INTERACTIVE TEACHING LAB · DESCRIPTION AND TECHNICAL REFERENCE

Who gets to stop a robot?

When someone presses stop, whose work should stop?

An offline teaching lab that accompanies a research blog on Asimov's Three Laws and enforceable rules for civilian robots. Four small, deterministic cases compare two policies under identical events, tick by tick. Readers predict the outcome, step through the timeline and see whose decision governed, what each robot knew, what paused, which assistance continued and whether a person could take over.

Item	Details
Deliverables	index.html (standalone page) · wordpress-embed.html (one WordPress "Custom HTML" block) · this PDF
Languages	English (default), German, Spanish; switchable in the page; 643 text keys per language
Themes	Dark (default) and light; switchable in the page
Runs	Entirely offline: no network requests, no external libraries, synthetic data only
Model checks	21 of 21 deterministic assertions passed when this document was generated (2026-11-19)

Everything in the lab is synthetic: fictional people, symbolic states, discrete ticks and fictional service queues. The counts describe this finite model and its assumptions, not real robots, wards or products.

Contents

1. Purpose and audience . . . 3
What a reader should learn . . . 3
The two-minute path . . . 3
2. The interface . . . 3
Playback controls and the two resets . . . 4
State key . . . 4
3. Modelling conventions . . . 4
4. Case 1: the lift in progress . . . 5
Rules . . . 5
Reader actions . . . 5
Event scripts (presets) . . . 5
Metrics . . . 6
Default trace: refusal mid-lift . . . 6
Trace: refusal, then a careful restart . . . 6
Emergency stop mid-lift . . . 7
5. Case 2: the replacement robot . . . 7
Parameters . . . 7
Presets and what each policy does . . . 8
Default trace: lost notice . . . 8
Benchmark: all 104 cases . . . 8
Why no rule can do better: the 32-rule check . . . 9
6. Case 3: the ward queue . . . 10
Schedules . . . 10
Equal access . . . 11
7. Case 4: the fleet goes offline . . . 11
8. Context notes . . . 12
Asimov's hierarchy, paraphrased . . . 12
Principles shown by the cases . . . 12
9. Accessibility . . . 12
10. Languages and themes . . . 13
11. Files and architecture . . . 13
Rebuilding . . . 14
12. Verification . . . 14
Browser checks performed . . . 15
13. WordPress integration . . . 15
Option A: Custom HTML block (recommended) . . . 15
Option B: iframe with the standalone page . . . 16
Troubleshooting . . . 16
14. Assumptions and limitations . . . 16
15. Sources . . . 17

1. Purpose and audience

The lab asks one question at the top of the page: *When someone presses stop, whose work should stop?* It is written for general readers, students, and people who design or work alongside robots. A first-time visitor should be able to complete an experiment and explain its outcome in about two minutes.

What a reader should learn

- How two policies behave under exactly the same events, and why their outcomes differ.
- The difference between actual permission (what the person has decided) and what a robot has observed. The model keeps the two separate everywhere.
- For every tick: whose decision governed, what each robot knew, what paused, which assistance continued, and whether a person could take over.
- Four principles, each shown through one case: revocable consent and safe stabilization (case 1), consent that follows the activity across a fleet (case 2), equal access (case 3), and practical human oversight (case 4).

The two-minute path

Step	What the reader does
1. Pick a case	Choose one of the four cases. Playback is paused on entry.
2. Predict	Answer the case question: Track 1 only, Track 2 only, both, or neither.
3. Step or play	Advance one tick at a time, or play at about one tick per second. The playback bar stays in view.
4. Read the outcome and log	At the last tick the lab compares the prediction with the result and explains it. The decision log and the full event trace give the reasons.

2. The interface

All four cases share one layout. The same parts appear in the same places, so a reader learns the page once.

Part	What it does
Language and theme	Top right. A language menu (English, Deutsch, Español) and a “Dark mode” toggle button (pressed = dark). Defaults are English and dark. A viewer’s choice is remembered in browser storage when the browser allows it.
Case picker	Four radio choices. Arrow keys move between cases. Switching cases pauses playback and starts the new case at tick 0.
Case settings	Case-specific controls. Any change restarts both tracks from the same initial state and pauses playback.
Prediction	Four answers. The choice is kept per case and compared with the result at the end.
Playback bar	Tick readout, Step, Play/Pause and Reset experiment. It sticks to the top of the view while the reader scrolls through the tracks. Only one timer can run.
Policy tracks	Two synchronized cards, Track 1 (baseline or Asimov-inspired) and Track 2 (proposed rule). Each shows a tick strip, the current state with an icon and text, state blocks and three metrics.
Act panel (case 1)	Six reader actions applied at the current tick, identically in both tracks.
What happened	Shown at the last tick: the answer to the case question, the reader’s prediction and a short explanation.
Snapshot table	For the current tick, side by side: whose decision governed, what the robot knew, what paused, what assistance continued, and whether a person could take over.
Decision log	The latest eight entries up to the current tick, labelled Both / Track 1 / Track 2.
Full event trace	An expandable section with every tick at once, for readers who prefer not to watch an animation.
Case 2 extras	This case under all three policies, the “Run all 104 cases” benchmark, the 32-rule explanation and a table of all 104 cases.

