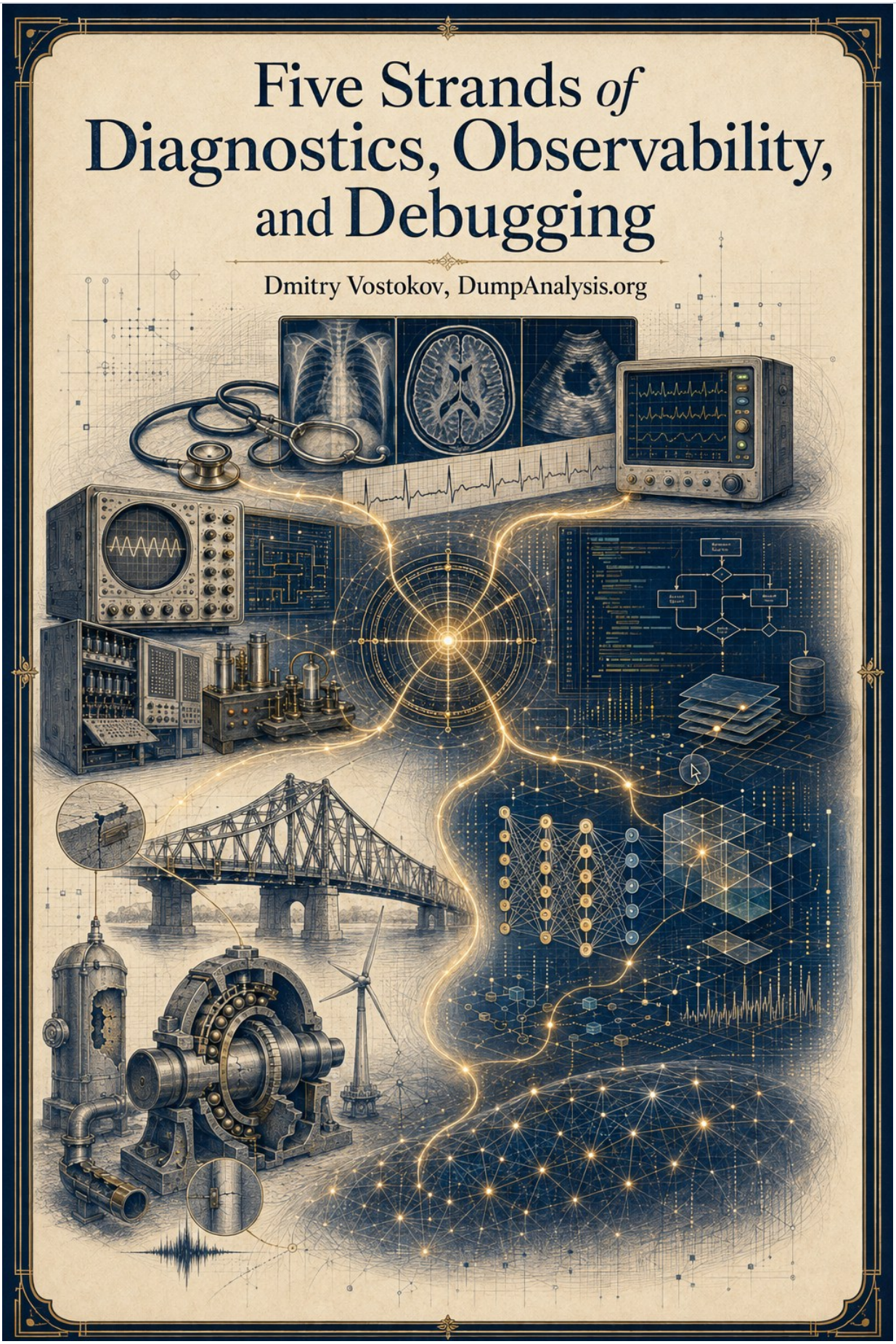
Five Strands *of* Diagnostics, Observability, and Debugging

Dmitry Vostokov, DumpAnalysis.org



A TECHNOLOGICAL AND INTELLECTUAL SYNTHESIS

Companion overview to the five linked histories

Concept and direction by Dmitry Vostokov, DumpAnalysis.org

Developed with AI assistance from GPT-6 Astra Max and Claude Fable 5.1 Extra. Dmitry Vostokov set the scope and argument, selected the source base, directed all revisions, and reviewed the manuscript. AI assisted with drafting, comparison, and copy-editing; factual and technical claims added during this revision were checked against the cited sources and the five companion histories.

September 2026 · Version 20

Abstract

Let us consider five histories of diagnostics, observability, and debugging: software, hardware, medical, ML and AI, and technical. Each history has its own diagnostic objects, instruments, and standards of evidence. Nevertheless, we encounter a recurrent situation in all five: we observe only part of a condition, several explanations fit the available evidence, and we need to decide what to examine or do next [1–5].

We begin with the objects themselves. Medical diagnosis concerns a person and a decision about care. Hardware diagnosis follows physical faults through circuits, encoded state, and service records. In software, we reconstruct execution from memory dumps, traces, logs, and other artifacts. ML and AI add data, learned behavior, evaluation, feedback, and agent actions. Technical diagnosis relates physical observations to the condition and integrity of an asset. These differences determine which comparisons are useful and where an analogy ends.

For our purposes, we organize diagnostic work into eight stages: Observe; Measure / preserve; Contextualize; Compare; Infer; Discriminate; Act; Verify and learn (Table 5). Sometimes we return to an earlier stage; sometimes we have to act before the mechanism is known. We then consider analysis patterns as a possible language for the operations performed on evidence. The proposal concerns how we investigate different objects. Its usefulness depends on whether an operation can be transferred with its evidence requirements, verification, and limits intact. Appendix C examines four such transfers, including a case that joins hardware evidence to an ML training regression [1–5].

How to read this synthesis

The five histories start from different practices. Medical case records preserve observations and their development over time. Technical and hardware diagnosis draw on mechanics, measurement, electrical testing, reliability, and maintenance. Stored programs and production systems supply new artifacts for software diagnosis. ML and AI bring together statistical model criticism, symbolic explanation, learning theory, software operations, and governance. We may find similar methods in several histories, although

they developed independently. A common name alone tells us little about their origin or the obligations attached to their use [1–5].

The companion histories provide the detailed historical record. We refer to them when summarizing a strand and use papers, standards, and official specifications for more specific comparisons. Chapters 26–30 also examine the pattern-oriented work from which our proposal develops. We present cross-strand mappings as analytical proposals: their use in a clinical or engineering investigation requires the original evidence and the qualifications of that domain.

Five linked histories

Each strand is developed in depth in its own companion history. The titles below are live links.

Table 1. The five linked companion histories.

STRAND	PRIMARY OBJECT	CHARACTERISTIC EVIDENCE	DISTINCTIVE EMPHASIS
Medical diagnostics	Person, body, disease process, care pathway	History, examination, specimens, images, waveforms, outcomes	Patient context, probability, ethics, communication, and clinical utility
Hardware diagnostics	Electronic and computing devices across physical and logical layers	Waveforms, test patterns, syndromes, counters, telemetry, physical failure analysis	Fault containment, first-failure capture, cross-layer matter and representation
Software diagnostics	Programs, executions, services, and organizations	Dumps, logs, metrics, traces, profiles, code, configuration	Reconstructing invisible and often distributed execution under partial evidence
ML and AI diagnostics	Data-model-pipeline-service-human systems	Residuals, splits, gradients, explanations, drift, evaluations, trajectories	Systems can fail while executing correctly; behavior is statistical, adaptive, and governed
Technical diagnostics	Physical structures, machines, plants, and engineered assets	NDE, strain, vibration, thermography, condition histories, SHM, failure analysis	Integrity, degradation physics, qualification, and consequence-based intervention

Neighboring practices and their primary questions

Table 2. A unified working vocabulary; real investigations combine several practices.

