STPA-Spec

A human- and machine-readable language for STPA analysis

STAMP Workshop 2026

https://satellogic.github.io/stpa-spec/

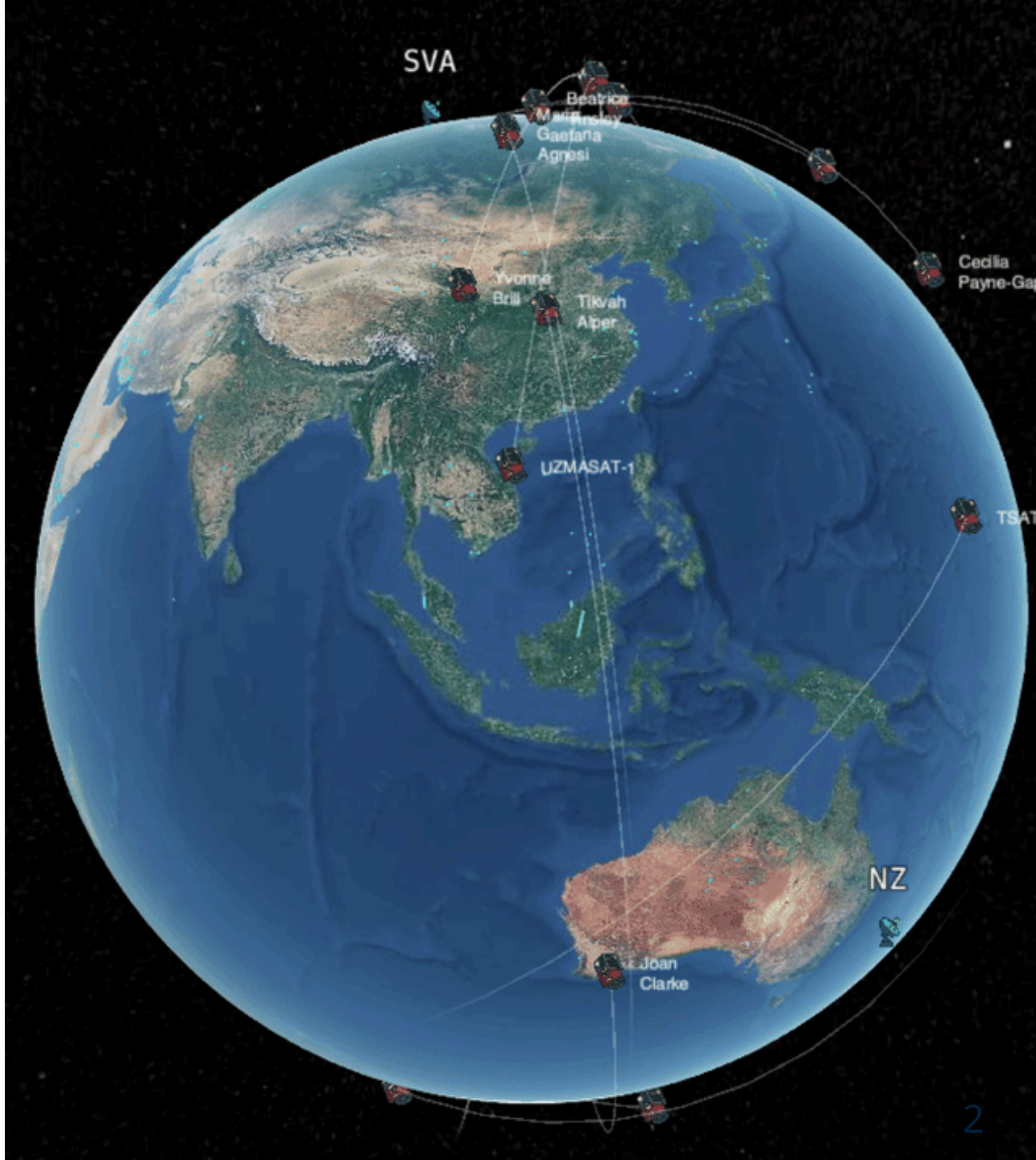
SATELLOGIC

Alan Kharsansky (alan@satellogic.com)
Manuel Diaz Ramos (manuel.diaz.ramos@satellogic.com)



Context

- Satellogic develops **remote sensing infrastructure** that enables continuous, high-frequency global monitoring.
- We **design**, **manufacture**, and **operate** earth observation satellites.
- Satellogic ❤️ STPA. We have been using it extensively in our safety analyses for some time now, with **excellent results**.



So, is there a problem?