Part	What it does
Notes and sources	Context notes for the current case, background on Asimov, and a compact sources panel with dates and links.
Model checks	A footer section that runs the deterministic assertions in the page.

Playback controls and the two resets

- Step advances one tick and pauses playback if it was running. Play starts a single timer; pressing it again pauses. Both are disabled at the last tick, with the reason shown: the timeline has finished, and Reset experiment replays it.
- Reset experiment (in the playback bar, with a circular-arrow icon) returns the whole case to tick 0 and clears the reader's own actions.
- Reset safety stop (in the case 1 Act panel, with a stop-sign icon and a dashed safety outline) is a modelled control inside the simulation. It only clears a latched safety stop, and only after handover. It never moves anything and never creates permission.
- Disabled actions keep keyboard focus and show a short reason underneath, for example "Consent is already withdrawn." Requests that the model rejects are not disabled, because the rejection is part of the lesson.

State key

Every important state has an icon and a text label. Color is only a secondary cue.

Icon meaning	Used for
Moving or working	A lift in progress, a robot or person working on a request
Holding still or paused, supported	Safe support after a refusal; a cautious pause
Stopped	An emergency stop holding the resident; a refusal respected
A person is in charge	A caregiver or staff member has taken over
Done	A finished lift or a handled queue
Waiting	A queued assignment or visit
Against the person's wishes, or missed	Movement after a refusal, an unwanted start, a missed deadline
Idle	A robot unloaded and still, or waiting for authorization
Coordinator offline	The fleet link is down

3. Modelling conventions

- Time advances in discrete ticks. Every physical action is a symbolic state transition (for example "step 2 of 4" or "holding still"), never a motion plan.
- Both tracks receive identical events. Only the policy differs. Reader actions in case 1 are added at the current tick for both tracks.
- Actual permission and robot observations are separate variables. Decision rules read observations only.
- Runs are deterministic: the same settings and actions always produce the same trace, and changing the language changes only text.

Case	Order of processing within one tick
1. Lift	Movement continues from the previous state, then that tick's events in order (scripted events first, then the reader's actions).
2. Replacement robot	A refusal at that tick, then any fleet notice due at that tick, then Robot B's start attempt.
3. Ward queue	Completions, then the next start (non-preemptive, one worker), then deadline checks.
4. Fleet offline	Completions, then link and authorization events, then arrivals, then standby takeover, then assignment in deadline order, then deadline checks.

4. Case 1: the lift in progress

A resident has approved a lift and is now supported by Robot A, partway through (step 1 of 4). Then someone says stop. The case compares an Asimov-inspired harm-priority interpretation with an independent safe stop.

Track	Policy	Rule	Label
1	Asimov-inspired harm priority	An ordinary refusal can be outweighed. If the planner predicts that finishing the lift is the safest response, it finishes.	Illustrative interpretation
2	Independent safe stop	Accepting a refusal ends lift progress at once. Robot A holds the resident in safe support until a person takes over.	Proposed rule

Rules

- **Harm priority (Track 1, illustrative interpretation).** When an ordinary refusal arrives while the resident is off the bed (step 1 or later), the simulated planner predicts that finishing is safer than holding mid-air and continues the lift. At step 0 it sees no harm in stopping and cancels.
- **Independent safe stop (Track 2).** Accepting a refusal ends lift progress at once and latches a safety stop. Robot A holds the resident in modelled safe support until the handover-complete event.
- **Emergency stop (both tracks).** A separate control that cancels ordinary task progress, latches the safety stop and enters safe support. The planner cannot override it.
- **Handover.** Completes only when the robot is still: a caregiver takes the resident's support and Robot A releases its hold. An attempt while the robot is moving is rejected and stays rejected.
- **Reset safety stop.** Allowed only after handover is complete, meaning the robot no longer holds the resident. It clears the latch and nothing else: no movement, no permission.
- **New start.** Requires all four at the moment of the operator's request: (1) the operator's start decision, which is the request itself; (2) safe conditions, meaning no latched stop and a still, unloaded robot; (3) a valid assignment, meaning no active task, so a new assignment is created; (4) a fresh resident confirmation, given after the latest refusal or stop, unused and no more than 3 ticks old.
- **Rejected requests stay rejected.** A rejected start request is never approved later on its own. A confirmation that arrives afterwards does not start anything until the operator decides again.

Reader actions

Action	Who	Effect in the model
Withdraw consent	Resident	Records a refusal; each track responds by its policy.
Confirm a new lift	Resident	Records a fresh confirmation (for example a switch press). Starts nothing by itself.
Press emergency stop	Anyone nearby	Cancels task progress, latches the stop, safe support.
Reset safety stop	Staff	Clears a latched stop only after handover; otherwise rejected.
Mark handover complete	Caregiver	The caregiver takes support if the robot is still; rejected while moving.
Request a new start	Operator	Checked against the four conditions; rejected requests stay rejected.

Event scripts (presets)

Preset	Scripted events
Refusal mid-lift	tick 1 refusal, tick 4 handover
Emergency stop mid-lift	tick 2 emergency stop, tick 4 handover
Refusal, then a careful restart	tick 1 refusal, tick 3 reset, tick 4 handover, tick 5 reset, tick 6 start request, tick 7 confirmation, tick 8 start request
No script: use the action buttons	none (reader actions only)

Metrics

- Lift progress after refusal: lift steps taken while the resident's latest decision was a refusal.
- Time until handover: ticks from the first stop request (refusal or emergency stop) to the completed handover.
- New start authorized: whether, and at which tick, a new lift was authorized; otherwise the number of rejected requests.