PRACTICE	PRIMARY QUESTION	CHARACTERISTIC EVIDENCE
Observation / inspection	What can be directly noticed, elicited, or measured now?	Signs, dimensions, images, waveforms, physical condition, narratives
Testing / measurement	Does the object, specimen, model, or system satisfy a stated expectation under defined conditions?	Stimuli, reference standards, test vectors, assays, limits, calibration records
Monitoring	How is selected condition changing, and when does it require attention?	Trends, telemetry, repeated specimens, counters, alarms, trajectories

PRACTICE	PRIMARY QUESTION	CHARACTERISTIC EVIDENCE
Diagnostics	Which state or mechanism best explains the evidence?	Correlated observations, competing hypotheses, residuals, causal tests, context
Debugging	Where and how does behavior depart from intent, and what changes under intervention?	Breakpoints, substitutions, controlled perturbations, replays, targeted tests
Observability	What hidden behavior can be inferred from designed outputs and preserved context?	Logs, traces, metrics, images, signals, provenance, identifiers, models
Prognosis	How may condition evolve, and when may intervention become necessary?	Degradation models, outcomes, uncertainty bounds, remaining-life estimates
Forensics / diagnostic review	What happened, in what sequence, and what was missed or misunderstood?	Preserved artifacts, timelines, chain of custody, autopsy or teardown, postmortems

SEVEN RECURRING CROSS-STRAND TRANSITIONS

- Sensory signs and local observations → quantified variables, structured records, and calibrated instruments
- One-time tests and snapshots → longitudinal histories, trajectories, and condition-aware baselines
- External examination → embedded, digital, remote, and continuously queryable observation
- Detection of anomaly → localization, mechanism, consequence assessment, and prognosis
- Individual cases and components → systems, populations, fleets, infrastructures, and organizations
- Unaided expert interpretation → AI-assisted ranking, explanation, and action under explicit governance
- Tool-specific artifacts → interoperable evidence with stable identity, time, provenance, and semantics (Ch. 31)

Contents

Seven parts · 33 chapters · three appendices · four worked cross-strand cases · one synthesis figure

Table 3. Article contents and opening pages.

PART I

Framing the five strands

1. Why five histories?

Let us begin with the diagnostic object. A person, a processor, a running program, a learned model, and a bridge do not supply evidence in the same way. A single history would have to move continually among different meanings of fault, observation, and intervention. We therefore keep five histories and compare their analysis methods only after considering the evidence of each [1–5].

Consider where diagnostic authority comes from in each case. A patient's account, examination, tests, and subsequent course contribute different evidence. For hardware, a design specification, firmware record, electrical measurement, and returned part may describe different levels of the same failure. Software supplies exact code and configuration, together with records of a particular execution. A learned model also needs data and evaluation. A physical asset needs a history of material, loading, environment, and maintenance. We have to retain these distinctions when comparing the conclusions.

An additional instrument gives us another observation path to examine. For example, a CT image depends on reconstruction, an error syndrome on decoding and physical mapping, and a memory dump on capture and matching symbols. A model score depends on its data and evaluation procedure; an ultrasonic indication depends on technique and geometry. In each case, we need to understand how the instrument produced the evidence before using that evidence to explain the condition.

2. Five diagnostic objects

Table 4. The object of diagnosis changes the meaning of evidence and intervention.

STRAND	DIAGNOSTIC OBJECT	HIDDEN STATE OR FAILURE OF INTEREST
Medical	A person within a biological, social, and care context	Physiology, pathology, disease extent, risk, response, and missed or delayed care
Hardware	A physical device or platform whose state is also encoded logically	Material defect, electrical margin loss, erroneous state, containment failure, and degradation
Software	A formal artifact executing in a changing technical and organizational environment	Incorrect state, unintended control flow, resource interaction, distributed causal sequence, and operational failure
ML and AI	A data-model-pipeline-service-human system that learns, predicts, generates, or acts	Miscalibration, leakage, shortcut learning, drift, unfairness, hallucination, unsafe trajectory, and governance failure
Technical	A physical structure, machine, plant, process, or cultural object	Damage, integrity loss, degradation mechanism, functional departure, and future condition

The objects sometimes overlap. We use the hardware strand mainly for electronic and computing devices, including the passage from a physical fault to an encoded error. The technical strand also includes machinery, structures, process equipment, infrastructure, and cultural objects. Computing enters that history when it acquires or analyzes physical evidence. This division gives each history a focus. An investigation may still need both [2, 5].

Consider an apparent model regression. The learned model may be poorly calibrated for a new population even though its program executes as specified. Alternatively, the change may originate in serving code, data transformations, retrieval, or an accelerator. We therefore begin at the level where the failure was observed and follow the evidence before assigning a mechanism to another level [1, 2, 4].

3. Shared vocabulary and limits of analogy

For our purposes, diagnosis includes obtaining signs, considering explanations, looking for evidence that separates them, and checking what happens after action. This organization lets us compare investigations across the five histories. The comparison still needs qualification. A clinical reference standard and a known-good computer answer different questions, and treatment carries obligations that component replacement does not. We use the analogy for the question it helps us formulate.

In software, debugging can include direct inspection and control of execution. A hardware investigation may vary load or substitute a unit; technical work may examine a failed part. We use debugging in medicine for the examination and correction of diagnostic reasoning or workflow. A patient can report experience and make choices, and an investigation can cause harm. Resetting, copying, and replaying therefore have only limited analogues in clinical work [3].

Observability relates available outputs to the state or behavior we want to infer. Control theory gives this relation a precise meaning for a specified dynamic model. In software, we ask whether production artifacts can answer questions about execution. ML and AI extend these questions to training, evaluation, and agent trajectories. Medical and physical observations depend on specimens, sensors, acquisition procedures, and context. We use the formal control-theoretic criterion when its model and assumptions apply [1, 3–6].

Consider two faults that produce the same available outputs. The system may expose many measurements and still leave these faults indistinguishable. We use diagnostic equivalence here for this inability to distinguish candidates under the current observations and test conditions. Another copy of the same evidence cannot resolve it. We need an additional stimulus, another measurement path, or a different operating condition that makes the alternatives predict different results. If no such test is available, the diagnosis retains the candidate set. This gives diagnosability a distinct role beside observability [2, 5, 7].

We often begin with an abductive step: this observation could have arisen in several ways. Further evidence may remove an explanation, support another, or reveal that our initial alternatives were incomplete. The resulting diagnosis remains conditional on what we could observe and test.

PART II

The strands in their own terms

4. Medical diagnostics

Let us start with the medical strand. A measurement gives us an observation about a person at a particular time. The history develops from case narratives and examination through laboratory medicine, imaging, molecular testing, continuous monitoring, surveillance, and computational support. With each method, we still need to ask what the result means for this person and the next care decision [3].

Different methods expose different aspects of the body. Imaging shows structure, microscopy shows cells and organisms, chemistry measures substances, and waveforms record physiological activity. Genomic tests and repeated monitoring add other views. Each view has an acquisition history. A specimen label, reconstruction method, or reference population can change the interpretation, so we keep the patient's narrative and the measurement procedure attached to the result.

A test may measure its target correctly and still leave the clinical question open. Analytical validity concerns the measurement; clinical validity concerns its relation to a condition; clinical utility concerns the decision or outcome that follows [8, 9]. Sensitivity and specificity describe performance under stated conditions, while predictive value also depends on pre-test probability [10]. Communication, consent, and follow-up belong to the diagnostic process because the result has to be acted on and reconsidered when it disagrees with the person's course.

Consider a biomarker that is associated with disease and measured reproducibly. Its use may still lead to unnecessary procedures if the decision threshold is unsuitable. A less precise test may be useful if it changes an urgent decision. The test's value depends on the clinical question, the alternatives, and what happens after the result is reported.

Screening adds another distinction. Earlier detection alone does not establish benefit: lead-time effects, preferential detection of slower disease, and overdiagnosis can change the apparent outcome. A false positive reports a target condition that is absent. Overdiagnosis identifies a real condition that would not have caused harm during the person's lifetime. We therefore follow the result through additional procedures, treatment, and outcomes when examining clinical utility [3, 11].

5. Hardware diagnostics

In the hardware strand, we follow physical condition through measurements, checking mechanisms, encoded errors, and service actions. Inspection, electrical bridges, meters, oscilloscopes, and maintenance records were joined by checking circuits, error-correcting codes, test equipment, scan chains, and built-in tests. Firmware records, management controllers, and fleet telemetry provide further views. Physical failure analysis can then examine a localized site in the laboratory [2].

