ADAPT — Secondary-Data Pilot on a Real, Independent SoS Dataset

A worked pilot that runs ADAPT's actual machinery on a real, public system-of-systems dataset the author did not produce — directly attacking the "single-source, self-reported" evidence gap, with no new fieldwork and no classified data.

Dataset

Dronology / SAFA-Artifacts (Agrawal, Khoshmanesh, Vierhauser, Rahimi, Cleland-Huang, Lutz, *ICSE 2019*) — public on GitHub (`SAREC-Lab/SAFA-Artifacts`) with a Zenodo DOI. **Dronology is itself a system-of-systems** coordinating multiple cooperating UAVs. The dataset ships **two versions (V0, V1)** of artifact trees in which hazards, requirements, design definitions, code, acceptance tests, and environmental assumptions are linked by **trace links**, each artifact carrying a `status` and an `is-delegated` (cross-boundary) flag. It is independent of ADAPT, and from **Jane Cleland-Huang's group — already reference [20] in the manuscript.**

Method — ADAPT constructs mapped to dataset measures

ADAPT construct	Measured as
Capability need / anchor	Root hazard of each tree
Prerequisite enforcement / branchless	Share of artifacts with no upstream trace to a root (branchless rate)
Critical-path structure	Longest dependency chain (max depth)
Cross-constituent dependency	`is-delegated` links
Definition of Done / maturity	Artifact `status` + completeness
Change propagation / blast radius	V0→V1 delta: downstream descendants of each changed/removed artifact, via the trace graph

`pilot_analysis.py` (stdlib only) reproduces everything below from the two JSON files.

Results

Measure	V0	V1
Roots (hazards = capability needs)	16	16
Artifacts	240	360
Trace links	234	355
Max dependency-chain	6	8

depth (critical path)		
Trace-completeness (1 – branchless)	**100% (0 branchless)**	**100% (0 branchless)**
Cross-boundary (`delegated`) dependencies	9	9
Artifact types/levels	13 (Code 73, EnvAssumption 45, Package 37, Design 29, Hazard 16, SafetyReq 14, …)	13 (Code 155, Package 68, …)

Evolution V0→V1: +205 added, −85 removed (240→360 artifacts), depth 6→8.
Change-impact via the trace graph: for each changed/removed artifact, ADAPT's graph localizes the downstream set — blast radius mean 0.2, **max 3**, 14 downstream impacts traced (most growth was new leaf code/packages under existing requirements, hence small radii).

What this validates — and what it does not

Validates (on real, independent SoS data): ADAPT's dependency/traceability constructs are *computable and meaningful*. The **0-branchless / 100%-trace-complete** invariant is precisely the target state ADAPT's prerequisite-enforcement drives toward; **critical-path depth** is measurable and grows with system evolution (6→8); **cross-boundary delegated dependencies** are identifiable (9); **DoD/maturity** discriminates; and **change-impact is read off the trace graph** (ADAPT §4.6's central claim), with a real maximum blast radius of 3 artifacts.

Does not validate the recovery *outcome* (faster time-to-baseline) — no public dataset records that; this is **construct/mechanism validation**, complementary to the §7 prospective study.

Honest caveats: (a) this is a curated safety-case tree, so 100% trace-completeness is the *target* state, not a stalled one — ADAPT's value is moving stalled programs toward it; (b) the `-simplified` export blanks many descriptions, so a raw "incomplete description" count would illustrate the *check*, not true maturity, and is omitted from claims; (c) V0→V1 deltas are mostly leaf additions, so blast radii are modest.

How to use it in the paper

Add a short **"External construct validation"** subsection (or cite this as Supplementary): *ADAPT's machinery, applied to an independent, public SoS dataset (Dronology/SAFA), yields the expected, interpretable governance measures and localizes change impact on the trace graph — evidence for the mechanisms that is independent of the author's four cases.* It is fully reproducible (`pilot_analysis.py` + the two JSON files).

Sources: SAFA-Artifacts — GitHub, Zenodo DOI badge in repo; Agrawal et al., *Leveraging Artifact Trees to Evolve and Reuse Safety Cases*, ICSE 2019.