surgeon provides *e-stop* too late, after patient or user is harmed by the device movement. [H-1.1, H-1.2, H-2]	Surgeon provides *e-stop* too long, after ready to resume surgery. [H-3] Surgeon stops providing *e-stop* too soon before the underlying event is resolved. [H-1.1, H-1.2, H-2]

H-1: The Robot injures the patient [L-1]
- H-1.1: The Patient's legs result in being of different lengths [L-1]
- H-1.2: The Patient's tendons or nerves fibers are damaged [L-1]

H-2 The Robot injures the Surgeon or staff [L-1]
H-3 The Robot is unable to perform surgical procedures [L-2]

Example UCA Syntax

	Not providing causes hazard	Providing causes hazard	Too Early, Too Late, Out of Order	Stopped Too Soon / Applied too long
<control action>	<controller> does not provide <control action> when <context> [<link to hazards>]	<controller> provides <control action> when <context> [<link to hazards>]	<controller> provides <control action> too late after (>TBD s after) <context> [<link to hazards>] <controller> provides <control action> too early before (>TBD s before) <context> [<link to hazards>]	<controller> stops providing <control action> too soon before (>TBD s before) <context> [<link to hazards>] <controller> continues providing <control action> too long after (>TBD s after) <context> [<link to hazards>]

(Thomas, 2018)

Note: This is one possible syntax, not a rule.
See the STPA Handbook for UCA definition and criteria.

© Copyright 2026 John Thomas

Breakout Exercise!

Step 3: Unsafe Control Actions (UCAs)

Control Action	Not Providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, Applied to long
move sawblade in-plane	The Surgeon does not provide *move sawblade in-plane* while at cutting plane. [H-3]	The Surgeon provides *move sawblade in-plane* while pushing into bone but not cutting. [H-1.1, H-1.2] The Surgeon provides *move sawblade in-plane* with a forces that overload the robotic arm. [H-1.1, H-1.2, H-2, H-3] The Surgeon provides move sawblade in-plane with force lower than the arm threshold force. [H-3]	The Surgeon provides move sawblade in-plane too early, before the sawblade is at the cutting planes. [H-1.1, H-1.2, H-2]	The Surgeon provides *move sawblade in-plane* too long, after the sawblade enters other tissue. [H-1.1, H-1.2]

H-1: The Robot injures the patient [L-1]
- H-1.1: The Patient's legs result in being of different lengths [L-1]
- H-1.2: The Patient's tendons or nerves fibers are damaged [L-1]

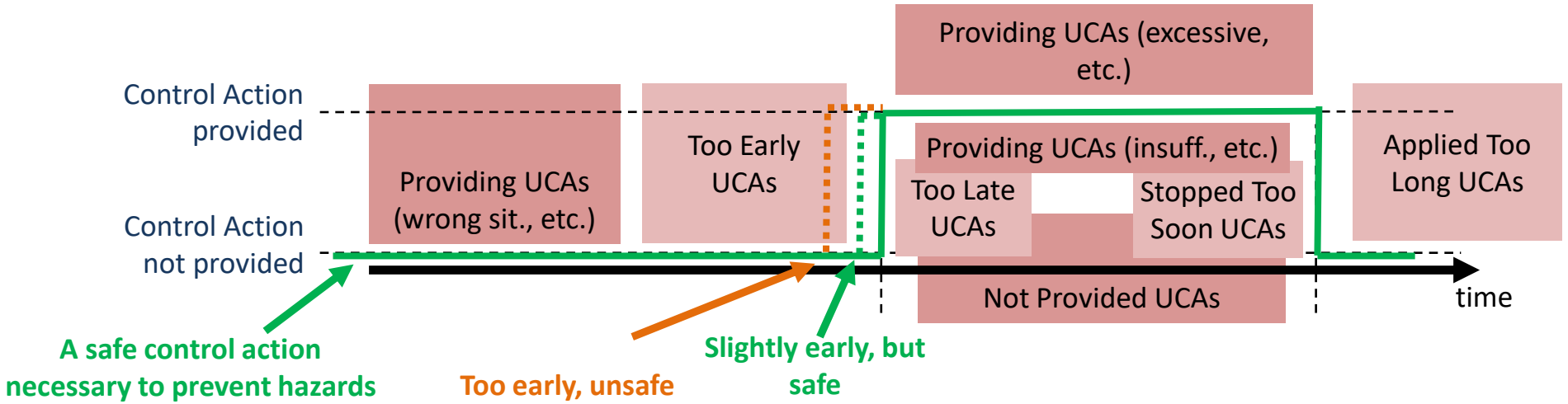
H-2 The Robot injures the Surgeon or staff [L-1]
H-3 The Robot is unable to perform surgical procedures [L-2]

Step 3: Unsafe Control Actions (UCAs)

Control Action	Not Providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, Applied to long
move to plane	The Joint-space Controller does not provide *move to plane* while Surgeon is directing it to. [H-3]	The Joint-space Controller provides *move to plane* too quickly for humans to avoid. [H-1.1, H-1.2, H-2]	The Joint-space Controller provides move to plane too early, before the current cut is complete. [H-1.1, H-1.2] The Joint-space Controller provides move to plane to early, before the sawblade is out of the bone. [H-1.1, H-1.2] The Joint-space Controller provides move to plane too early, before the Surgeon has authorized it. [H-1.1, H-1.2, H-2]	The Joint-space Controller provides *move to plane* too long [1 second], after the Surgeon has stopped authorizing it. [H-1.1, H-1.2] The Joint-space Controller provides move to plane too long, while the sawblade is already within the tolerances of 2 mm and 2 degree. [H-1.1, H-1.2]