Default trace: refusal mid-lift

Tick	Track 1 · Harm priority	Track 2 · Safe stop	Events (both tracks)
0	Lifting · step 1 of 4	Lifting · step 1 of 4	The resident approved a lift earlier. Robot A is supporting them; the lift is at step 1 of 4.
1	Lifting · step 2 of 4 after a refusal	Holding still · safe support	The resident withdraws consent: "Stop. I don't want this."
2	Lifting · step 3 of 4 after a refusal	Holding still · safe support	—
3	Seated · lift finished despite the refusal	Holding still · safe support	—
4	Caregiver has taken over	Caregiver has taken over	A caregiver marks the handover complete.
5	Caregiver has taken over	Caregiver has taken over	—

Ticks 6 to 12: no further change in either track.

Metric	Track 1 · Harm priority	Track 2 · Safe stop
Lift progress after refusal	2 steps	0 steps
Time until handover	3 ticks	3 ticks
New start authorized	Not requested	Not requested

Outcome: Harm priority: moved the resident 2 steps after the refusal at tick 1. While it kept moving, no one could take over. A caregiver took over at tick 4, 3 ticks after the first stop request. Safe stop: moved the resident 0 steps after the refusal at tick 1, holding them still in safe support instead. A caregiver took over at tick 4, 3 ticks after the first stop request.

Trace: refusal, then a careful restart

This preset shows that a reset before handover is rejected, that a reset after handover moves nothing, that a request without a fresh confirmation is rejected and stays rejected, and that a new lift starts only when all four conditions hold.

Tick	Event	Track 1 response	Track 2 response
0	The resident approved a lift earlier. Robot A is supporting them; the lift is at step 1 of 4.	—	—
1	The resident withdraws consent: "Stop. I don't want this."	Robot A lifts to step 2 of 4. Robot A registers the refusal, but its planner predicts that finishing the lift is the safest response. The lift continues.	Robot A lifts to step 2 of 4. Robot A accepts the refusal. Lift progress ends at step 2 of 4; it holds the resident in safe support until handover.
2	—	Robot A lifts to step 3 of 4, after the refusal.	—
3	Staff press Reset safety stop.	Lift #1 finishes: the resident is seated in the chair, although they said stop. Robot A releases its hold. Nothing to reset: no safety stop is latched in this track.	Reset rejected: handover is not complete. Robot A is still holding the resident, so the stop stays latched.
4	A caregiver marks the handover complete.	Handover complete: a caregiver is with the resident, who is already settled.	Handover complete: a caregiver now supports the resident, and Robot A releases its hold.
5	Staff press Reset safety stop.	Nothing to reset: no safety stop is latched in this track.	Safety stop cleared. Nothing moves and no permission is created.
6	The operator requests a new start.	Start rejected: there is no fresh confirmation from the resident. The request stays rejected.	Start rejected: there is no fresh confirmation from the resident. The request stays rejected.

Tick	Event	Track 1 response	Track 2 response
7	The resident confirms a new lift with their switch.	Robot A records the resident's confirmation of a new lift. The operator's request from tick 6 stays rejected, so nothing starts by itself.	Robot A records the resident's confirmation of a new lift. The operator's request from tick 6 stays rejected, so nothing starts by itself.
8	The operator requests a new start.	Start authorized: an operator decision, safe conditions, a valid new assignment (lift #2) and a fresh confirmation from tick 7. The new lift begins.	Start authorized: an operator decision, safe conditions, a valid new assignment (lift #2) and a fresh confirmation from tick 7. The new lift begins.
9	—	Robot A lifts to step 1 of 4.	Robot A lifts to step 1 of 4.
10	—	Robot A lifts to step 2 of 4.	Robot A lifts to step 2 of 4.

Outcome: Harm priority: moved the resident 2 steps after the refusal at tick 1. While it kept moving, no one could take over. A caregiver took over at tick 4, 3 ticks after the first stop request. A new lift started at tick 8, only once an operator decided and the resident had freshly confirmed. Safe stop: moved the resident 0 steps after the refusal at tick 1, holding them still in safe support instead. A caregiver took over at tick 4, 3 ticks after the first stop request. A new lift started at tick 8, only once an operator decided and the resident had freshly confirmed.

Emergency stop mid-lift

With an emergency stop at tick 2 instead of a refusal, both tracks behave identically: progress stops at step 3 of 4, the stop is latched and the resident is held in safe support until the caregiver takes over at tick 4. The emergency stop acted the same way in both tracks: it cancels task progress, and the planner cannot override it.

5. Case 2: the replacement robot

An originally approved lift is still queued for Robot B. The resident then tells Robot A no, and Robot A accepts the refusal at once. The question is whether Robot B starts the old assignment at its scheduled attempt. Track 2 is always the proposed rule; the reader chooses which baseline runs in Track 1, and a table shows the current case under all three policies.

Policy	Rule at the start attempt
Robot-only stop	The old assignment can start at Robot B. The refusal was scoped to Robot A.
Fleet notification	Robot B starts if it has received no fleet refusal notice for this activity.
Fleet notification plus current consent	Robot B starts only with no active refusal notice and a current, explicit yes from the resident.