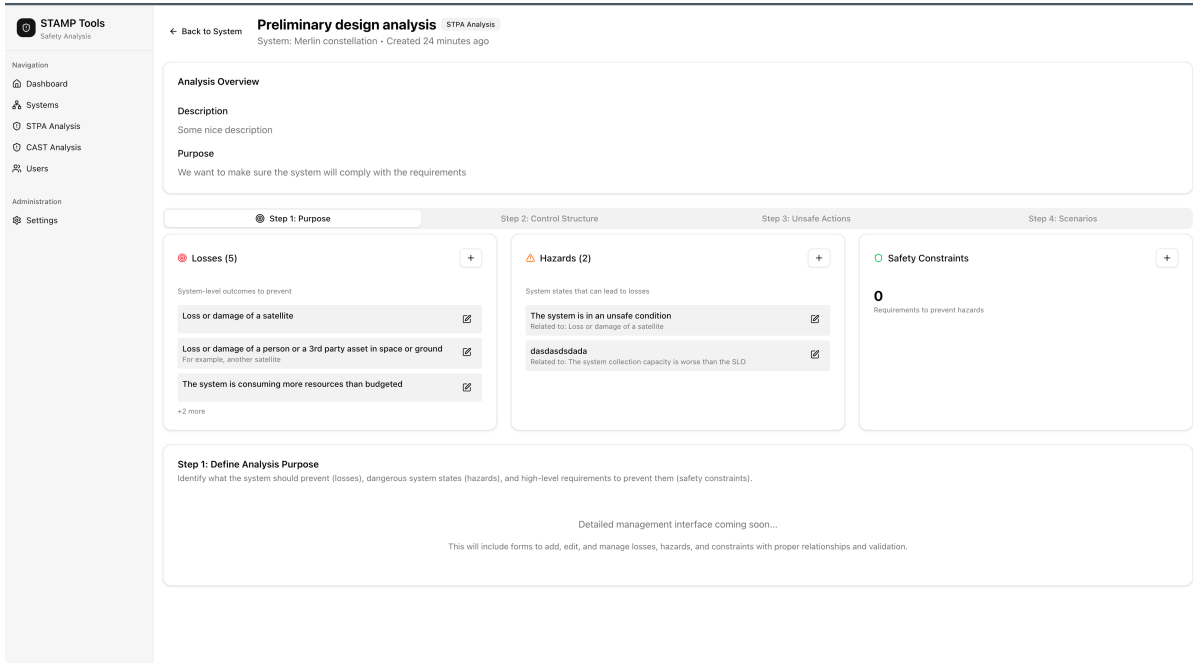
- Every team was documenting it **differently**
- Analyses were difficult to **share**, especially across **distributed teams**
- Spreadsheets and docs were becoming **unmanageable**



The failed "AHA!" moment

We should create a tool!

- Everyone can vibe-code a tool in a weekend these days (we know you have tried it!)
- Creating a reliable and useful tool requires effort, time, and domain knowledge. Weekend prototypes typically fail quickly.
- There are plenty of tools available (and they are great!), but none are *exactly what you need*.



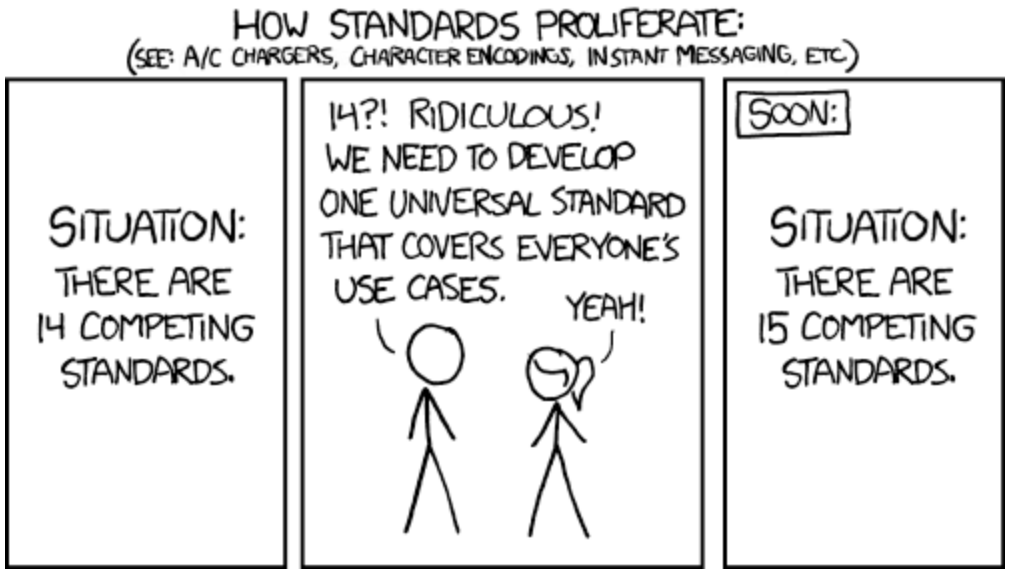
Real failed STPA tool attempt

So we went back to square one...