We distinguish the fault, the error it produces, and the failure that may become visible outside the component. For example, an ECC correction tells us that an error was detected and corrected. It leaves the reason for the changed bit to be investigated. A machine check identifies a reporting location under a particular decoder. Replacing a unit may restore service while the original mechanism remains unknown. Detection, isolation, correction, containment, recovery, and prognosis each require their own evidence [12].

Sometimes one hardware event appears as an electrical waveform, a logical transaction, a firmware record, an operating-system report, and management telemetry. Later examination may add a microscopic feature. We join these representations through identity and time. Retry, reset, and replacement can remove the first-failure state, so the order of collection matters. Where further examination is justified, the returned part provides a route from the field symptom to a physical mechanism and a possible design or manufacturing change.

Consider increasing correctable memory errors. The counter indicates that a checking mechanism detected and corrected events. To investigate a recurring cause, we may need syndromes, physical address mapping, module identity, temperature and workload, firmware interpretation, replacement outcome, and examination of the returned part. Silent data corruption presents the related limit: a wrong result can pass through a system without an error signal, so independent computation, end-to-end checks, and outcome comparison may be needed to find it [2].

Physical failure analysis gives us another sequence of observations. X-ray inspection can show buried structure. Emission microscopy and electrical localization can narrow a suspect region before SEM or focused-ion-beam work examines it more closely. Preparation may alter or destroy evidence. We therefore record the original condition and the sequence of examinations, including what becomes unavailable after each step [2].

6. Software diagnostics

Software diagnostics develops around artifacts of execution. Checking routines, selected output, memory dumps, symbolic debugging, static and dynamic analysis, logs, profiles, metrics, and distributed traces

preserve different aspects of behavior. Incident records add the analysis and the actions taken. We can follow the history by asking which problem became answerable from each new artifact [1].

Consider a corrupted heap in a memory dump. We can inspect the damaged state, but the write that produced it may require an earlier trace or a replay. Source code gives us the program; configuration, data, scheduling, dependencies, hardware, and users determine the particular execution. A dump preserves selected state, a trace selected events, a metric an aggregate, and a profile samples. The diagnostic question determines which views we need to combine.

Before adding telemetry, we examine whether its records can be joined and interpreted. Identifiers, matching binaries and symbols, propagated context, time, collection health, and access to less transformed artifacts determine what we can reconstruct [13, 14]. Collection and intervention may also change execution. A debugger can alter scheduling, logging can add load, and a restart can remove the first-failure state. We record service restoration separately from the evidence supporting its causal explanation.

Suppose a distributed trace has a missing span. Perhaps the downstream call never happened. Perhaps context propagation failed, sampling excluded the span, or collection lost it. We can compare request logs and metrics, examine propagation and drop counters, and make a controlled request. The observation path becomes part of the case before we use the missing span to infer missing activity.

7. ML and AI diagnostics

The ML and AI strand includes residual analysis, statistical quality control, cybernetics, symbolic explanation, validation, benchmarks, learning curves, calibration, interpretation, and robustness tests. Data and pipeline checks connect these methods to production monitoring. Foundation models add open-ended outputs, retrieval, automated evaluation, and agents with tools. We now examine datasets, checkpoints, gradients, predictions, explanations, feedback, and recorded trajectories as diagnostic artifacts [4, 15, 16].

A model can complete its computation and fail its task. Leakage may inflate a benchmark, a learned shortcut may disappear in another population, or confidence may become poorly calibrated. An aggregate result may hide a subgroup with many errors. An agent may produce a plausible answer after calling the wrong tool or causing an unsafe side effect. These cases require us to examine the model together with its data, pipeline, serving system, users, and affected people.

The instrument used to explain a model is another object we can test. A feature attribution, saliency map, drift detector, or model-based judge has assumptions and failure modes. We can compare it with independent outcomes, apply sanity checks, and examine whether an intervention changes the predicted

behavior. For an agent, we also retain instructions, retrieval, available memory, tool calls and responses, retries, and side effects. These records let us investigate steps that the final answer may omit.

Some outcomes arrive after the initial diagnosis. A model label, clinical outcome, returned component, or structural failure may become available only later. A dashboard prepared too early can overrepresent cases that resolve quickly. We retain the pending cases and revisit the same cohort when later evidence arrives, keeping changes to its membership and denominator explicit [3–5].

Suppose overall classifier accuracy appears stable while confidence becomes too high after a population change. We can compare the production inputs with the holdout population, examine calibration against later outcomes, and separate affected subgroups. Delayed labels complicate this comparison: recent predictions may have no verified outcome yet, and the apparent trend may change when those labels arrive. A successful software execution tells us little about these behavioral and decision errors [4, 16].

8. Technical diagnostics

The technical strand concerns the performance and integrity of physical structures and systems. We encounter inspection, mechanics, metrology, fatigue and fracture analysis, nondestructive evaluation, condition monitoring, structural-health monitoring, failure analysis, and prognostics. The objects range from machines and pressure equipment to aircraft, bridges, manufacturing processes, and cultural objects. Their material condition and the consequence of possible damage guide the next examination [5, 17].

Consider an ultrasonic reflector. We still have to decide whether it represents a flaw, characterize that flaw, and assess its consequence. A thermal region or vibration peak also admits several explanations, including changes in load or resonance. Procedure qualification, probability of detection, access, geometry, calibration, and environment tell us what the examination could find and how well it could characterize the result [18].

An estimated flaw size acquires meaning from material, orientation, stress, environment, and future loading. Fitness-for-service analysis connects these observations to operation, repair, replacement, and reinspection [19]. The examination itself may alter the object. This is especially apparent in cultural heritage, where destructive confirmation can remove material of historical significance. We preserve the condition and treatment record and state what remains uncertain after noninvasive examination [5].

We can therefore follow an ultrasonic indication through several claims. A qualified technique supports a size estimate with uncertainty. Loading, geometry, material properties, and the intended operating period then support an assessment. Continued use, restriction, repair, replacement, or another inspection follows from that assessment and the applicable decision rules. Each step adds information to the original indication.

PART III

The shared historical arc

9. From signs to quantified evidence

The histories begin with signs recognized through experience: a symptom, a sound, an unusual output, or a visible change. Instruments make some of these observations measurable and repeatable. We can compare a temperature, voltage, execution time, residual, or displacement with a stated reference. The measured quantity still depends on what was selected for observation and how it was obtained [1–5].

The number depends on how it was obtained. Blood pressure varies with cuff and posture, a waveform with probe and bandwidth, and latency with clocks and sampling. Model accuracy depends on the evaluation split; an NDE indication on geometry and technique. We retain units, method, calibration, operating conditions, and uncertainty before combining measurements into a trend or a model.

The original record may answer a question that the summary cannot. A narrative preserves context, a waveform preserves shape, and a dump preserves relationships among selected values. We keep a route from a derived feature or score back to the available artifact, with the transformations used to produce it.

10. From snapshots to trajectories

A single observation gives us one view of a changing object. Repeated observations add a trajectory: a patient's course, a component's error history, a service's latency, a model's calibration, or an asset's degradation. We can then ask when the departure began and which change preceded it [1–5].

A trajectory also needs a comparable baseline. The instrument, method, population, workload, or operating regime may have changed between observations. Consider a step in a trend immediately after a sensor replacement or software release. We first determine which part of the step belongs to the object and which part belongs to the observation method.

Continuous collection still leaves gaps. Sampling, retention, outages, and access determine which intervals survive. Repeated normal values can also come from a stuck instrument. We record coverage and changes to the collection policy so that a later analyst can distinguish an observed quiet period from an unobserved one.

11. From external examination to embedded and continuously queryable observation

Observation increasingly takes place inside the object or its operating environment. Hardware checks, software instrumentation, implanted or wearable sensors, embedded monitoring, and remote acquisition

provide evidence during use. Models and digital twins add derived views. Their value depends on the path from the original observation to the displayed interpretation [1–5].

The embedded observer can fail as well. A sensor may drift, a firmware decoder may misinterpret a record, or a sampler and collector may omit events. DICONDE gives us a documented example of shared imaging infrastructure: it adapts DICOM concepts to nondestructive evaluation while retaining industrial objects and technique information. Trace context and telemetry conventions help join software observations in a similar infrastructural role. In each case, the implementation and collection path still need checking [13, 14, 20, 21].