H-1: The Robot injures the patient [L-1]
  H-1.1: The Patient's legs result in being of different lengths [L-1]
  H-1.2: The Patient's tendons or nerves fibers are damaged [L-1]
H-2 The Robot injures the Surgeon or staff [L-1]
H-3 The Robot is unable to perform surgical procedures [L-2]

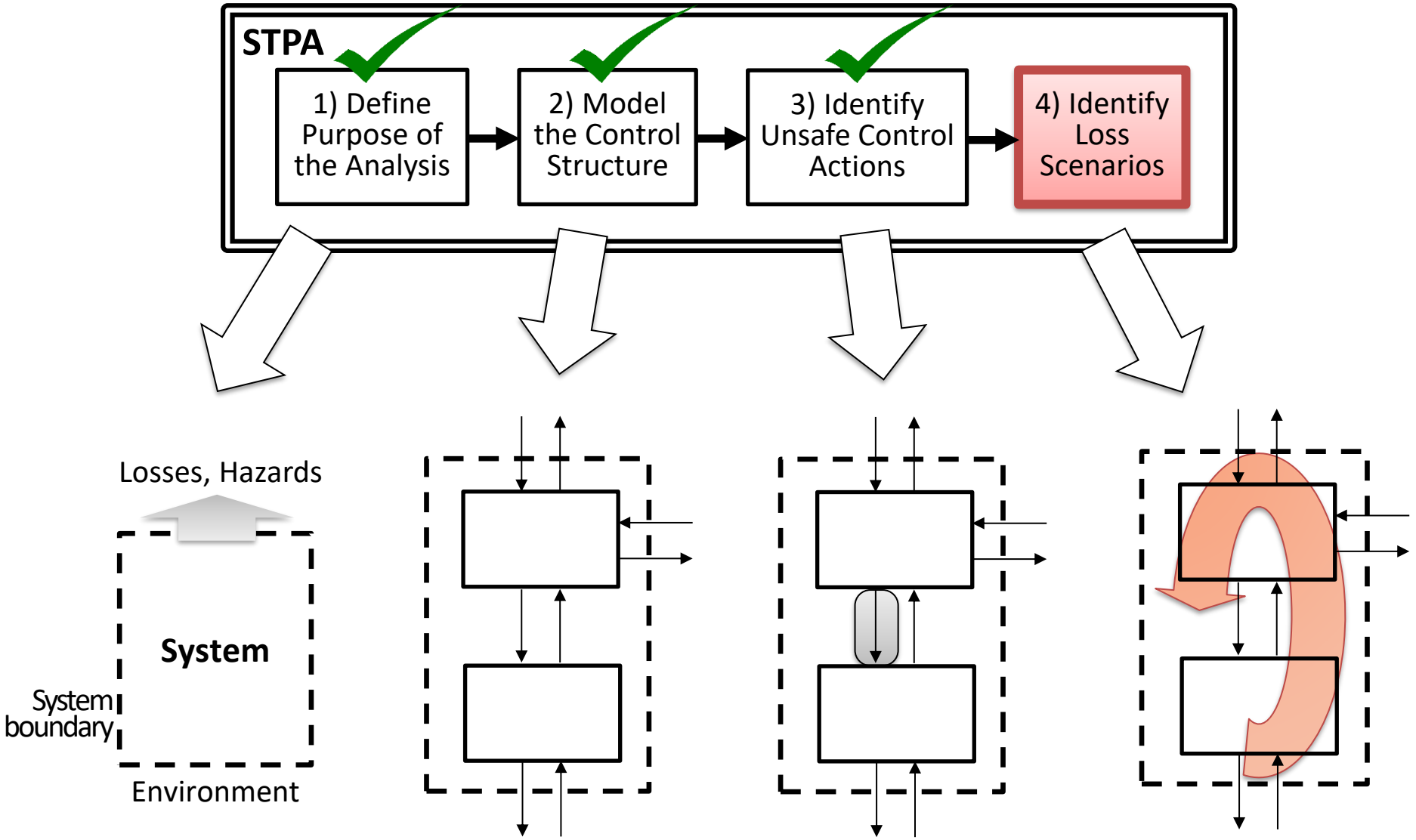
STPA UCA Bounding



	Not providing causes hazard	Providing causes hazard *[in wrong situation, excessive, insufficient, repetitive, wrong direction, etc.]*	Too early, too late, out of order	Stopped Too Soon / Applied too long
Control Actions	?	?	?	?