The need of an STPA Specification

That:

- Defines a consistent structure for STPA artifacts
- Is human- and machine-readable
- Facilitates sharing and collaboration
- Supports traceability and validation



https://xkcd.com/927/

> Specifications become standards once they are adopted and actually used.

The proposal

A text-based structured specification for documenting STPA artifacts

- Human-readable
- Machine-readable
- Linkable and extensible
- Leverages Git for versioning, collaboration, and change tracking
- Open-source and community-maintained

> This was inspired by the successful STAC standard for geospatial catalogs. See https://stacspec.org for more information.

Example Loss in JSON and YAML

```json
{
  "stpa_spec_version": "1.0.0",
  "type": "artifact_collection",
  "title": "737 MAX investigation",
  "description": "Defined losses for the STPA analysis",
  "artifacts": [
    {
      "id": "L-0001",
      "type": "loss",
      "title": "Loss of life or injury to people",
      "status": "accepted",
      "links": []
    },
    {
      "id": "L-0002",
      "type": "loss",
      "title": "Loss of or damage to vehicle ",
      "status": "accepted"
    },
    ...
  ]
}
```

```yaml
stpa_spec_version: 1.0.0
type: artifact_collection
title: 737 MAX investigation
description: Defined losses for the STPA analysis
artifacts:
  - id: L-0001
    type: loss
    title: Loss of life or injury to people
    status: accepted
    links: []
  - id: L-0002
    type: loss
    title: 'Loss of or damage to vehicle '
    status: accepted
  - id: L-0003
    type: loss
    title: Loss of or damage to objects outside the vehicle
    status: accepted
    links: []
```

The syntax is simple; the important property is explicit structure and traceability.

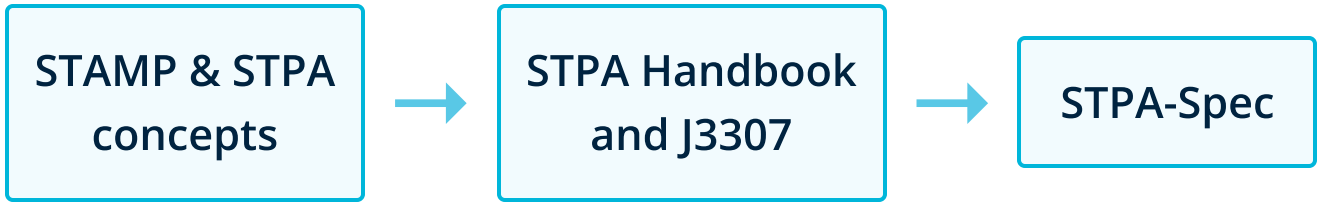
More examples (YAML format)

```yaml
stpa_spec_version: 1.0.0
id: H-0001
type: hazard
title: 'Aircraft violate minimum separation standards in flight'
status: proposed
links:
- rel: leads_to
  to: L-0001
- rel: leads_to
  to: L-0002
```

```yaml
stpa_spec_version: "1.0.0"
id: UCA-0001
type: uca
title: Fails to close V-12 on high pressure
uca_type: not-provided
links:
  - rel: source
    to: CTR-0001
  - rel: applies_to
    to: CA-0001
  - rel: hazard
    to: H-0001
```

These artifacts can live in the same file as part of an artifact collection or be split across files. Either works.

So, what it is and what it is not?



The specification:

- Uses **established STPA terminology** by The Handbook and the SAE J3307 standard.[1].
- Defines how artifacts produced during the STPA process should be **written, organized and documented**
- Does **not redefine** the underlying concepts
- Does **not prescribe** one analysis tool
- Provides aids for validation, documentation and exchange.

[1] SAE J3307 is a registered trademark of SAE International. This product is independently developed by Satellogic and is not affiliated with, sponsored by, or endorsed by SAE International.