12. From detection to localization, mechanism, consequence, and prognosis

An alert establishes an initial claim. We may then need to locate the problem, distinguish its mechanism, assess the consequence, and estimate future condition. A medical screen, machine check, service alert, drift detector, or NDE indication has a particular scope. Further claims require evidence beyond that scope [12, 17, 18].

We can write the sequence as detection → localization → mechanism → consequence → prognosis. Sometimes an investigation stops earlier because the available evidence already supports the necessary action. We record that endpoint. Recovery tells us what changed after an action; a causal explanation needs its own support. Prognosis also needs assumptions about future treatment, load, environment, or use.

13. From local artifacts to systems and populations

The diagnostic object can expand from one case or component to a pathway, fleet, service, data pipeline, infrastructure, or population. This lets us relate a local sign to a remote dependency or shared exposure. Aggregation can reveal recurrence, although it may also remove the detail needed to explain an individual failure [1–5].

For example, a group of crashes may point to a release, and a group of hardware errors may point to a manufacturing lot. Surveillance can reveal a cluster; model monitoring can reveal a subgroup regression. We use the distribution to choose the next cases to examine. We also ask which cases entered the record and which were followed to an outcome.

The population finding should lead us back to an inspectable artifact: a representative dump, device, specimen, request trace, evaluation record, or failed part. We can then test whether the local mechanism explains the distribution that drew our attention. An aggregate health score is useful when this return path remains available.

14. From human craft to AI-assisted diagnosis

Early automation encoded checks with stated expectations, such as limits, parity, assertions, control charts, rules, and test vectors. Statistical methods added ways to rank anomalies. AI can classify images, retrieve cases, summarize records, suggest explanations, and select investigative actions. We evaluate these operations against the evidence and task for which they are used [1–5].

Suppose an AI explanation is based on a mislabeled specimen or an incomplete trace. Its conclusion inherits the defect in the evidence. We therefore retain the original artifacts and ask what the proposed explanation distinguishes from the alternatives. Fluency alone supplies no additional observation.

We give assistance a defined scope. Its source artifacts, generated inferences, alternatives, uncertainty, and proposed tests should be available for review. The authority to change a clinical pathway, production service, model, or asset depends on consequence and reversibility. A review step is useful only when the reviewer has time, information, and the practical ability to disagree.

PART IV

A unified evidence architecture

15. The diagnostic continuum

Table 5. A general diagnostic continuum that can be instantiated differently in each strand.

STAGE	CORE QUESTION	TYPICAL OUTPUT
Observe	What sign, state, event, or account is available?	Narrative, image, waveform, event, sample, counter, trace, physical indication
Measure / preserve	How was it produced, and can it be retained faithfully?	Calibrated value, raw artifact, specimen, dump, immutable record, acquisition metadata
Contextualize	Which identity, configuration, environment, operating state, and time apply?	Provenance bundle, topology, patient or asset history, version, units, status
Compare	What qualified baseline, reference, model, peer, or expectation is relevant?	Residual, delta, threshold crossing, anomaly, discordance, cohort difference
Infer	Which states or mechanisms could explain the observations?	Ranked hypotheses, differential diagnosis, fault set, causal graph, uncertainty
Discriminate	What test or intervention would separate the alternatives?	Repeat or orthogonal test, perturbation, substitution, ablation, replay, controlled load
Act	What response is justified by evidence, consequence, and authority?	Treatment, repair, rollback, containment, inspection interval, abstention, escalation
Verify and learn	Did the action change the predicted outcome, and what should be retained?	Outcome evidence, regression test, postmortem, updated model, standard, or training

Table 5 gives us the working sequence: Observe; Measure / preserve; Contextualize; Compare; Infer; Discriminate; Act; Verify and learn. Each stage answers a different question. We may repeat a comparison after finding another artifact, return to acquisition when two explanations remain indistinguishable, or revise an inference after an intervention. The sequence helps us locate the current claim and the evidence still required.

Sometimes action comes first. An urgent treatment, rollback, or containment step may precede a complete explanation. That action changes the object and may remove evidence, while its outcome provides another observation. We record what was known at the time and return to the unresolved questions afterward.

Observability concerns the evidence and context we can obtain. Diagnostics compares explanations, while debugging uses inspection and controlled changes to investigate departures from intent. Monitoring repeats selected observations, and prognosis concerns possible future states. Governance determines who may act. Keeping these meanings explicit helps us follow a measurement through an inference to an authorized decision.

16. Identity, provenance, and time

Let us first attach the evidence to the right object. A clinical result needs patient and specimen identity. A hardware syndrome needs slot, address mapping, and firmware version. A stack needs its build and symbols. An AI result needs model, prompt, retrieval, policy, and evaluator versions. A physical indication needs asset, location, procedure, and operating condition. Detailed analysis of the wrong object can still produce a plausible explanation.

We use provenance to record how an observation became the result in front of us. Acquisition, filtering, transformation, reconstruction, annotation, and aggregation can all affect its meaning. Where possible, we retain the original specimen or raw artifact and link each derived image, feature, stack, health score, or explanation to the operations that produced it.

Time also has several meanings. A wall clock relates an event to people and external changes; a monotonic clock measures duration. Message relationships may establish causal order when wall clocks disagree. Cycles, exposures, age, and treatment stages can describe physical or biological progress more directly. We preserve the temporal relation needed to examine the proposed mechanism.

A snapshot also has a capture interval. In an ideal memory dump, all selected values belong to one instant. In a real acquisition, different regions may be copied at different times while parts of the system continue changing. The combined artifact can then contain a state that never existed as a whole. A scan or a set of distributed polls has a related temporal qualification. We retain acquisition start and end, ordering, pause conditions, and relevant configuration changes before treating the artifact as simultaneous evidence [1, 2, 5, 22, 23].

The reason for capture belongs with this metadata. In the 4WS approach described in “Five Golden Rules,” we ask what was observed, when and where it occurred, why the artifact was requested, and what supporting information is available [22]. These questions transfer naturally to an evidence request. We record the question the collection was meant to answer, the method used, and any omitted region or interval. Otherwise, a later analyst may receive a valid artifact that cannot answer the actual problem.

17. Baselines, thresholds, and reference standards

We call a value abnormal in relation to a reference. This may be the person's earlier result, a peer population, a known-good device, a validation set, a design model, a control chart, or an acceptance rule. Selecting that reference is an analysis operation. It determines which departures we can recognize and which remain inside our definition of normal.

Consider the same threshold under two operating conditions. A physiological value may have a different meaning for another person; a hardware error rate may change with workload; latency may change with

traffic. Model calibration depends on population and time, and structural response on load and environment. When a threshold triggers an action, we also examine false calls, missed cases, delay, and intervention cost.

Calibration needs a more precise meaning here. Metrological traceability connects a measurement result to a reference through a documented calibration chain, with uncertainty contributed along the chain. The serial number and custody history establish provenance but do not, by themselves, establish this relation. Traceability also leaves open whether the uncertainty is small enough for the intended decision. Near an acceptance threshold, we therefore retain the uncertainty and the decision rule with the reported value [2, 5, 24].

We also examine how the reference result became available. Suppose only positive screens or alarms receive independent verification. The unverified negative cases may contain misses, so the observed cases cannot establish performance for the whole population. This is a form of verification bias. A reference standard can itself be imperfect. In ML, related examples crossing training and holdout boundaries create another problem with the comparison. We preserve how cases were selected, which received verification, and which outcomes remain unknown [3, 4, 25].

A reference can become stale. We retain its version and review it when instruments, population, material, configuration, or workflow change. Automatic adaptation can absorb gradual degradation into the baseline unless we preserve its history. An unchanged baseline can become equally misleading when the original conditions no longer apply.

18. Probability, loss, and the value of a test

A result changes our judgment in relation to prior knowledge. In medicine, a likelihood ratio updates pre-test odds to post-test odds. Sensitivity and specificity describe a test under defined target conditions, while predictive values depend on prevalence. A hardware syndrome also means something different in a suspect lot and in a general fleet. We have to preserve the population and setting behind the comparison [8, 10].