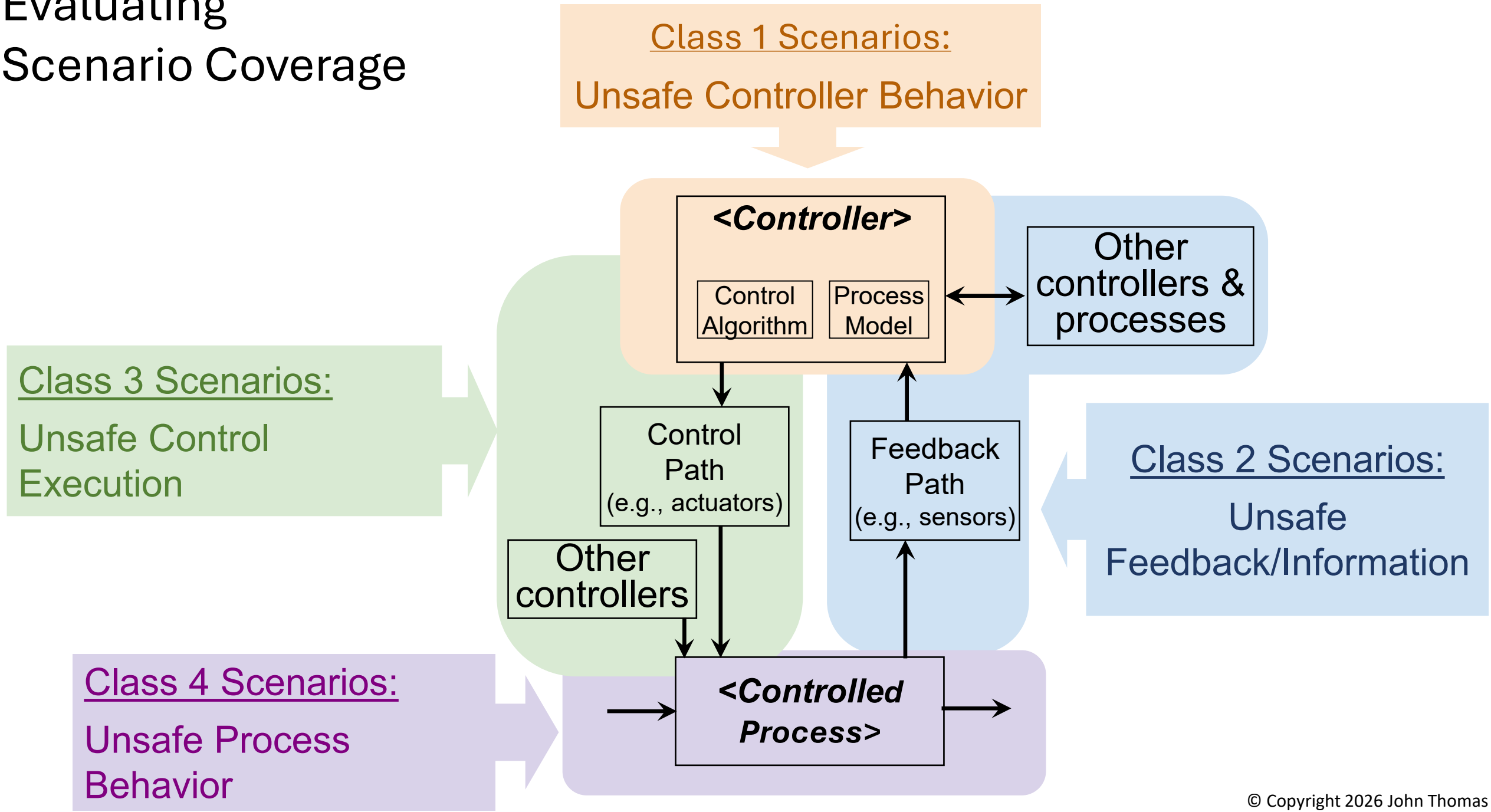
The complete set of UCAs will fully bound the necessary safe behavior

© Copyright 2025 John Thomas



(Leveson and Thomas, 2018)

Evaluating Scenario Coverage



© Copyright 2026 John Thomas

STPA Scenario Classes

UCA:
<Controller> does not provide *<Control Action>* while *<Context>*

Class 1 Scenario:
Unsafe Controller Behavior
- *<Controller>* does not provide *<Control Action>* while *<Context>, and*
- *<Controller>* received feedback correctly indicating *<Context>*

Class 2 Scenario:
Unsafe Feedback Path
- *Feedback received by <Controller>* does not adequately indicate *<Context>, and*
- *<Context> is true*

Class 3 Scenario:
Unsafe Control Path
- *<Controller>* provides *<Control Action>* while *<Context>, and*
- *<Process>* does not receive *<Control Action>* while *<Context>*

Class 4 Scenario:
Unsafe Controlled Process Behavior
- *<Process>* receives *<Control Action>* while *<Context>, and*
- *<Process>* does not respond by <Control Action Response>

UCA:

Joint-Space Controller does not provide move to plane when Surgeon is requesting motion

Controller = Joint-Space Controller

Control Action = move to plane

Context = Surgeon is requesting motion

© Copyright 2026 John Thomas

STPA Scenario Classes

UCA:

Joint-space Controller does not provide **move to plane** while **Surgeon is requesting motion**

Class 1 Scenario:
Unsafe Controller Behavior
- **Joint-space Controller** does not provide **move to plane** while **Surgeon is requesting motion**, and
- **Joint-space Controller** received feedback correctly indicating **Surgeon is requesting motion**

Class 2 Scenario:
Unsafe Feedback Path
- Feedback received by **Joint-space Controller** does not adequately indicate **Surgeon is requesting motion**, and
- **Surgeon is requesting motion** is true

Class 3 Scenario:
Unsafe Control Path
- **Joint-space Controller** provides **move to plane** while **Surgeon is requesting motion**, and
- **Robot arm** does not receive **move to plane** while **Surgeon is requesting motion**

Class 4 Scenario:
Unsafe Controlled Process Behavior
- **Robot arm** receives **move to plane** while **Surgeon is requesting motion**, and
- **Robot arm** does not respond by **moving to cutting plane**

UCA:

Joint-Space Controller does not provide move to plane when Surgeon is requesting motion

Controller = Joint-Space Controller

Control Action = move to plane

Context = Surgeon is requesting motion

© Copyright 2026 John Thomas

Step 4: Scenarios (Example)