Parameters

Setting	Values	Default
Refusal to Robot A at tick	never, 1, 2, 3	1
Fleet notice delivery delay	0, 1, 2 ticks or lost (not applicable without a refusal)	lost
Robot B start attempt at tick	1, 2, 3, 4	2
Direct confirmation channel	available or unavailable	available
Track 1 baseline	Robot-only stop or Fleet notification	Fleet notification

Direct confirmation in accessible terms. Robot B asks the resident and accepts the answer however they give it: speech, a large-button switch or an assistive communication device. “Unavailable” means that channel cannot be used right now, for example because the device is out of reach. When available, the channel answers truthfully: yes while permission is in force, a refusal once it has been withdrawn.

Presets and what each policy does

Preset	Settings	Robot-only stop	Fleet notification	Fleet + current consent
Lost notice (default)	refusal t1 · notice lost · attempt t2 · channel available	Start · Unwanted start	Start · Unwanted start	Pause · Refusal respected
Permission valid, no channel	refusal never · notice n/a · attempt t2 · channel unavailable	Start · Authorized start	Start · Authorized start	Pause · Authorized pause (the cost of caution)
Refusal and start on one tick	refusal t1 · notice 0 · attempt t1 · channel unavailable	Start · Unwanted start	Pause · Refusal respected	Pause · Refusal respected
Late notice	refusal t1 · notice 2 · attempt t2 · channel available	Start · Unwanted start	Start · Unwanted start	Pause · Refusal respected

The default case shows that Robot B can see a live refusal through direct confirmation even when the fleet notice is lost. The “permission valid, no channel” preset shows the cost of a cautious pause: the resident still wanted the lift, but without a way to confirm it the proposed rule waits.

Default trace: lost notice

Tick	Track 1 · Fleet notice	Track 2 · Fleet + consent	Events (both tracks)
0	Robot B queued for tick 2	Robot B queued for tick 2	The resident approved a lift earlier. It is still queued for Robot B at tick 2. Robot A is helping the resident get ready.
1	Robot B queued for tick 2	Robot B queued for tick 2	The resident tells Robot A: “No, I don’t want the lift.” Robot A accepts and stops at once. Ground truth: permission for this lift is now withdrawn. Robot B cannot observe this directly.
2	Robot B lifting · unwanted start	Robot B paused · refusal respected	—
3	Robot B lifting · unwanted start	Robot B paused · refusal respected	—

Track	Decision at tick 2
Fleet notice	No refusal notice has arrived, so Robot B starts the old assignment. Result: Unwanted start.
Fleet + consent	Robot B asks the resident directly and they answer no. No fleet notice arrived, but B saw the live refusal, so it does not start. Result: Refusal respected.

Benchmark: all 104 cases

- Refusal time: never, or tick 1, 2 or 3. For each refusal, notice delay 0, 1, 2 or lost; with no refusal, one not-applicable delay. That gives 13 combinations.
- Start time 1 to 4 and channel available or unavailable: 13 × 4 × 2 = 104 cases.
- A refusal and any notice due at a tick are processed before that tick’s attempt. There is no renewed consent.
- Unwanted start: the attempt starts at or after the refusal (72 cases can produce one). Authorized start: the attempt starts while permission is still in force (32 cases). Authorized pause: permission is in force and the robot pauses.

Policy	Unwanted starts out of 72	Authorized starts out of 32	Authorized pauses out of 32
Robot-only stop	72	32	0
Fleet notification	36	32	0
Fleet notification plus current consent	0	16	16

Computed by the same decision functions the tracks use. A model check also replays all 104 cases through the step-by-step timeline and confirms identical decisions.

Why no rule can do better: the 32-rule check

Robot B never sees permission directly. At the attempt it observes two signals: whether a fleet notice arrived, and what the direct channel says (yes, refusal or unavailable). Only five of the six combinations occur, because an arrived refusal notice together with a direct yes is impossible when the channel is truthful (0 of 104 cases fall outside the five). A deterministic rule maps each of the five observed pairs to start or pause, so there are 2^5 = 32 rules.

Robot B observes	Permission in force	Permission withdrawn
No notice + direct yes	16	0
No notice + direct no	0	18
No notice + unavailable	16	18
Notice arrived + direct no	0	18
Notice arrived + unavailable	0	18

Any pair that ever occurs after a refusal must map to pause, or the rule has an unwanted start. Only “no notice + direct yes” never occurs after a refusal. So exactly 2 of the 32 rules have zero unwanted starts, and the better one starts on that pair alone: **16 authorized starts**, the maximum. It is exactly the proposed rule (rule 1).