Other strands use related methods with different meanings. A probability-of-detection study describes inspection performance for specified flaws and conditions. A fitness-for-service assessment relates estimated damage to loading and future use. In ML, calibration compares stated probabilities with observed frequencies. We can compare their roles while retaining the target, population, and measurement path of each method [16, 18, 19, 26].

We select a test for the decision it may change. Its expected value has to justify its cost, including possible injury, delay, privacy loss, operator time, computation, and lost evidence. Missing a dangerous condition may be more costly than a false alarm. Destructive examination needs different grounds from a reversible

rollback. When the available evidence cannot support an action, an agent may need to abstain and escalate.

Probability and decision models help express these choices. They still depend on correctly identified specimens, understood trace coverage, and a model appropriate to the sequence being examined. Authorization remains a separate requirement. Later, we use analysis patterns to organize these evidence operations around the formal methods available in each domain.

19. Causal isolation, intervention, and verification

To separate explanations, we can obtain another observation or change a bounded factor. An assay can be repeated with another method; a hardware unit can be examined under a different load; a software execution can be replayed. ML may use an ablation, and technical work a targeted inspection or controlled load. We choose the operation from what the alternatives predict and what the object permits.

Before intervening, we state the hypothesis and the result that would count against it. We preserve perishable evidence and consider how the intervention changes the case. A debugger can alter scheduling, a probe can load a circuit, a biopsy changes the specimen, and repeated loading can extend damage. If every possible result is accepted as support afterward, the test has separated nothing.

There may also be more than one fault. One fault can mask another, or two apparently separate failures can share a power supply, environment, or maintenance action. Replacing a component may remove one symptom while leaving the shared cause. We compare candidate combinations when the observations require them and retain common-cause alternatives. A successful intervention shows that the changed factor mattered under the tested conditions; further evidence is needed to establish a unique explanation [2, 5, 27].

After the action, we examine the predicted change and look for new harm. Regression tests, clinical follow-up, returned-part analysis, and repeated monitoring connect the intervention to a more durable conclusion. Restoring service may be urgent; establishing why the restoration worked can require another investigation.

20. Uncertainty, consequence, and action

We attach uncertainty to the claim it qualifies. It may concern acquisition, coverage, interpretation, the model, the reference standard, or future conditions. Similar estimated probabilities can lead to different actions for a possible crack, a missed clinical condition, silent computation error, or a model failure concentrated in one group. The consequence matters as well as the estimate.

We also identify who bears that consequence: the patient, users of a service, an operator, people affected by an AI system, or those who depend on a physical asset. Technical evidence alone cannot choose their decision thresholds. Authority and responsibility remain with the relevant people and institutions.

An inconclusive result can be useful when it states its limit. We may defer a classification, restrict operation, request another inspection, or refer a case for review. The record should identify the missing evidence and the next step that could reduce uncertainty. Abstention is then an explicit diagnostic outcome.

PART V

Convergence without collapse

21. What the strands borrow from one another

The strands borrow methods and instruments from one another. Computing uses diagnostic terminology from medicine and engineering; reliability methods, fault trees, and control ideas appear in several technical domains. Statistical quality control informs laboratories, manufacturing, and model monitoring. We check the history of a proposed transfer because similar procedures can also develop independently [1–5].

DICOM and DICONDE provide a specific example. Medical imaging standards bind an image to identity and acquisition information. ASTM E2339 adapts this organization to nondestructive evaluation and defines information objects for industrial inspection. The infrastructure has a documented relationship. The interpretation still belongs to the particular patient, weld, or other object being examined [20, 21].

Software telemetry supplies evidence for ML and AI through logs, metrics, traces, propagated context, profiling, and incident records. ML supplies methods for image classification, fault ranking, and condition monitoring in the other strands. We therefore encounter diagnosis in both directions. A training failure may originate in an accelerator or pipeline, and the model used to diagnose it may itself need investigation [1, 2, 4, 13, 14].

22. Boundary zones and overlapping objects

Table 6. Boundary zones where no single strand is sufficient.

BOUNDARY	WHY IT IS SHARED	EVIDENCE THAT MUST BE JOINED
Medical device failure	Patient outcome depends on hardware, software, calibration, workflow, and interpretation	Device logs, waveforms, configuration, clinical record, maintenance, human factors, outcome

BOUNDARY	WHY IT IS SHARED	EVIDENCE THAT MUST BE JOINED
AI infrastructure incident	Model behavior and job progress depend on accelerators, networks, storage, orchestration, and data	GPU/HBM telemetry, fabric traces, system logs, checkpoints, loss curves, topology, workload phase
Autonomous vehicle or robot	Physical sensing, embedded hardware, control software, learned perception, and environment interact	Sensor data, hardware errors, software traces, model outputs, control actions, scene and maintenance history
Digital twin decision	A maintained model mediates between a physical asset and an intervention	Asset identity, sensor quality, model version, calibration, uncertainty, predicted and observed consequence
Hospital or industrial cyber-physical system	Safety and service depend on physical process, devices, networks, software, and organization	Process variables, alarms, device and network evidence, procedures, changes, staffing, response timeline

At a boundary, we state the claim at each layer. An exception may follow a hardware error, a model regression may arise in feature processing, and a clinical mistake may begin with specimen identity or device configuration. We widen the investigation when another layer could change the action or prevention. When further expansion is unlikely to change either, we can stop and record the unresolved levels.

The evidence may also cross organizational boundaries. A hospital, vendor, service team, and clinical unit may each hold part of a case. In an AI infrastructure incident, the pieces may belong to model, platform, hardware, and data teams. Their identifiers, clocks, definitions, and access rights determine which records can be joined. An unavailable join is itself a finding about the observation path.

23. Irreducible differences

The limits of analogy begin with the person. A patient reports experience, makes choices, and can be harmed by an investigation. Communication, consent, and care shape the diagnostic decision. Some hardware or technical objects can be disassembled or sacrificed for analysis; that experimental freedom cannot be carried into medicine. Imperfect reference standards and unequal access add further limits [3].

Software often supplies exact code and may permit copying, instrumentation, replay, or minimization. The relevant production state can still disappear on exit or redeployment, and distributed event order may remain partial. A physical component can retain damage, although access may be difficult and examination destructive. In both cases, we ask which evidence survives and which operations would change it [1, 2, 5].

For ML and AI, learned parameters interact with data, population, sampling, retrieval, feedback, and environment. The same code can produce different outputs and external effects. An improvement for one group may be a regression for another. We specify the task, population, time, metrics, and permitted actions, then continue evaluation after deployment [4].

Technical diagnosis connects detectability to structural consequence. Hardware adds logical encoding and error correction to physical mechanisms. Software permits changes to code and configuration without repairing matter. Across these differences, we can still transfer a question such as which comparison would distinguish two explanations. The actual test and authority to perform it remain specific to the domain.

PART VI

Toward a general diagnostic discipline

24. Shared principles

Start with the diagnostic object and decision. Are we screening, detecting, localizing, explaining, containing, treating, repairing, or estimating future condition? The same artifact may be useful for one question and insufficient for another.

Preserve evidence before it changes. Physiology, memory, buffers, software deployments, model versions, machine state, surfaces, and scenes may not be recoverable later.

Preserve identity, provenance, and time. A patient, specimen, component, build, model, asset, method, and configuration must remain attached to the observation that concerns it.

State separately what was observed, what mechanism was inferred, and what was done. A counter, image, or score can support an explanation without containing it.

Compare partly independent evidence paths. Agreement is useful when the methods have different failure modes. Disagreement may reveal a fault in the object, the observer, or the assumptions used to join them.

Record missing evidence. A dropped span, dead sensor, absent label, unreadable scan, or stale symbol must not be interpreted as a normal result.

Detect a problem as near to its origin as the system allows. Early checks can reduce propagation and preserve a smaller state for analysis, but the check itself needs validation.

Limit consequence while inquiry continues. Containment, restricted operation, feature flags, safety-netting, abstention, and staged release can be appropriate before a mechanism is established. We should preserve evidence of the action and its effects.

Test the means of diagnosis. Calibration, controls, known fault injection, telemetry loss tests, evaluator sanity checks, and review of human oversight show where the observation path fails.