Unsafe Control Action	Scenario Class 1: Unsafe Controller Behavior	Scenario Class 2: Unsafe Feedback Path	Scenario Class 3: Unsafe Control Path	Scenario Class 4: Unsafe Controlled Process Behavior
The Joint-space Controller does not provide *move to plane* when Surgeon is requesting motion.	The Joint-space Controller does not provide *move to plane* when Surgeon is requesting motion. The Joint-space Controller receives *enable automatic movement* correctly indicates Surgeon is requesting motion. **Refined Scenarios:** • Joint sensor outputs have drifted over time and it thinks it is at the desired plane • Controller has done the inverse kinematics incorrectly; hence, the motion will be in the wrong plane • Solving the inverse kinematics takes too long, timeout hit without arriving at a solution • The desired motion is outside of the workspace of the arm **Solutions / Recommendations / Mitigations:** • Use absolute position sensors on joints • Notify user if in safe state • Check for error by feeding results from inverse kinematics into forward kinematics • Increase the workspace (link lengths) of the arm	*Enable automatic motion* received by Joint-space Controller does not adequately indicate Surgeon is demanding motion. Surgeon is requesting motion is true. **Refined Scenarios:** • Footpedal is jammed or unplugged • External noise interference with the signal **Solutions / Recommendations / Mitigations:** • Use safe-critical hardware • IEC 60601-1-2	The Joint-space Controller does provide *move to plane* when Surgeon is requesting motion. *Move to plane* commands is not received by *robot arm* when Surgeon is requesting motion. **Refined scenarios** • Safety controller has sent a STO (safe torque of) command to inhibit movement **Solutions / Recommendations / Mitigations:** • Notify user if arm is in a safe state	*Move to plane* is received by robot arm when Surgeon is requesting motion. Robot arm does not respond by moving to cutting plane. **Refined sceanrios** • The arm is in gimbal lock (singularity), so the command will not result in any movement • The arm is pushed up against object, and it can't move **Solutions / Recommendations / Mitigations:** • Notify user if in singularity • Notify user if joint torques have passed expected values

Let’s try another one!

Breakout 3: STPA Scenario Classes

UCA:
<Controller> does not provide *<Control Action>* while *<Context>*

Class 1 Scenario:
Unsafe Controller Behavior
- *<Controller>* does not provide *<Control Action>* while *<Context>, and*
- *<Controller>* received feedback correctly indicating *<Context>*

Class 2 Scenario:
Unsafe Feedback Path
- *Feedback received by <Controller>* does not adequately indicate *<Context>, and*
- *<Context> is true*

Class 3 Scenario:
Unsafe Control Path
- *<Controller>* provides *<Control Action>* while *<Context>, and*
- *<Process>* does not receive *<Control Action>* while *<Context>*

Class 4 Scenario:
Unsafe Controlled Process Behavior
- *<Process>* receives *<Control Action>* while *<Context>, and*
- *<Process>* does not respond by <Control Action Response>

UCA:

Surgeon does not provide move sawblade in-plane while at cutting plane

Controller = Surgeon

Control Action = move sawblade in-plane

Context = at cutting plane

Deliverable: Write the four scenario classes for our UCA above

© Copyright 2026 John Thomas

Breakout 3: STPA Scenario Classes

UCA:
<Controller> does not provide *<Control Action>* while *<Context>*

Class 1 Scenario:
Unsafe Controller Behavior
- **Surgeon** does not provide **move sawblade in-plane** while **at cutting plane**, and
- **Surgeon** received feedback correctly indicating **at cutting plane**

Class 2 Scenario:
Unsafe Feedback Path
- Feedback received by **Surgeon** does not adequately indicate **at cutting plane**, and
- **at cutting plane** is true

Class 3 Scenario:
Unsafe Control Path
- **Surgeon** provides **move sawblade in-plane** while **at cutting plane**, and
- **Joint-space Controller** does not receive **move sawblade in-plane** while **at cutting plane**

Class 4 Scenario:
Unsafe Controlled Process Behavior
- **Joint-space Controller** receives **move sawblade in-plane** while **at cutting plane**, and
- **Joint-space Controller** does not respond by **move in plane**

UCA:

Surgeon does not provide move sawblade in-plane while at cutting plane

Controller = Surgeon

Control Action = move sawblade in-plane

Context = at cutting plane

© Copyright 2026 John Thomas

Scenario Refinement

Breakout 4: Unsafe Controller Behavior

UCA: The Surgeon does not provide move sawblade in-plane while at cutting plane. [H1]

Class 1 Scenario:
Unsafe Controller Behavior
- *The Surgeon* does not provide *move sawblade in-plane* while at the cutting plane, and
- *The **visualization of cut** correctly shows that the sawblade is at the cutting plane.*

Class 2 Scenario

Class 3 Scenario

Class 4 Scenario

Refined Scenarios

Responsibilities
The Surgeon is responsible for stopping the surgery when _____.

Decision Making
If the Surgeon believes movement is not allowed or unsafe, then the *Surgeon* may decide not to *move the sawblade.*

Mental Model
PM-1: The Surgeon incorrectly believes he is in a keep-out zone, that the robot is not working, or that he is at the wrong cutting plane.

Interpretation of inputs
The Surgeon may interpret *contact indication or haptic resistance* as a sign that he cannot or should not move here.

Controller States / Modes
If the *Surgeon* is confused by other inputs or expects a different cutting plane, he may hold off on *moving the sawblade*.

Other Inputs
Although *the visualization* is correct, a fouled sensor, a failed reciprocate command, or an unreachable hand-control may cause the *Surgeon* to withhold movement.

Examples from past projects

© Copyright 2026 John Thomas

Breakout 5: Unsafe Feedback

UCA: The Surgeon does not provide move sawblade in-plane while at cutting plane.

Class 1 Scenario:

Unsafe Controller Behavior

- *The Surgeon does not provide move sawblade in-plane while at cutting plane.*
- *The input visualization of cut correctly shows that the sawblade is at the cutting plane.*

Class 2 Scenario:

Unsafe Feedback/Information

- *The input visualization of cut received by the Surgeon does not adequately indicate that sawblade is at cutting plane, and*
- *The sawblade is at the cutting plane is true*

Class 3 Scenario

Class 4 Scenario

Refined Scenarios

Why? Because:

- Visualization of cut is misleading
- Showing the wrong cutting plane
- Showing a sagittal/coronal/axial view instead of a cut-plane view
- Showing the mirror of the cut
- Showing the location of the sawblade incorrectly
- Tracking, calibration, or registration issues
- Tracking is lost
- Showing a portion of bone as cut when it is not
- Showing a portion of bone as not cut when it is
- Etc.