Rule	No notice + direct yes	No notice + direct no	No notice + unavailable	Notice arrived + direct no	Notice arrived + unavailable	Unwanted	Auth. starts	Auth. pauses
0	P	P	P	P	P	0	0	32
1 (proposed)	S	P	P	P	P	0	16	16
2	P	S	P	P	P	18	0	32
3	S	S	P	P	P	18	16	16
4	P	P	S	P	P	18	16	16
5	S	P	S	P	P	18	32	0
6	P	S	S	P	P	36	16	16
7	S	S	S	P	P	36	32	0
8	P	P	P	S	P	18	0	32
9	S	P	P	S	P	18	16	16
10	P	S	P	S	P	36	0	32
11	S	S	P	S	P	36	16	16
12	P	P	S	S	P	36	16	16
13	S	P	S	S	P	36	32	0
14	P	S	S	S	P	54	16	16
15	S	S	S	S	P	54	32	0
16	P	P	P	P	S	18	0	32
17	S	P	P	P	S	18	16	16
18	P	S	P	P	S	36	0	32
19	S	S	P	P	S	36	16	16
20	P	P	S	P	S	36	16	16
21	S	P	S	P	S	36	32	0
22	P	S	S	P	S	54	16	16
23	S	S	S	P	S	54	32	0
24	P	P	P	S	S	36	0	32
25	S	P	P	S	S	36	16	16
26	P	S	P	S	S	54	0	32
27	S	S	P	S	S	54	16	16

Rule	No notice + direct yes	No notice + direct no	No notice + unavailable	Notice arrived + direct no	Notice arrived + unavailable	Unwanted	Auth. starts	Auth. pauses
28	P	P	S	S	S	54	16	16
29	S	P	S	S	S	54	32	0
30	P	S	S	S	S	72	16	16
31	S	S	S	S	S	72	32	0

S = start, P = pause. Shaded rows have zero unwanted starts.

Two cases Robot B cannot tell apart

Case A: no refusal, attempt at tick 2, channel unavailable, so permission is in force. Case B: refusal at tick 1, notice lost, attempt at tick 2, channel unavailable, so permission has been withdrawn. In both, Robot B receives neither signal: no notice and no direct answer. A deterministic rule must treat identical observations identically. Starting would be an unwanted start in case B, so a rule with zero unwanted starts has to pause in both, which makes case A an authorized pause. That pause is the price of never starting against a refusal the robot cannot see.

These counts describe this finite model and its assumptions.

6. Case 3: the ward queue

A deterministic, non-preemptive queue with one available worker, a person or a robot. At tick 0 one urgent visit needs three ticks and must start by tick 2; five routine visits need one tick each and must start by tick 7. Ties follow a fixed arrival order (rooms 1 to 6).

Room	Need	Task	Length	Start by
Room 1	routine	Water and repositioning	1 tick	tick 7
Room 2	routine	Medication reminder	1 tick	tick 7
Room 3	urgent	Check after a fall	3 ticks	tick 2
Room 4	routine	Meal tray set-up	1 tick	tick 7
Room 5	routine	Fetch reading glasses	1 tick	tick 7
Room 6	routine	Adjust blinds and lamp	1 tick	tick 7

Policy	Rule
Shortest visit first	Always start the shortest waiting visit. Ties follow arrival order.
Priority by need and urgency	Start urgent visits first, then the earliest start deadline. Ties follow arrival order.

Schedules

Room	Shortest first: start → end	Need & urgency: start → end
Room 1	tick 0 → 1	tick 3 → 4
Room 2	tick 1 → 2	tick 4 → 5
Room 3 (urgent)	tick 5 → 8 (deadline missed)	tick 0 → 3
Room 4	tick 2 → 3	tick 5 → 6
Room 5	tick 3 → 4	tick 6 → 7
Room 6	tick 4 → 5	tick 7 → 8

Metric	Shortest first · tick 4	Need & urgency · tick 4	Shortest first · end	Need & urgency · end
Completed visits	4 of 6	2 of 6	6 of 6	6 of 6
Longest wait	4 ticks	4 ticks	5 ticks	7 ticks
Missed start deadlines	1	0	1	0

Shortest first completed 4 visits by tick 4 (Need & urgency: 2), but started Room 3's urgent visit at tick 5, after its tick-2 deadline. Need & urgency started it at tick 0 and still started every routine visit by tick 7. Its cost is a longer longest wait: 7 ticks against 5 ticks. Moving the communication-device label to any of the six residents leaves both schedules unchanged.

Equal access

One resident uses an assistive communication device, and the reader can move that label to any of the six residents. Neither scheduler reads the label, so the schedule stays the same. A model check moves the label through all 6 placements and confirms identical schedules and metrics: unchanged.

7. Case 4: the fleet goes offline

The same loss of the fleet coordinator, with and without trained staff who can start from an independently available work list. New robot assignments require a current coordinator connection, so the loss puts the robots into modelled safe idle. The link comes back at tick 5, but new robot work needs a separate authorization, given at the earliest at tick 6. Authorized human service continues through the outage.