Git-based version control

- Complete history
- Branches for design alternatives
- Pull requests for multidisciplinary review
- Tags for approved baselines
- Change attribution

A repository as an analysis workspace

```
satellite-early-orbits/
├── losses.yaml
├── hazards.yaml
├── control-structure.yaml
├── control-actions.yaml
├── unsafe-control-actions.yaml
├── scenarios/
├── requirements/
├── tests/
├── evidence/
└── README.md
```

A semantic change becomes visible:

```
- SC-001: Panels shall deploy after separation.
+ SC-001: Panels shall deploy after separation and after
+          the post-separation inhibit interval, unless a
+          valid inhibit is active.
```

Now we can build the tools



Tools can then provide:

- Editors
- Validators
- Visualizations
- Reports
- Requirements integrations
- Safety-case integrations
- AI assistants

Tools should expose the analysis—not define what it means.

Example: What can be checked automatically?

A validator can detect:

- Duplicate identifiers
- Missing required fields
- Broken references
- Invalid artifact types
- Hazards without loss references
- UCAs without control actions
- Requirements without source constraints
- Tests without linked requirements
- Evidence without an associated claim

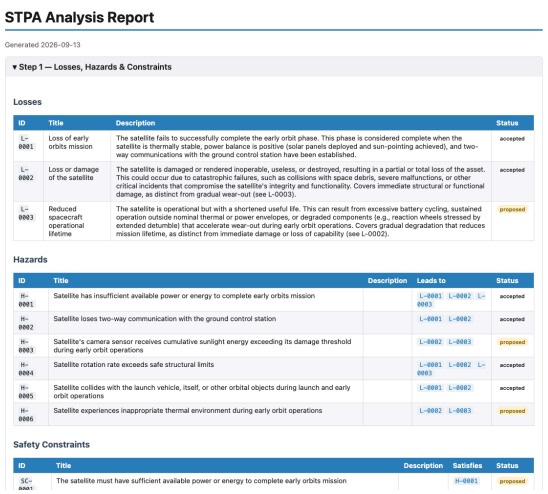
> Validation improves consistency. It does not prove that the analysis is adequate.

Example: The CLI tool

(Already provided in the repository)

```
$ python tools/cli/stpa.py validate
$ python tools/cli/stpa.py graph . --render png
$ python tools/cli/stpa.py convert losses.json --to-yaml --output losses.yaml
$ python tools/cli/stpa.py stats .
$ python tools/cli/stpa.py report .
```

Generates docs and diagrams, and validates analysis structure…


STPA Analysis Report

Generated 2026-09-13

▾ Step 1 — Losses, Hazards & Constraints

Losses

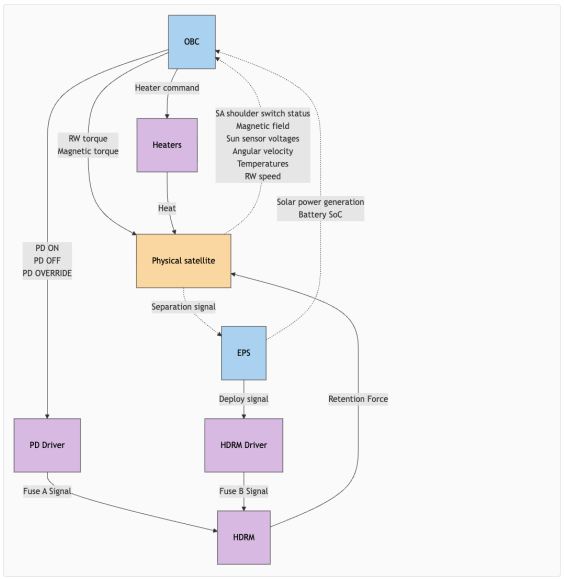
ID	Title	Description	Status
L-0001	Loss of early orbits mission	The satellite fails to successfully complete the early orbit phase. This phase is considered complete when the satellite is thermally stable, power balance is positive (solar panels deployed and sun-pointing achieved), and two-way communications with the ground control station have been established.	accepted
L-0002	Loss or damage of the satellite	The satellite is damaged or rendered inoperable, useless, or destroyed, resulting in a partial or total loss of the asset. This could occur due to catastrophic failures, such as collisions with space debris, severe malfunctions, or other critical incidents that compromise the satellite's integrity and functionality. Covers immediate structural or functional damage, as distinct from gradual wear-out (see L-0003).	accepted
L-0003	Reduced spacecraft operational lifetime	The satellite is operational but with a shortened useful life. This can result from excessive battery cycling, sustained operation outside nominal thermal or power envelopes, or degraded components (e.g., reaction wheels stressed by extended detumble) that accelerate wear-out during early orbit operations. Covers gradual degradation that reduces mission lifetime, as distinct from immediate damage or loss of capability (see L-0002).	proposed