Examples from past projects

© Copyright 2026 John Thomas

Breakout 6: Unsafe Control Execution

UCA: The Surgeon does not provide move sawblade in-plane while at cutting plane.

Class 1 Scenario:

Unsafe Controller Behavior

- *The Surgeon does not provide move sawblade in-plane while at cutting plane*
- *The input visualization of cut correctly shows that the sawblade is at the cutting plane*

Class 2 Scenario:

Unsafe Feedback/Information

- *The input visualization of cut received by the Surgeon does not adequately indicate that sawblade is at cutting plane, and*
- *The sawblade is at the cutting plane is true*

Class 3 Scenario:

Unsafe Control Execution

- *The Surgeon does provide move sawblade in-plane while at cutting plane, and*
- *Move sawblade in-plane is not received by Joint-space Controller while at cutting plane*

Class 4 Scenario

Refined Scenarios

Why? Examples:

- Communication line is cut: severed cable, unseated connector, or dropped link
- Electromagnetic noise overloads the signal: EMI from OR equipment or motors
- Signal wires incorrectly connected: cross-wired, loose, or wrong port
- Noise in the force-torque (FT) sensor masks the command or trips a limit
- Command buffered or delayed, arriving too late to act on
- Etc.

Examples from past projects

© Copyright 2026 John Thomas

Breakout 7: Unsafe Process Behavior

UCA: The Surgeon does not provide move sawblade in-plane while at cutting plane

Class 1 Scenario:

Unsafe Controller Behavior

- *The Surgeon does not provide move sawblade in-plane while at cutting plane*
- *The input visualization of cut correctly shows that the sawblade is at the cutting plane*

Class 2 Scenario:

Unsafe Feedback/Information

- *The input visualization of cut received by the Surgeon does not adequately indicate that sawblade is at cutting plane, and*
- *The sawblade is at the cutting plane is true*

.

Class 3 Scenario:

Unsafe Control Execution

- *The Surgeon does provide move sawblade in-plane while at cutting plane, and*
- *Move sawblade in-plane is not received by Joint-space Controller while at cutting plane*

Class 4 Scenario:

Unsafe Process Behavior

- *Move sawblade in-plane is received by Joint-space Controller when at cutting plane, and*
- *Joint-space Controller does not respond by moving in-plane*

Refined Scenarios

Why?

- Controller requires a valid IK solution, which is unavailable (cannot solve the inverse-kinematics problem in time)
- Controller may run out of usable joint travel, or a joint fails, so the move cannot complete
- Controller responds too late (or not at all) because reaction forces grow too large, or virtual forces are calculated incorrectly, inhibiting motion
- Controller may see that the knee is moving, in which case it will automatically hold position, which results in no in-plane motion
- If an E-stop has been pressed (Controller is in a stopped mode), then all motion commands will be ignored

Examples from past projects

© Copyright 2026 John Thomas

Step 4: Scenarios (Exercise #1)

Unsafe Control Action	Scenario Class 1: Unsafe Controller Behavior	Scenario Class 2: Unsafe Feedback Path	Scenario Class 3: Unsafe Control Path	Scenario Class 4: Unsafe Controlled Process Behavior
The Surgeon does not provide *move sawblade in-plane* while at cutting plane.	The Surgeon does not provide *move sawblade in-plane* while at cutting plane. The input *visualization of cut* correctly shows that the sawblade is at the cutting plane. **Refined Scenarios:** • The Surgeon is confused by the other inputs given • *Contact indication* indicates contact with bone • debris has fouled sensor • *Haptic feedback* provides a large resistance force than expected • Haptic system provides a threshold force to enable movement, but Surgeon thinks he is in a keep out zone or robot is not working • *View of patient* • expecting a different cutting plane • The location of the hand-control for *move sawblade in-plane* is in a place that the Surgeon cannot easily reach or manipulate • The *reciprocate sawblade* command doesn't work, so no reason for Surgeon to move the sawblade in-plane **Solutions / Recommendations / Mitigations** • Visual indicator to display when move in-plane is allowed • Contactless contact sensor • Threshold force not too large • Human-factors study for type and location of handle	The input *visualization of cut* received by the Surgeon does not adequately indicate that sawblade is at cutting plane. The sawblade is at the cutting plane is true. **Refined Scenarios:** The visualization of cut is misleading • Showing the wrong cutting plane • Showing a Sagittal/coronal/axial view instead of a cut plane view • Showing the mirror of the cut • Showing the location of the sawblade incorrectly • issues with tracking, calibration or registration • what happens when tracking is lost? • Showing portion of bone is cut when it is not • Showing portion of bone is not cut when it is **Solutions / Recommendations / Mitigations** • Don't show the visualization of the cut when switching the cutting plane. It's not relevant to what the surgeon is doing, and could cause confusion contributing to this scenario. • Provide a 3D view with sawblade and cutting plane • visual status indicating the current cutting plane • Notify Surgeon when tracking is lost and cutting plane	The Surgeon does provide *move sawblade in-plane* while at cutting plane. *Move sawblade in-plane* is not received by Joint-space Controller while at cutting plane. **Refined Scenarios:** • communication line cut • electromagnetic noise overloads command signal • signal wires are incorrectly connected • noise in FT sensor signal **Solutions / Recommendations / Mitigations** • Watchdog timer goes off if communications is not received within two cycles • System satisfies IEC 60601-1-2 for EMI • Connector type is unique (or uniquely keyed) and tested during manufacturing	*Move sawblade in-plane* is received by Joint-space Controller when at cutting plane. Joint-space Controller does not respond by *move in plane* **Refined Scenarios:** • Knee movng • E-stop pressed by clinical staff • Reaction forces become to great • virtual forces are incorrectly calculated • failure of joint • failure to solve IK problem in time **Solutions / Recommendations / Mitigations** • E-stop easily reachable for clinical staff • Warning if joint movement is less 2 degrees off of expected • Warning if calculation does not converge within 1 second • Warning if knee is moving • Haptic feedback has threshold forces