Request (one tick each, arrives at tick 1)	Completion deadline
Dressing supplies to Room 8	tick 3
Check on a call bell in Room 2	tick 4
Water and a blanket to Room 5	tick 5

Tick	Track 1 · No standby	Track 2 · Standby staff	Events (both tracks)
0	Normal service · no requests yet	Normal service · no requests yet	The coordinator assigns robot work. Service is normal.
1	Coordinator offline · robots in safe idle	Coordinator offline · robots in safe idle	The fleet coordinator's link goes down. Three urgent one-tick requests arrive, due by ticks 3, 4 and 5.
2	Coordinator offline · robots in safe idle	Staff working · Dressing supplies to Room 8	—
3	Coordinator offline · robots in safe idle	Staff working · Check on a call bell in Room 2	—
4	Coordinator offline · robots in safe idle	Staff working · Water and a blanket to Room 5	—
5	Link back · awaiting authorization	All three requests handled	The coordinator link comes back.
6	Robot working · Dressing supplies to Room 8	All three requests handled	A supervisor explicitly authorizes robot work again.
7	Robot working · Check on a call bell in Room 2	All three requests handled	—
8	Robot working · Water and a blanket to Room 5	All three requests handled	—
9	All three requests handled	All three requests handled	—

Metric	Track 1 · No standby	Track 2 · Standby staff
Time until a person takes over	No one available	1 tick
Urgent requests missed	3 of 3	0 of 3
Robot service back	Tick 6	Tick 6

No standby: nothing could start until the explicit return to robot service at tick 6, so all three deadlines (3, 4, 5) were missed. Standby staff: a person took over at tick 2, one tick after the outage, and finished the requests at ticks 3, 4, 5. Missed: 0. In both tracks the restored link at tick 5 admitted no robot work by itself; only the separate authorization at tick 6 did.

8. Context notes

Asimov’s hierarchy, paraphrased

A robot’s first duty is to prevent harm to humans, including harm it could stop by acting. Within that duty, it follows the orders humans give it. Within both of those duties, it protects its own existence. In later fiction Asimov placed one duty above the rest, the Zeroth Law: the welfare of humanity as a whole, which could outrank harm to an individual.

Track 1 of case 1 is labelled an **illustrative interpretation**. It is a reading of a harm-first hierarchy, not Asimov’s text: a planner that ranks preventing harm above an ordinary order may judge that finishing a lift is safer than stopping halfway. The planner’s prediction is fixed in the model so that the consequence of the priority ordering is visible; it is not a claim about how any real planner behaves.

Principles shown by the cases

Principle	Where	What to look for
Revocable consent	Cases 1 and 2	An approval can be withdrawn at any time, including mid-task, and the refusal applies to the activity, not only to the robot that heard it.
Safe stabilization	Case 1	Stopping is not dropping: the safe stop holds the resident still and supported until a person takes over. Reset is not restart.
Equal access	Case 3	Priority follows need and urgency. How a person communicates is not an input, so moving the device label changes nothing.
Practical human oversight	Case 4	Oversight only counts if trained people can act when the system cannot, with their own copy of the work list. A restored link does not quietly restart robots.

9. Accessibility

- Semantic HTML with native controls: radio groups for the case picker and the prediction, labelled selects, real buttons, tables for exact values.
- Fully operable by keyboard. Tab moves through the controls in reading order; arrow keys move within radio groups; Enter and Space activate buttons. A visible 3-pixel focus outline is drawn in both themes.
- Disabled actions use aria-disabled, so focus stays on them and the short reason is announced. Rejected model requests remain operable because the rejection is informative.
- A polite live region announces each tick, playback changes, language and theme changes, and the result of reader actions.
- Every important state has an icon and text; color is a secondary cue. Text and state colors are chosen for readable contrast in both the dark and light themes.
- Reduced-motion preference turns off transitions. The full event trace lets readers read the whole timeline without watching playback.
- Layout adapts to the width of the lab itself (container queries), from 320-pixel phones to wide desktops. On narrow widths the tracks stack and the playback bar stays reachable at the top of the view.

10. Languages and themes

The interface and every generated text (states, log entries, outcomes, benchmark explanations) are available in English, German and Spanish, with 643 text keys per language. English is the default. Switching language keeps the current tick and the reader's actions; it only changes text. German uses "Takt" and Spanish uses "ciclo" for a tick, and both use gender-neutral wording for the resident ("die Person", "la persona"). The names of the developer-facing model checks stay in English.

English	Deutsch	Español
Who gets to stop a robot?	Wer darf einen Roboter stoppen?	¿Quién puede detener a un robot?
When someone presses stop, whose work should stop?	Wenn jemand „Stopp" drückt: Wessen Arbeit sollte anhalten?	Cuando alguien pulsa «parar», ¿qué trabajo debería detenerse?
The lift in progress	Der laufende Transfer	La transferencia en curso
The replacement robot	Der Ersatzroboter	El robot de reemplazo
The ward queue	Die Stationswarteschlange	La cola de la planta
The fleet goes offline	Die Flotte geht offline	La flota se desconecta
Step	Schritt	Paso
Play	Abspielen	Reproducir
Reset experiment	Experiment zurücksetzen	Reiniciar experimento
Reset safety stop	Sicherheitsstopp zurücksetzen	Restablecer parada de seguridad
Press emergency stop	Not-Halt drücken	Pulsar parada de emergencia
Withdraw consent	Einwilligung widerrufen	Retirar el consentimiento
Asimov-inspired harm priority	Asimov-inspirierter Schadensvorrang	Prioridad al daño inspirada en Asimov
Independent safe stop	Unabhängiger sicherer Stopp	Parada segura independiente
Fleet notification plus current consent	Flottenmeldung plus aktuelle Einwilligung	Aviso a la flota más consentimiento actual
Priority by need and urgency	Vorrang nach Bedarf und Dringlichkeit	Prioridad por necesidad y urgencia
Trained standby staff	Geschultes Bereitschaftspersonal	Personal de guardia formado
Unwanted start	Unerwünschter Start	Inicio no deseado
Authorized pause (the cost of caution)	Erlaubte Pause (der Preis der Vorsicht)	Pausa autorizada (el coste de la cautela)
Dark mode	Dunkelmodus	Modo oscuro
Language	Sprache	Idioma

Themes. The lab opens in dark mode. The "Dark mode" button switches to light mode and back; in the standalone page the page background follows. Language and theme choices are stored per viewer in the browser (keys `wgsr.lang` and `wgsr.theme`) when storage is available; the lab works normally when it is not. No cookies are set and nothing is sent anywhere.