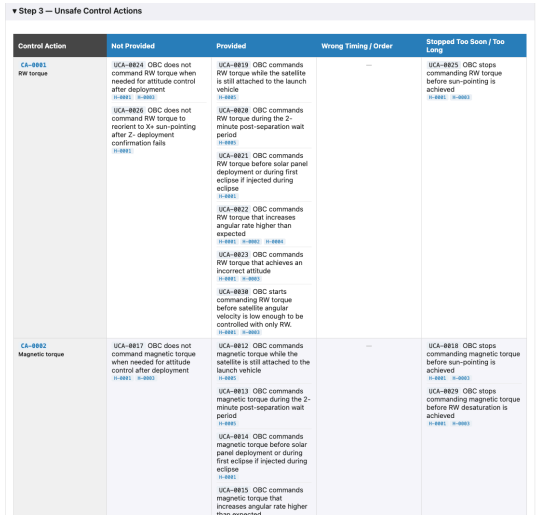
Hazards

ID	Title	Description	Leads to	Status
H-0001	Satellite has insufficient available power or energy to complete early orbits mission		L-0001 L-0002 L-0003	accepted
H-0002	Satellite loses two-way communication with the ground control station		L-0001 L-0002	accepted
H-0003	Satellite's camera sensor receives cumulative sunlight energy exceeding its damage threshold during early orbit operations		L-0002 L-0003	proposed
H-0004	Satellite rotation rate exceeds safe structural limits		L-0001 L-0002 L-0003	accepted
H-0005	Satellite collides with the launch vehicle, itself, or other orbital objects during launch and early orbit operations		L-0001 L-0002	accepted
H-0006	Satellite experiences inappropriate thermal environment during early orbit operations		L-0002 L-0003	proposed

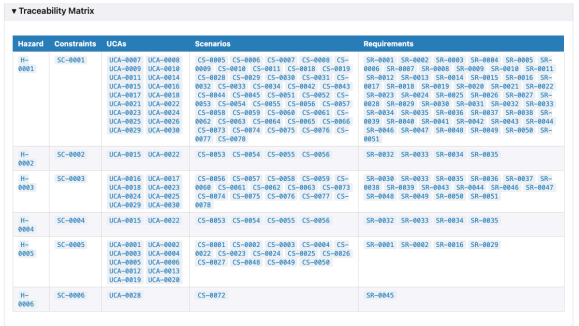
Safety Constraints

ID	Title	Description	Satisfies	Status
SC-0001	The satellite must have sufficient available power or energy to complete early orbits mission		H-0001	proposed