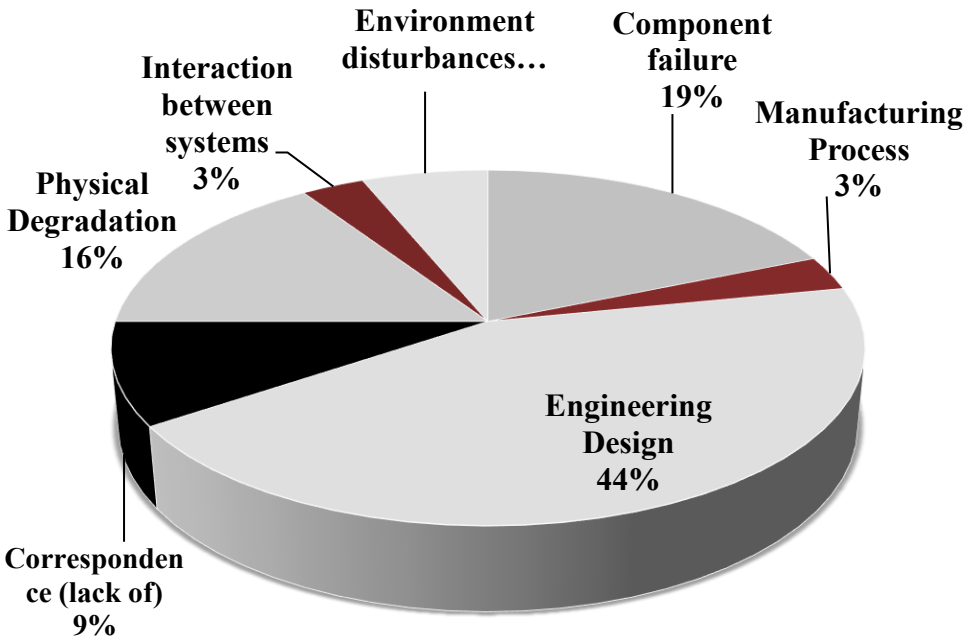
Step 4: Scenarios (Exercise #2)

Unsafe Control Action	Scenario Class 1: Unsafe Controller Behavior	Scenario Class 2: Unsafe Feedback Path	Scenario Class 3: Unsafe Control Path	Scenario Class 4: Unsafe Controlled Process Behavior
The Joint-space Controller provides *move to plane* with parameters that move the robot too quickly for Clinical Staff to avoid.	Joint-space Controller provides *move to plane* with parameters that moves the robot too quickly for Clinical Staff to avoid. Joint-space Controller received *joint velocities* that indicates that move the robot too quickly for Clinical Staff to avoid. **Refined Scenarios:** • The Joint-space Controller does not constrain the absolute robot speed • Joint velocity limits or link length parameters are changed by cyber attack **Solutions / Recommendations / Mitigations:** • sets a maximum arm speed • Joint-space Controller program stored in non-writable disc • CRC check done on Joint-space Controller program during initialization	*Joint positions* received by Joint-space Controller does not adequately indicate that the robot moves too quickly for Clinical Staff to avoid. The robot moves too quickly for Clinical Staff to avoid is true **Refined Scenarios:** • Improper units in settings joint position • Incorrect differentiation from position to speed • Due to manufacturing, maintenance or cyber attack, the zero position recorded in the controller for a joint is off, so thinks it has a short distance to move but actually has a large one . • Sensors mode gets altered (cyber attack or reset accidentally) **Solutions / Recommendations / Mitigations:** • Only use one set of units at control level • Use separate position sensor on joint output (link side) and speed sensor on joint input (motor side). Your motor will be moving much faster, so error in motor side can be reduced by gear ratio • Calibrate arm prior to delivery or after any joint maintenance • Set joint-level speed limits • Calibration data stored in non-	Joint-space Controller does not provide *move to plane* with parameters that move the arm to quickly for Clinical Staff to avoid. Robot arm receives *move to plane* with parameters that move the arm too quickly for Clinical Staff to avoid. **Refined Scenarios:** • Individual joint commands were received out of sequence or delayed • Individual joint commands received by wrong actuators **Solutions / Recommendations / Mitigations:** • Use communication protocol that ends one message to all joints, instead of sequentially • Actuator handshake acknowledges that message received, multiple missed acknowledgements results in safety stop • Clock synchronization between all joints • Unique ID for each actuator	*Move to plane* is not received by robot arm with parameters that move too quickly for Clinical Staff to avoid. Robot arm responds by moving too quickly for Clinical Staff to avoid. **Refined Scenarios:** • Actuator mode altered by automatic reset or cyber attack • Actuator PID gains changed by automatic reset or cyber attack • Knee moves, so associated plane moves creating large distance to move is time **Solutions / Recommendations / Mitigations:** • Calibrate actuator system during manufacturing • Actuator mode and gains cannot be modified in the field • Stop if knee moves more than a prescribed amount

Step 4: Scenarios (Exercise #3)