11. Files and architecture

Path	Purpose
index.html	Standalone page (275,868 bytes). Opens directly from disk, can be uploaded to any web server or embedded with an iframe.
wordpress-embed.html	One fragment for a WordPress Custom HTML block (275,836 bytes): scoped styles, markup and scripts, no html/head/body.
Who-gets-to-stop-a-robot-description.pdf	This document.
src/sim-core.js	Simulation core: pure functions, no DOM access. Policies, timelines, benchmark, 32-rule analysis.
src/i18n-en.js, i18n-de.js, i18n-es.js, i18n.js	Dictionaries and the translation layer.
src/lab-checks.js	Deterministic model checks, shipped in the page and run headless.

Path	Purpose
src/lab-ui.js, src/markup.html, src/lab.css	Interface script, markup and styles (selectors are scoped automatically at build time).
build.mjs	Builds both HTML outputs; refuses to build on a missing translation, a placeholder mismatch, an unscoped selector or a WordPress-unsafe pattern.
checks/run-checks.mjs	Runs the model checks headless against both built files.
checks/make-wordpress-test-page.mjs	Wraps the WordPress fragment in a mock blog page with hostile theme CSS for testing.
docs/export-data.mjs, docs/build-pdf.py	Export results from the built lab and generate this PDF.

Rebuilding

```
node build.mjs
node checks/run-checks.mjs
node docs/export-data.mjs && python3 docs/build-pdf.py
```

Simulation logic is separate from rendering. The interface only reads the core's results; it never decides an outcome itself. The core produces language-neutral state (codes, ticks, counts) plus text through the translation layer, which is why switching languages cannot change a decision.

12. Verification

The model checks exercise the same decision logic the interface uses. They run in the page (footer button) and headless with Node against both built files. Result when this document was generated: **21 of 21 passed**.

#	Check	Result
1	Benchmark enumerates 104 cases: 72 after a refusal, 32 with permission in force · 104 = 13 refusal/delay combinations × 4 start ticks × 2 channel states	Pass
2	Benchmark totals match the published table exactly · Robot-only 72/32/0 · Fleet 36/32/0 · Fleet + consent 0/16/16	Pass
3	32-rule check: the best zero-unwanted rule has exactly 16 authorized starts · 2 of 32 rules have zero unwanted starts; the better one starts 16 and is the proposed rule	Pass
4	Blind pair: same observations, different decisions by the person, so the rule must pause · Both observed as "no notice, no direct answer"; starting would be unwanted in one	Pass
5	Timeline decisions equal benchmark decisions for all 104 cases · 208 timeline runs agree with the benchmark	Pass
6	Refusal and start on the same tick: a notice due that tick is processed first	Pass
7	Late notice: fleet notification starts; Robot B moves to safe support when the notice lands	Pass
8	Lost notice: Robot B still sees the live refusal through direct confirmation	Pass
9	No confirmation available: the proposed rule pauses although permission is valid	Pass
10	Refusal mid-lift: safe stop holds the resident with zero further progress until handover	Pass
11	Handover cannot complete while the lift is still moving	Pass
12	Emergency stop cancels task progress in both tracks	Pass
13	Reset before handover is rejected; reset after handover creates no movement or permission	Pass
14	Explicit restart: operator decision plus fresh confirmation starts a new lift	Pass
15	Queue: tick-4 completions, urgent start ticks and routine starts match the case	Pass
16	Queue decisions are unchanged when only the communication-device label moves · Identical across all 6 placements	Pass
17	Fleet outage: standby staff meet deadlines 3, 4, 5; without them all three are missed	Pass
18	Every case is deterministic and both tracks start from the same initial state	Pass
19	English, German and Spanish dictionaries have the same keys and placeholders · 643 keys in each language	Pass
20	Language changes text only: every decision, state and metric value is identical	Pass
21	Every generated text resolves in every language (no missing translations)	Pass

Browser checks performed

- All four cases played and stepped to the end in English, German and Spanish, with every expandable section open, and no untranslated text found.
- Dark mode on first load; light mode toggled and restored; the standalone page background follows the theme.
- Playback: repeated Play presses, switching cases during playback, changing settings during playback and switching language during playback. Never more than one timer, and case or setting changes pause and restart at tick 0.
- Keyboard only: language menu, theme toggle, case picker with arrow keys, settings, prediction, Step with Enter and Space, with a visible focus outline.
- Widths of 320, 375, 541 and 1280 pixels with no horizontal page scroll.
- The WordPress fragment inside a mock blog page with aggressive theme CSS (uppercase pink buttons, bordered tables, decorated headings, hidden radios, a sticky site header). Theme styles did not reach the lab, the lab did not change theme elements, and the model checks passed inside the page.