Make the diagnostic budget explicit. Sampling, retention, inspection interval, test cost, access, privacy, and staff time determine what can be known later. Record material exclusions and the resulting coverage limits.

Return the findings to design and practice. A diagnosis may change instrumentation, manufacturing, treatment pathways, maintenance, software tests, datasets, or organizational procedure. Closing a case without checking recurrence loses that opportunity.

We can apply these principles before an incident. Let us consider the diagnostic questions a future failure might raise and arrange access to the evidence they require. Hardware may need test points and retained first-error records; software needs suitable artifacts, matching symbols, and usable identifiers. An asset may need inspection access and a baseline before service. This is design for later diagnosis. Its choices determine which alternatives a future analyst will be able to distinguish [1, 2, 5, 22].

25. A cross-strand evidence taxonomy

Table 7. Evidence categories that recur across the five histories.

CATEGORY	WHAT IT ESTABLISHES	EXAMPLES ACROSS STRANDS
Identity and context	What object, version, environment, and operating state the claim concerns	Patient/specimen; serial/slot; build/configuration; model/prompt; asset/location/load
Raw observation	The least-transformed available record	Narrative, waveform, image, dump, packet, sensor sample, radiograph, scan, fracture surface
Derived feature	A measured or computed property extracted from raw evidence	Interval, spectrum, stack, span duration, attribution, flaw size, health indicator
Expectation / model	What should happen under stated assumptions	Physiology, circuit or component model, specification, statistical model, finite-element or degradation model
Temporal evidence	How state and intervention changed through time	Symptom course, event trace, error trend, learning curve, agent trajectory, crack-growth history
Comparative evidence	How the case differs from baseline, peers, controls, or cohorts	Reference interval, known-good unit, healthy service, holdout set, fleet or structural peer
Intervention evidence	Whether changing a factor changes the predicted outcome	Treatment response, substitution, replay, rollback, ablation, controlled load, excavation
Outcome and consequence	What the state did to function, safety, cost, or mission	Clinical outcome, service impact, containment, model harm, structural capacity, recurrence
Observation-path health	Whether the evidence system itself was functioning	Quality controls, calibration, loss counters, sensor self-test, schema/version checks, evaluator validation
Governance and accountability	Who authorized, interpreted, acted, and reviewed	Consent, audit trail, change record, incident role, approval, inspection qualification, model card
Evidence integrity and adversarial resistance	Whether evidence is authentic, complete, and resistant to deliberate manipulation	Specimen chain of custody; signed logs; protected calibration records; poisoning checks; tamper-evident inspection data

CATEGORY	WHAT IT ESTABLISHES	EXAMPLES ACROSS STRANDS
Resource and capture budget	What coverage, fidelity, retention, and latency the investigation can afford	Test cost and access; telemetry sampling; compute and storage; inspection interval; escalation criteria
Delayed and unresolved outcomes	Which later evidence is pending, censored, disputed, or never observed	Clinical follow-up; returned parts; delayed ML labels; warranty and inspection outcomes

Table 7 groups evidence by what it preserves. A patient's narrative, an oscilloscope waveform, a packet capture, and a fracture surface offer different routes into a case. Some retain sequence, some expose a physical mechanism, and some record the authority for a decision. We also distinguish availability from integrity: an observation may be absent because collection failed, or present but altered through deliberate manipulation.

Consider three dashboards fed by the same sensor. Agreement among them gives us three presentations of one observation path. A controlled intervention may provide more discrimination, although a striking result can also be incidental. We ask how each artifact was produced, which alternatives it excludes, and whether it explains the consequence under investigation.

A negative finding needs a coverage statement. The absence of an error report, organism in a culture, trace span, model alert, or NDE indication excludes a mechanism only to the extent that the method could have revealed it. Was the instrument operating? Was the relevant region or interval examined? Would the proposed mechanism have produced a detectable sign? We retain these questions beside the finding [1–5].

26. Analysis patterns as a candidate unifying layer

We can now consider analysis patterns as a candidate language for work across the five strands. Pattern-Oriented Diagnostics and Pattern-Oriented Observability have a developed software history. We take selected operations from that history and ask whether they remain useful when their evidence and procedure are qualified for another domain. Appendix C supplies four worked demonstrations. A general catalogue would require further known uses and validation [28–32].

Let us consider the operation before the mechanism. For a disease, cracked weld, marginal memory cell, race condition, or drifting model, we may ask which baseline applies, which artifact belongs to the object, or which observation could distinguish two explanations. These questions have a recurrent form. An analysis pattern can describe the form and the conditions needed to apply it.

We call recurrent diagnostic situations problem patterns and the operations used to examine their evidence analysis patterns. An implementation pattern realizes an operation in a tool or workflow; a presentation pattern makes its result inspectable. The terminology needs context. Martin Fowler uses analysis patterns for reusable conceptual models of business domains. Here we use it for diagnostic analysis operations [33–35].

For example, an Adjoint Thread of Activity can be obtained by selecting messages according to a criterion other than the emitting thread. Adjoint Space relates a trace to another evidence space, such as memory, which may contain information missing from the messages [36]. These examples show the kind of operation we want a pattern description to preserve: what is selected, how the records are related, and which additional question the resulting view can answer.

At DumpAnalysis.org, Pattern-Oriented Diagnostics and Observability distinguishes problem patterns from analysis patterns. The latter give us operations for selecting, transforming, comparing, and interpreting diagnostic artifacts. The four-part Pattern-Oriented Observability sequence examines observability within the same pattern-oriented and systemic diagnostic programme [28–32].

- **Part 1** - memory as the base slab beneath metrics, logs, and traces, with memory state and emitted messages related through analysis patterns and a memory-trace duality [29].
- **Part 2** - a semiotics of memory, traces, and logs in which icons, indexes, Iconic Traces, and the Dia|gram language represent structure, behavior, observation, and interpretation [30].
- **Part 3** - observation spaces for memory, traces, logs, metrics, narratives, states, and data, with analysis patterns serving as ways to compare and navigate diagnostic spaces [31].
- **Part 4** - the Diagnostics–Observability Square, which distinguishes observability and diagnosability as properties from observation and diagnostics as processes, and names pattern-oriented observability as part of pattern-oriented and systemic diagnostics [32].

Table 8. Proposed cross-strand diagnostic-analysis-pattern families. The third column always follows this order: Medical; Hardware; Software; ML/AI; Technical. The labels are a synthesis rather than a claim that a formal catalogue already exists in every domain.

ANALYSIS-PATTERN FAMILY	REUSABLE OPERATION	PATTERN-ORIENTED PROJECTION ACROSS THE FIVE STRANDS
Baseline Comparison	Compare a case with a qualified prior state, peer, control, specification, or model.	Personal trajectory; known-good unit; healthy service cohort; holdout or reference distribution; qualified structural baseline.
Identity and Provenance Reconstruction	Rebuild the chain from diagnostic object through acquisition, transformation, and interpretation.	Patient and specimen; component, slot, and firmware; build and symbols; data, model, prompt, and evaluator; asset, location, procedure, and calibration.
Temporal Reconstruction	Order events, states, interventions, and gaps so that transition and mechanism can be examined.	Clinical course; first-failure record; execution or distributed trace; learning curve or agent trajectory; load and damage history.
Multimodal Correlation	Join evidence from partly independent modalities while preserving disagreement and missingness.	Examination, laboratory, and imaging; waveform, syndrome, and microscopy; dump, log, metric, trace, and profile; evaluation, attribution, and feedback; NDE, load, and fractography.
Differential Hypothesis Construction	Generate and rank competing states or mechanisms without turning a ranking into a verdict.	Differential diagnosis; candidate physical faults; software causal hypotheses; model, data, and pipeline failure classes; degradation mechanisms.

ANALYSIS-PATTERN FAMILY	REUSABLE OPERATION	PATTERN-ORIENTED PROJECTION ACROSS THE FIVE STRANDS
Causal Isolation by Intervention	Change one bounded factor and test whether the predicted observation changes.	Treatment response; component substitution or margining; replay, rollback, or minimized input; ablation or counterfactual test; controlled load, targeted inspection, or teardown.
Observation-Path Diagnosis	Test whether the sensor, specimen path, probe, telemetry pipeline, reconstruction, or evaluator is itself failing.	Quality controls; calibration and self-test; loss counters and symbol checks; judge and explanation sanity checks; sensor drift and inspection qualification.
Containment and Verification	Limit consequence, preserve evidence, and confirm that the intervention changes the expected outcome without unacceptable new harm.	Safety-netting or follow-up; fault containment and degraded mode; feature flag or rollback; abstention and staged release; restricted operation, repair, and reinspection.