▾ Step 3 — Unsafe Control Actions

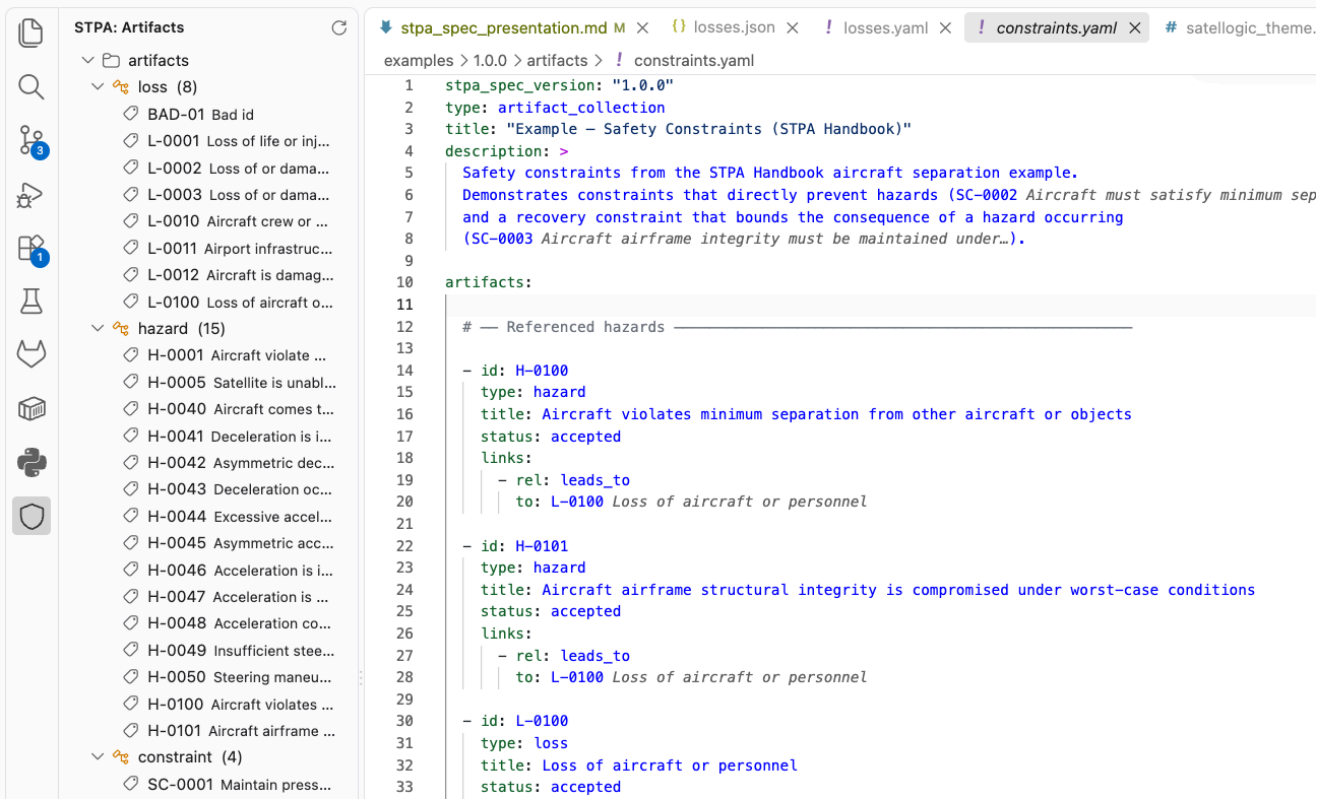
Control Action	Not Provided	Provided	Wrong Timing / Order	Stopped Too Soon / Too Long
CA-0001 RW torque	UCA-0024 OBC does not command RW torque when needed for attitude control after deployment H-0001 H-0002 UCA-0026 OBC does not command RW torque to reorient to X+ sun-pointing after Z- deployment confirmation fails H-0001	UCA-0019 OBC commands RW torque while the satellite is still attached to the launch vehicle H-0005 UCA-0020 OBC commands RW torque during the 2-minute post-separation wait period H-0005 UCA-0021 OBC commands RW torque before solar panel deployment or during first eclipse if injected during eclipse H-0001 UCA-0022 OBC commands RW torque that increases angular rate higher than expected H-0001 H-0002 H-0004 UCA-0023 OBC commands RW torque that achieves an incorrect attitude H-0001 H-0005 UCA-0030 OBC starts commanding RW torque before satellite angular velocity is low enough to be controlled with only RW. H-0001 H-0005	—	UCA-0025 OBC stops commanding RW torque before sun-pointing is achieved H-0001 H-0003
CA-0002 Magnetic torque	UCA-0017 OBC does not command magnetic torque when needed for attitude control after deployment H-0001 H-0003	UCA-0012 OBC commands magnetic torque while the satellite is still attached to the launch vehicle H-0005 UCA-0013 OBC commands magnetic torque during the 2-minute post-separation wait period H-0005 UCA-0014 OBC commands magnetic torque before solar panel deployment or during first eclipse if injected during eclipse H-0001 UCA-0015 OBC commands magnetic torque that increases angular rate higher	—	UCA-0018 OBC stops commanding magnetic torque before sun-pointing is achieved H-0001 H-0003 UCA-0029 OBC stops commanding magnetic torque before RW desaturation is achieved H-0001 H-0003