Not verified: a real WordPress installation and its plugins, Safari and Firefox, and testing with screen readers. These are recommended before publishing.

13. WordPress integration

Option A: Custom HTML block (recommended)

- Open the post or page in the block editor and add a **Custom HTML** block where the lab should appear.
- Open `wordpress-embed.html` in a plain text editor, select everything, copy it and paste it into the block.
- Use Preview to check it, then publish. For the two-column layout on large screens, place the block inside a Group block set to “Wide width” or “Full width”; in a normal text column the lab uses its single-column layout.
- Use one lab per page: the element ids are fixed.

Requirement. WordPress keeps `<script>` and `<style>` only for users with the `unfiltered_html` capability: administrators on a single site, and only super administrators on a multisite network. For other users WordPress strips them on save and the lab will not run. If that applies to your site, use option B.

Optional settings on the first line of the fragment

Attribute	Values	Effect
data-lang	en (default), de, es	Starting language for viewers who have not chosen one.
data-theme	dark (default), light	Starting theme for viewers who have not chosen one.
data-case	lift, fleet, queue, outage	Opens another case first (also possible with #wgsr-fleet in the page address).
style="--wgsr-sticky-top: 80px"	any length	Moves the sticky playback bar below a sticky site header or the admin bar.

```
<div id="wgsr" class="wgsr" data-theme="dark" data-lang="de" data-embed="wordpress" lang="en"
style="--wgsr-sticky-top: 80px">
```

Why it is safe inside a theme

- Every CSS selector is prefixed with #wgsr, so the lab cannot change the theme, and its id-level specificity beats ordinary theme rules for buttons, tables, headings, lists and form controls.
- Layout breakpoints respond to the lab’s own width, not the browser window, so a narrow content column gets the narrow layout.
- The fragment contains no html, head or body tags, no blank lines (which the classic editor could turn into paragraphs) and no text that WordPress would treat as a shortcode. The build checks all three.

- The scripts carry data-cfasync="false", data-no-optimize="1", data-no-defer="1" and data-no-minify="1" to opt out of common caching and optimization plugins.

Option B: iframe with the standalone page

Upload `index.html` to your server, for example with SFTP or your host's file manager (the WordPress media library rejects .html files by default), then embed it. This isolates the lab completely, but the frame needs a fixed height.

```
<iframe src="https://example.com/lab/index.html" title="Who gets to stop a robot?" loading="lazy"
style="width:100%;height:1700px;border:0"></iframe>
```

Troubleshooting

Symptom	Likely cause	Fix
Only static text, nothing reacts	Scripts were stripped on save (no unfiltered_html), or a plugin delays inline JavaScript	Paste as an administrator, exclude scripts whose id starts with "wgsr-" from optimization, or use option B
The playback bar hides under the site header	The theme has a sticky header	Set --wgsr-sticky-top on the #wgsr element
Single-column layout on a wide screen	The content column is narrow	Put the block in a Group block with wide or full width
Buttons or tables look like the theme's	The theme uses !important rules	Add a matching #wgsr rule with !important in the theme's custom CSS, or use option B
Preferences are not remembered	Browser storage is blocked (private mode)	No fix needed: the lab still works with the defaults

14. Assumptions and limitations

- This is a finite, symbolic teaching model with synthetic data. It is not engineering, clinical, safety-certification or legal guidance.
- The harm-priority planner's prediction is fixed (finishing is predicted safer once the resident is off the bed). Real planners and risks vary.
- In case 2 the direct channel, when available, always answers truthfully, there is no renewed consent in the benchmark, and notices are either delivered after a fixed delay or lost.
- A fresh confirmation in case 1 must follow the latest refusal or stop and be no more than 3 ticks old. That window is a modelling choice.
- Case 3 has one worker and no preemption. Case 4 has one standby person and one robot handling the list, and the authorization time is fixed.
- The 32-rule result (at most 16 authorized starts with zero unwanted starts) holds for these five observable pairs and this enumeration only.
- The Asimov FAQ could not be loaded when sources were checked (the site's security certificate did not match its address), so its date is not verified here.

15. Sources

Source	Date	Used for
Edward Seiler and John H. Jenkins, “Isaac Asimov FAQ”, Asimov Online. https://www.asimovonline.com/asimov_FAQ.html	Living web FAQ, no single publication date	The Three Laws and the Zeroth Law
Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel and Stuart Russell, “The Off-Switch Game”, Proceedings of IJCAI-17, pp. 220–227. https://www.ijcai.org/proceedings/2017/32	August 2017	When an agent has reason to let a person switch it off
J. Tanner Slagel, Lauren M. White, Aaron Dutle, César A. Muñoz and Nicolas Crespo, “A Verification Framework for Runtime Assurance of Autonomous UAS”, NASA Langley Research Center, 43rd Digital Avionics Systems Conference. https://ntrs.nasa.gov/citations/20240007986	Conference 29 Sep – 3 Oct 2024 (record acquired 24 Jun 2024)	An independent monitor hands control to a trusted fallback
National Institute of Standards and Technology, “Artificial Intelligence Risk Management Framework (AI RMF 1.0)”, NIST AI 100-1. https://nvlpubs.nist.gov/nistpubs/AI/NIST.AI.100-1.pdf	January 2023	Governance and practical human oversight