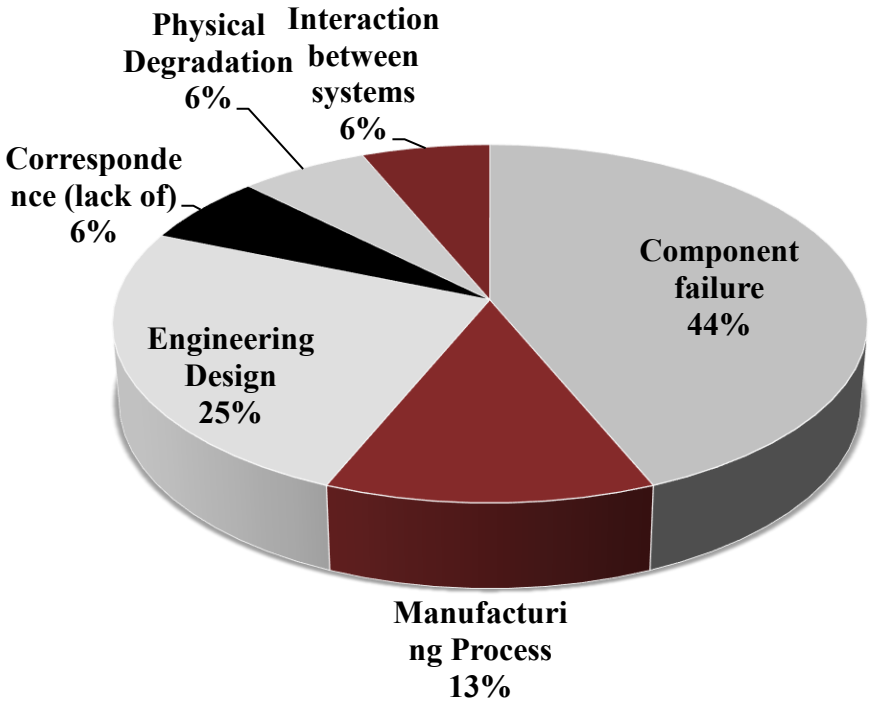
Unsafe Control Action	Scenario Class 1: Unsafe Controller Behavior	Scenario Class 2: Unsafe Feedback Path	Scenario Class 3: Unsafe Control Path	Scenario Class 4: Unsafe Controlled Process Behavior
The Joint-space Controller provides *move to plane* too long, after the Surgeon has stopped authorizing it.	The Joint-space Controller continues providing *move to plane* too long, after the Surgeon has stopped authorizing it. Joint-space Controller received *enable automatic movement* that indicates Surgeon has stopped authorizing it on time **Refined Scenarios:** • Delay in controller processing **Solutions / Recommendations / Mitigations:** • Joint-space controller should be a hard real-time system that guarantees maximum latency • Joint-space controller should have a processor with sufficient power and speed	*Enable automatic movement* received by Joint-space Controller does not indicate Surgeon has stopped authorizing it. (inappropriate duration) Surgeon has stopped authorizing it is true. **Refined Scenarios:** • Signal fails high due to switch failure **Solutions / Recommendations / Mitigations:** • Use OSSD (Output Signal Switching Devices) to detect faults • Highly reliable switch	Joint-space Controller provides *move to plane* with appropriate duration *Move to plane* is received by Robot arm when Surgeon has stopped authorizing it. (inappropriate duration) **Refined Scenarios:** • Joint-space controller updated new joint positions, but communications with arm lost afterwards **Solutions / Recommendations / Mitigations:** • Safety controller to cut arm power if joint positions stop responding	*Move to plane* is received by robot arm with appropriate duration. Robot arm responds by moving to plane too long after Surgeon has stopped authorizing it. (inappropriate duration) **Refined Scenarios:** • Missed command signal to stop • Payload is too heavy, control cannot be guaranteed • power signal loss **Solutions / Recommendations / Mitigations:** • • Use arm with payload mass and center of mass within controllable limits • Automatic brakes when power lost