▾ Traceability Matrix

Hazard	Constraints	UCAs	Scenarios	Requirements
H-0001	SC-0001	UCA-0007 UCA-0008 UCA-0009 UCA-0010 UCA-0011 UCA-0014 UCA-0015 UCA-0016 UCA-0017 UCA-0018 UCA-0021 UCA-0022 UCA-0023 UCA-0024 UCA-0025 UCA-0026 UCA-0029 UCA-0030	CS-0005 CS-0006 CS-0007 CS-0008 CS-0009 CS-0010 CS-0011 CS-0018 CS-0019 CS-0026 CS-0029 CS-0030 CS-0031 CS-0032 CS-0033 CS-0034 CS-0042 CS-0043 CS-0044 CS-0045 CS-0051 CS-0052 CS-0053 CS-0054 CS-0055 CS-0056 CS-0057 CS-0058 CS-0059 CS-0060 CS-0061 CS-0062 CS-0063 CS-0064 CS-0065 CS-0066 CS-0073 CS-0074 CS-0075 CS-0076 CS-0077 CS-0078	SR-0001 SR-0002 SR-0003 SR-0004 SR-0005 SR-0006 SR-0007 SR-0008 SR-0009 SR-0010 SR-0011 SR-0012 SR-0013 SR-0014 SR-0015 SR-0016 SR-0017 SR-0018 SR-0019 SR-0020 SR-0021 SR-0022 SR-0023 SR-0024 SR-0025 SR-0026 SR-0027 SR-0028 SR-0029 SR-0030 SR-0031 SR-0032 SR-0033 SR-0034 SR-0035 SR-0036 SR-0037 SR-0038 SR-0039 SR-0040 SR-0041 SR-0042 SR-0043 SR-0044 SR-0046 SR-0047 SR-0048 SR-0049 SR-0050 SR-0051
H-0002	SC-0002	UCA-0015 UCA-0022	CS-0053 CS-0054 CS-0055 CS-0056	SR-0032 SR-0033 SR-0034 SR-0035
H-0003	SC-0003	UCA-0016 UCA-0017 UCA-0018 UCA-0023 UCA-0024 UCA-0025 UCA-0029 UCA-0030	CS-0056 CS-0057 CS-0058 CS-0059 CS-0060 CS-0061 CS-0062 CS-0063 CS-0073 CS-0074 CS-0075 CS-0076 CS-0077 CS-0078	SR-0030 SR-0033 SR-0035 SR-0036 SR-0037 SR-0038 SR-0039 SR-0043 SR-0044 SR-0046 SR-0047 SR-0048 SR-0049 SR-0050 SR-0051
H-0004	SC-0004	UCA-0015 UCA-0022	CS-0053 CS-0054 CS-0055 CS-0056	SR-0032 SR-0033 SR-0034 SR-0035
H-0005	SC-0005	UCA-0001 UCA-0002 UCA-0003 UCA-0004 UCA-0005 UCA-0006 UCA-0012 UCA-0013 UCA-0019 UCA-0020	CS-0001 CS-0002 CS-0003 CS-0004 CS-0022 CS-0023 CS-0024 CS-0025 CS-0026 CS-0027 CS-0048 CS-0049 CS-0050	SR-0001 SR-0002 SR-0016 SR-0029
H-0006	SC-0006	UCA-0028	CS-0072	SR-0045

Example: The VS Code extension

- A simple extension to browse STPA analyses.
- Autocomplete
- Syntax validation
- Quick navigation between related artifacts

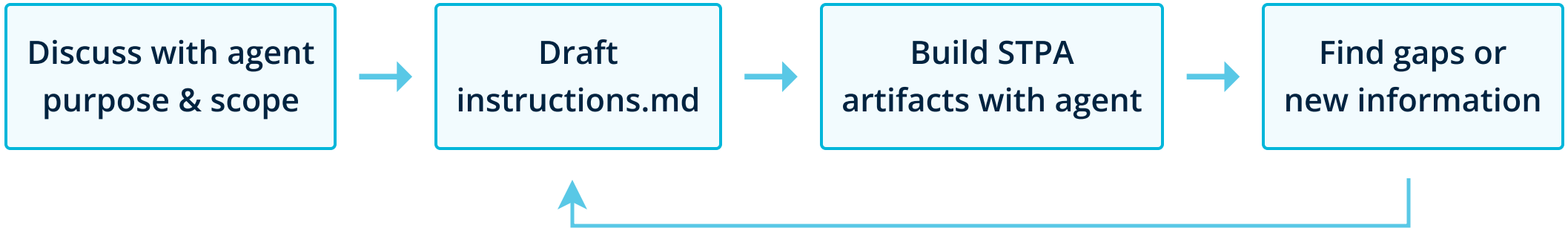


Working with AI in the loop



- The skill is generic and public, included in the stpa-spec repository.
- Reading it is what turns the agent into an experienced STPA engineer that will help you in your analysis.
- The analysis repo can submodule the stpa-spec repository.
- The instructions file is specific to the system being analyzed, and lives in that analysis' own repository.

The instructions file keeps evolving



- The first pass sets **purpose and boundaries** of the analysis
- Later passes add facts we forgot, got wrong, or never mentioned
- The instructions file becomes a living, versioned system description
- Don't wait for a complete instructions file — start building artifacts early and keep iterating.

The instructions file grows with the analysis — it's never really "done". It's not only a machine-readable file but also a human-oriented documentation of the problem.

A representative early-orbit example

Analysis scope:

> From separation from the launch vehicle until the satellite reaches a stable, power-positive, communicating state.