An agent gives us a useful example of the relation between memory and traces. A memory view may preserve instructions, retrieved material, context, checkpoints, and tool state at one step. A trace records selected prompts, calls, replies, retries, handoffs, updates, and side effects. We use one view to question the other: how did this state arise, and what action followed from it? The records remain partial, and a generated rationale does not expose all the internal computation [4, 29–32].

Suppose a failed tool call is retried and a payment is duplicated, while the final answer reports one successful payment. Patterns for repeated activity, sequence discontinuity, missing state, or inconsistent messages give us starting points. We then need tool identifiers, retry policy, side-effect records, and the evidence available to the agent at each step. These artifacts determine whether the proposed mechanism fits the case.

For an analysis pattern to travel, we describe its intent, context, problem, required evidence, procedure, alternatives, observer effects, resulting context, verification, stopping conditions, and known uses. Known uses establish recurrence; stopping conditions say when another step is unlikely to change the decision enough to justify it. A clinical workflow, inspection procedure, debugger command sequence, telemetry query, or agent tool realizes the pattern only after qualification for that use.

The analysis also produces artifacts. Queries, selected records, transformations, rejected alternatives, and reasons for stopping can be preserved for later examination. In software, A.C.P. organizes work around artifacts, checklists, and patterns; Analysis Pattern Networks relate operations whose results become inputs to further analysis [1]. A cross-strand implementation also needs to preserve permissions, acquisition limits, and the authority behind each action.

We judge a transferred pattern by the evidence it helps us find or test. Reusing a name is only the beginning. We have to qualify the required observations, permitted intervention, authority, and verification, then compare the resulting investigation with established practice in that domain.

27. Comparison with other unifiers and what would refute the proposal

Several frameworks already organize parts of this work. Model-based diagnosis derives candidate faults from inconsistencies with a structural model. Fault detection and isolation examines residuals in dynamic systems. Bayesian decision theory relates beliefs to consequences, and control-theoretic observability asks whether outputs determine state under a specified model. FMEA and fault trees examine anticipated failures; semiotics examines signs and interpretation. We retain these methods where their assumptions apply and their formulation is more precise than a pattern description [6, 10, 17, 27, 30, 37–40].

Table 9. Candidate cross-strand unifiers: scope, requirements, strengths, and limits.

CANDIDATE	WHAT IT UNIFIES	REQUIRES	STRENGTH	LIMIT ACROSS FIVE STRANDS
Diagnostic analysis patterns	Operations performed on evidence	Pattern description, domain qualification, known uses	Transfers inquiry without requiring one ontology	Recurrence and benefit must be demonstrated; names can become relabelling
Model-based diagnosis	Candidate generation from inconsistency	Structural model, component modes, observations	Formal and discriminating where models exist	Weak where structure or failure modes cannot be specified credibly
Fault detection and isolation	Residual generation and fault isolation	Dynamic process or signal model, sensors, thresholds	Strong for machinery, vehicles, plants, and control systems	Limited for narrative, ethical, institutional, and open-ended diagnostic objects
Bayesian decision theory	Belief updating and action under loss	Probabilities or likelihoods, outcomes, loss function	Genuinely cross-domain and explicit about thresholds	Does not reconstruct identity, sequence, mechanism, or authority by itself
Control-theoretic observability	Inferability of state from outputs	Specified dynamic state-space model	Formal criterion with clear mathematical meaning	Not directly portable to changing socio-technical systems without a defensible model
FMEA and fault trees	Anticipated failure modes, effects, and causal combinations	System boundary, enumerated modes or top event	Widely transferred and useful before operation	Completeness is difficult; novel interactions and learned behaviour escape enumeration

CANDIDATE	WHAT IT UNIFIES	REQUIRES	STRENGTH	LIMIT ACROSS FIVE STRANDS
Semiotics	Relations among signs, objects, and interpretations	Representational vocabulary and interpretive rules	Explains mediation and meaning across artifacts	Provides less direct guidance for test selection, action, and verification

Our proposal concerns the organization of inquiry across heterogeneous evidence. When an appropriate structural, dynamic, probabilistic, or other formal model is available, an analysis procedure should use it. The pattern describes where the operation fits and what evidence it requires. Its name supplies neither additional evidence nor mathematical validity.

The proposal can fail. Independent practitioners may find that a mapping merely renames unrelated operations, that its evidence or stopping condition cannot be specified in a strand, or that it produces worse decisions than established practice. We would then narrow or reject that transfer. The four cases in Appendix C demonstrate possible mappings; the other families in Table 8 still need their own examination.

We can test a proposed transfer on the same cases used to assess an established domain procedure. Before the comparison, we specify the diagnostic question, permitted evidence, verification, and stopping conditions. We then record missed alternatives, unsupported conclusions, harmful actions, effort, and disagreements between analysts. A useful result would show where the pattern improves the inquiry and where it adds no value. This is a proposed evaluation method, and the worked cases alone do not supply its results.

28. AI as instrument, subject, and governor

AI can occupy three positions in a diagnostic case. It may be the instrument that classifies an image or ranks faults, the object whose data and behavior we investigate, or a participant that recommends or performs an action. We state the role before evaluating its claim. When a model diagnoses another system, its own observation path remains open to examination.

For an agent, we relate preserved context to instructions, retrieval, tool calls, observations, retries, and side effects. Which recorded step supports the explanation? Which earlier event is missing? Memory and trajectory artifacts help answer these questions while retaining partial order and incomplete coverage where the records require them [4, 29–32].

An AI intervention also changes later evidence. A new maintenance schedule, clinical priority, service configuration, or routing decision affects the outcomes on which the system may be evaluated. We

therefore retain records of authorization and action and use comparisons that account for this feedback. A recommendation and an executed command need distinct records.

A diagnostic copilot can maintain a case file, cite artifacts, repeat queries, retain alternatives, and propose bounded tests. We evaluate the resulting evidence and outcomes, including missed cases and harmful actions. The human review step needs examination too. Automation bias, fatigue, and production pressure can turn review into routine acceptance [41–44].

Where the workflow allows it, we can record an independent judgment before showing the recommendation. We can also audit accepted suggestions and examine overrides, escalations, and disagreement. These operations require appropriate training and staffing. If we rely on human review, we need evidence that the review occurred and could change the decision.

29. Reflexive diagnostics: diagnosing observability, diagnostics, and debugging

Sometimes the fault lies in the means of diagnosis. A broken identifier, missing telemetry, unsuitable baseline, misleading dashboard, invalid evaluator, or unsafe probe changes our view of the original object. We call the investigation of these failures reflexive diagnostics. It extends higher-order analysis of diagnostic analysis [45] and the observation-path questions of Pattern-Oriented Observability [29–32].

Pattern-Oriented Observability examines these relationships through observation paths and the Diagnostics–Observability Square [29–32]. Consider a debugger that changes timing, a drifting sensor, an invented source in an AI explanation, or an agent that modifies the state used to assess its action. The case now includes the diagnostic procedure and the evidence of its effects.

Each strand already examines parts of this path. Medicine checks specimens, assays, handoffs, and decision support. Hardware tests probes, reporting channels, and built-in test coverage. Software examines logging, propagation, profiling, and incident procedure. ML and AI evaluate explanations, judges, drift detectors, and agents. Technical work qualifies sensors and NDE procedures. We include the analyst because fatigue, automation bias, and incentives can affect interpretation [41, 42, 44, 46].

We also examine integrity. A specimen may be substituted, an inspection record falsified, a log altered, or an evaluation set poisoned. Apparently independent records may then share an adversarial source. Authentication, chain of custody, sequencing, controls, and independent checks help us investigate this possibility. Missing and overridden records remain part of the case [47, 48].