Types of accident causes found by STPA



STPA causes for UCA1

Types of accident causes found by FMECA



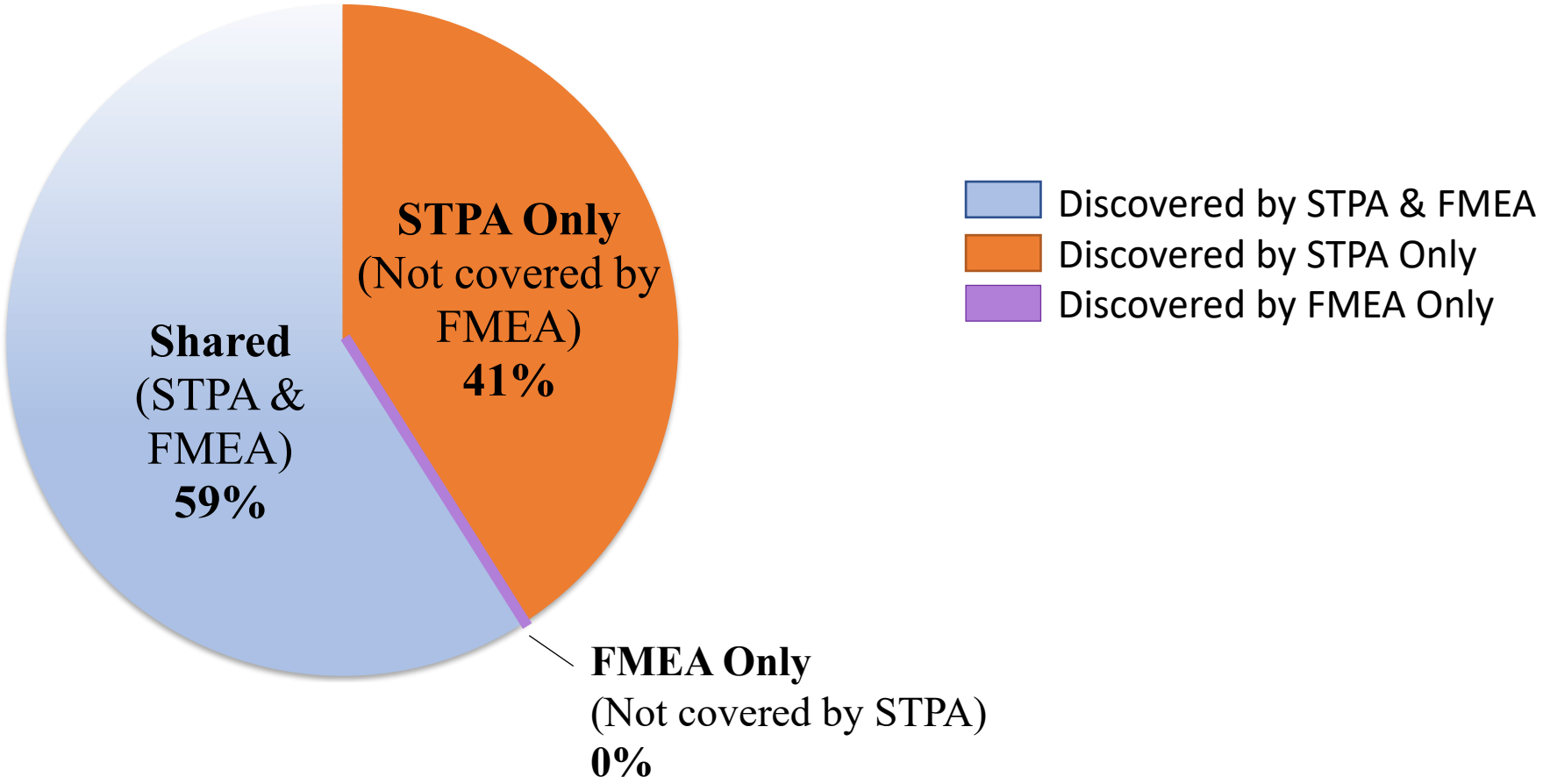
FMECA causes for FM1

Rodrigo Sotomayor, 2015, "Comparing STPA and FMEA on an Automotive Electric Power Steering System"

A comparison of STPA and FMEA

Rodrigo Sotomayor

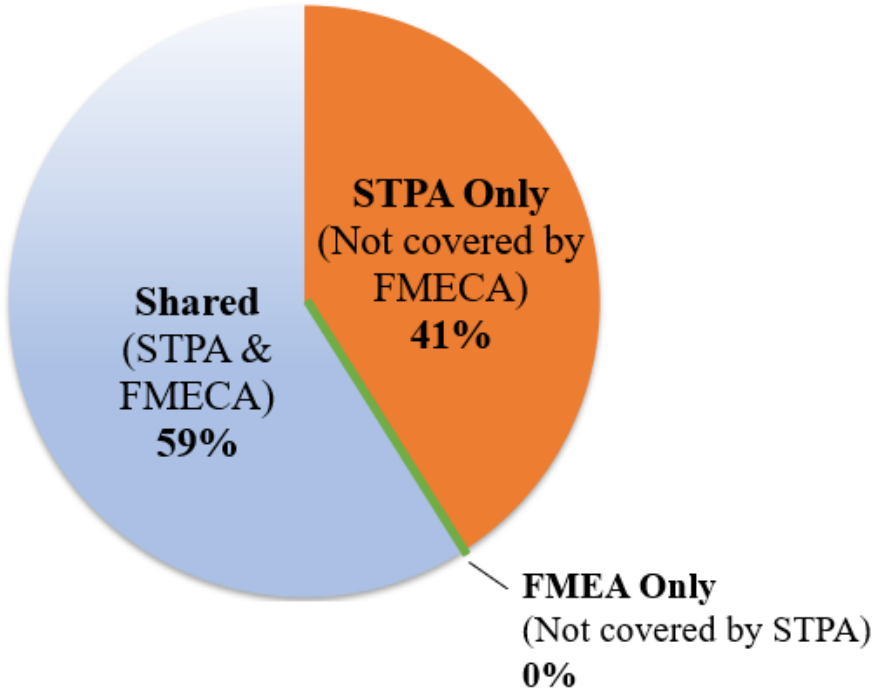
Application: Electric Power Steering System



Rodrigo Sotomayor, 2015, "Comparing STPA and FMEA on an Automotive Electric Power Steering System"

*Using SAE J1739 (DFMEA and PFMEA)

Automotive Electric Power Steering System
STPA causes vs. FMECA causes



137 vs. 95 total causes found
(but there are overlaps)

Examples of causes and new discoveries not identified by FMECA

The method for determining vehicle speed is incorrect; the default control algorithm behavior when it detects a sensor fault will lead to a hazard if it occurs during highway driving.

The control algorithm has no prioritization for critical operation components if there is low voltage available.

The design of another controller will inadvertently limit speed if auxiliary assistance is provided (Cruise control).

High friction event is detected at low speed.

Assistance would not be provided because there is a conflict between steering angle and speed signals.

Normal sensor operation could result in delayed signal information over BUS, leading to hazard

Chime is not loud enough or displayed in a way it is easily noticeable by the driver.

Focus: What methods can consistently <u>discover</u> these overlooked flaws?
NOT: Can flaws, once they are known, be back-fit into a method or spreadsheet in hindsight?

Rodrigo Sotomayor, 2015, "Comparing STPA and FMEA on an Automotive Electric Power Steering System"

© Copyright 2025 John Thomas