- Deployment is triggered by **two independent methods** that don't coordinate with each other (this was added by STPA)
- Both methods act on the **same one-shot mechanism**, so either one can deploy it
- No attitude precondition gates deployment
- Without the deployable elements, the fixed power source alone may not sustain the satellite once attitude control is active
- Full deployment confirmation is **indirect**: partial switches plus a sensor pattern plus a power increase
- The same command can be safe or unsafe depending purely on timing and system state — not on any component failure
- All of this happens **autonomously**, with no ground commanding

From losses to ground tests



The analysis produced two concrete outputs:

- An **actual redesign of the autonomous safe mode** — the uncoordinated trigger was a design gap surfaced by STPA, not a pre-existing requirement
- A set of **ground tests**, run before launch, to verify the hazards cannot occur

STPA earns its keep when it changes the design and produces test procedures engineers actually run.

The traceability matrix, generated

▼ Traceability Matrix

Hazard	Constraints	UCAs	Scenarios	Requirements
H-0001	SC-0001	UCA-0007 UCA-0008 UCA-0009 UCA-0010 UCA-0011 UCA-0014 UCA-0015 UCA-0016 UCA-0017 UCA-0018 UCA-0021 UCA-0022 UCA-0023 UCA-0024 UCA-0025 UCA-0026 UCA-0029 UCA-0030	CS-0005 CS-0006 CS-0007 CS-0008 CS-0009 CS-0010 CS-0011 CS-0018 CS-0019 CS-0028 CS-0029 CS-0030 CS-0031 CS-0032 CS-0033 CS-0034 CS-0042 CS-0043 CS-0044 CS-0045 CS-0051 CS-0052 CS-0053 CS-0054 CS-0055 CS-0056 CS-0057 CS-0058 CS-0059 CS-0060 CS-0061 CS-0062 CS-0063 CS-0064 CS-0065 CS-0066 CS-0073 CS-0074 CS-0075 CS-0076 CS-0077 CS-0078	SR-0001 SR-0002 SR-0003 SR-0004 SR-0005 SR-0006 SR-0007 SR-0008 SR-0009 SR-0010 SR-0011 SR-0012 SR-0013 SR-0014 SR-0015 SR-0016 SR-0017 SR-0018 SR-0019 SR-0020 SR-0021 SR-0022 SR-0023 SR-0024 SR-0025 SR-0026 SR-0027 SR-0028 SR-0029 SR-0030 SR-0031 SR-0032 SR-0033 SR-0034 SR-0035 SR-0036 SR-0037 SR-0038 SR-0039 SR-0040 SR-0041 SR-0042 SR-0043 SR-0044 SR-0046 SR-0047 SR-0048 SR-0049 SR-0050 SR-0051
H-0002	SC-0002	UCA-0015 UCA-0022	CS-0053 CS-0054 CS-0055 CS-0056	SR-0032 SR-0033 SR-0034 SR-0035
H-0003	SC-0003	UCA-0016 UCA-0017 UCA-0018 UCA-0023 UCA-0024 UCA-0025 UCA-0029 UCA-0030	CS-0056 CS-0057 CS-0058 CS-0059 CS-0060 CS-0061 CS-0062 CS-0063 CS-0073 CS-0074 CS-0075 CS-0076 CS-0077 CS-0078	SR-0030 SR-0033 SR-0035 SR-0036 SR-0037 SR-0038 SR-0039 SR-0043 SR-0044 SR-0046 SR-0047 SR-0048 SR-0049 SR-0050 SR-0051
H-0004	SC-0004	UCA-0015 UCA-0022	CS-0053 CS-0054 CS-0055 CS-0056	SR-0032 SR-0033 SR-0034 SR-0035
H-0005	SC-0005	UCA-0001 UCA-0002 UCA-0003 UCA-0004 UCA-0005 UCA-0006 UCA-0012 UCA-0013 UCA-0019 UCA-0020	CS-0001 CS-0002 CS-0003 CS-0004 CS-0022 CS-0023 CS-0024 CS-0025 CS-0026 CS-0027 CS-0048 CS-0049 CS-0050	SR-0001 SR-0002 SR-0016 SR-0029
H-0006	SC-0006	UCA-0028	CS-0072	SR-0045

Takeaways

- **Semantics first:** represent what STPA artifacts mean and how they relate.
- **Structured language:** provide a common format compatible with established STPA practice, the Handbook, and J3307.
- **Text-based:** keep artifacts human-readable and machine-readable.
- **AI-enabled:** support grounded queries, traceability, and AI assistance in the process of building the model.
- **Git-based:** make analysis history, alternatives, and semantic changes reviewable.
- **Tools came later:** build editors, validators, visualizations, and integrations around the specification.

Make STPA analysis portable, reviewable, and reusable.

Thank you

Use it, contribute, modify it, propose changes, and create tools based on STPA-Spec

https://github.com/satellogic/stpa-spec

Reference material is available in the live docs: https://satellogic.github.io/stpa-spec/