30. Pattern narratives and narratological unification

An investigation usually applies several analysis patterns. We recognize a departure, preserve artifacts, compare a baseline, reconstruct a sequence, test alternatives, and check an intervention. The record of

these operations forms a diagnostic narrative. It lets another analyst follow how we reached the conclusion and identify the evidence that could change it.

Higher-Order Pattern Narratives makes the analysis itself available for analysis. We can examine a sequence of pattern applications and find, for example, a filter that removed the relevant event or a jump from symptom to cause. Unified Computer Diagnostics: Incorporating Hardware Narratology extends reconstruction across hardware and software records [45, 49]. The resulting narrative may still contain missing intervals and unresolved relationships.

Consider an agent's final response alongside its trajectory. The response may omit retrieved material, a failed tool call, repeated attempts, a memory update, or an external side effect. A memory pattern examines selected state; a trace pattern examines selected transitions. We compose these results to investigate how one state led to another, retaining the gaps in the record.

The narrative remains constrained by artifacts. We identify recorded events separately from inferred links, preserve provenance and missingness, and revise the reconstruction when another observation disagrees. Patterns describe repeatable local operations. The narrative records how their results were joined and where the explanation remains conditional.

Several traces or reports may describe the same activity from different viewpoints. We align them through object identity, event relationships, and an explicit model of the activity. Disagreements and unobserved intervals remain visible. The reconstruction is a hypothesis that we can test against another view and revise when that view supplies contrary evidence [1, 4].

To make that reconstruction reviewable, we preserve the inquiry as well as its final diagram or report. A useful record includes the question, artifact versions, commands or queries, filters, intermediate results, alternatives considered, and reasons for accepting or rejecting them. Another analyst can then repeat a step, identify a missing assumption, or continue from the unresolved point. The diagnostic narrative becomes evidence for examining the analysis itself [1, 22].

PART VII

Future directions and conclusion

31. Interoperable evidence ecosystems

We need to join records while retaining the structure of each kind of evidence. Images, waveforms, dumps, traces, specimens, models, maintenance records, and outcomes have different uses. Stable identity, relevant time relations, provenance, and documented meanings let us ask questions across them. DICOM and DICONDE, OpenTelemetry, model registries, and asset histories provide partial examples [1–5, 13, 14, 20, 21].

An evidence graph can connect observations, versions, components, hypotheses, interventions, and outcomes. We give its relations explicit meanings: physical connection, derivation, temporal order, or inferred causality. From a displayed conclusion, we should be able to return to the source artifacts and see which links remain hypothetical. This is the verification work the graph needs to support.

Adaptive capture gives us a way to allocate a limited diagnostic budget. We can retain inexpensive signals continuously and collect a dump, detailed waveform, targeted trace, repeated specimen, focused scan, or agent trajectory when a particular condition appears. We also record what the sampling policy can miss. Some mechanisms require prehistory that cannot be recovered after the trigger, repair, or redeployment [50].

32. Governed diagnostic agents and digital twins

A digital twin needs an identified counterpart and a declared diagnostic or predictive purpose. Configuration, observations, and assumptions must remain related to that object. A diagnostic agent needs corresponding records for its tools, permissions, evidence, hypotheses, and actions. We compare predictions with later outcomes and record when either representation has drifted beyond the conditions of its intended use.

For each agent action, we retain the query, returned observation, retrieved material, inference, and authorization. Read access can support an initial investigation, followed by a proposed bounded intervention when the evidence warrants it. Validation and review increase with consequence. We also specify when the agent should stop or refer the case for another judgment.

The means of observation should expose their limits. Sensors need health checks and telemetry paths need loss measures. Evaluations need coverage and delayed-outcome handling; clinical and engineering workflows need unresolved-result tracking. Twins need checks of prediction error. We compare these reports with partly independent evidence because a self-reported healthy observer can still be wrong.

33. Conclusion

We can now return to the five diagnostic objects. Medicine follows signs through care and outcome. Hardware relates physical faults to encoded errors and service effects. Software reconstructs execution from partial artifacts. ML and AI add learned behavior, data dependence, evaluation, and actions. Technical diagnosis relates measurements to physical integrity and future use. Each history develops ways to obtain evidence and standards for acting on it [1–5].

New methods join the existing repertoire. An examination remains useful with genomic testing, an oscilloscope with fleet telemetry, and a memory dump with distributed tracing. Residual analysis can accompany model interpretation, and visual inspection can accompany structural monitoring. We choose the view that answers the current question and retain the conditions under which it was obtained.

The shared work begins before the failure, with the evidence we arrange to preserve. During the case, we check identity, acquisition, references, and coverage; compare alternatives; and verify the consequences of action. We also preserve unresolved explanations and the analysis that led to the decision. The object and its risks determine how these operations are performed. AI can assist when its evidence and interventions remain inspectable.

Analysis patterns may give us a language for these operations. The software lineage at DumpAnalysis.org supplies examples in memory, trace, and log analysis [28–32]. We can examine an agent's memory with its tool trajectory, relate a logical hardware error to a physical site, or compare medical and technical observations with their different qualifications intact. Appendix C demonstrates specific mappings. Further uses must establish where the resulting procedures are effective [45, 49].

Let us retain the eight stages as questions we can ask of a case: Observe; Measure / preserve; Contextualize; Compare; Infer; Discriminate; Act; Verify and learn. What was available at each stage, what operation was performed, and what did it establish? A diagnosis that can be revisited through its artifacts helps us examine the current conclusion and prepare a better observation path for the next case.

APPENDIX A

Comparative matrix

Table 10. A compact comparison of the five diagnostic strands.

STRAND	PRIMARY QUESTION	CORE EVIDENCE	DISTINCTIVE LIMIT / DUTY
Medical	What condition best explains this person's evidence, and what should care do next?	Narrative, examination, specimens, images, signals, population evidence, follow-up	Reference standards are imperfect; patient agency, communication, equity, and clinical utility are integral
Hardware	Which physical or design fault produced the observed error or service failure?	Waveforms, checks, syndromes, counters, firmware records, topology, returned-part analysis	Correction can hide a fault; silent corruption may escape error reporting; physical analysis may be destructive
Software	What happened in execution, why, and which change will prevent recurrence?	Code, configuration, dumps, logs, metrics, traces, profiles, deployment and incident records	State is volatile and distributed; instrumentation and remediation can perturb the system
ML and AI	Why did learned behavior fail under this data, task, population, or trajectory?	Residuals, splits, gradients, features, attributions, evaluations, drift, feedback, agent actions	Behavior can fail without a software error; metrics and explanations are conditional and must be tested
Technical	What damage, degradation, or integrity state best explains the physical observations?	Inspection, NDE, strain, vibration, thermography, material analysis, loads, condition histories	Detectability, access, qualification, and structural or conservation consequence bound conclusions

APPENDIX B

Shared glossary

Working cross-strand definitions. Domain standards may use narrower meanings.

Abductive reasoning. Inference from observations to one or more plausible explanatory hypotheses, followed by comparison and testing rather than treatment as a deduction.

Abstention. A deliberate decision not to classify, predict, diagnose, or act because evidence, qualification, or authority is insufficient.

Analysis pattern. A reusable transformation, comparison, or interpretive operation that makes diagnostic evidence intelligible and supports recognition or testing of problem patterns.

Anomaly. A departure from a qualified baseline, model, distribution, peer population, or expected relationship.

Automation bias. A tendency to over-rely on automated recommendations or to under-detect automation errors, especially when automation is usually reliable.

Backpropagation telemetry. Monitoring of gradients, activations, losses, parameter updates, and numerical conditions during model training.

Baseline. A reference condition, population, model, version, or period against which later evidence is compared.

Calibration. In metrology, establishing and using a relation between indications and reference values with their uncertainties under stated conditions. In probabilistic prediction, agreement between stated probabilities and observed frequencies.

Capture interval. The period over which an artifact is acquired; its parts may describe different moments unless the acquisition method establishes a consistent state.

Causal isolation. Discrimination among mechanisms that can produce similar symptoms, usually through independent evidence or controlled intervention.

Consequence assessment. Evaluation of what an identified state could do to health, function, safety, mission, environment, cost, or rights.

Context. Identity, configuration, environment, operating state, population, purpose, and time needed to interpret an observation.

Coverage. The portion of relevant states, faults, cases, paths, populations, or physical regions that an observation or test can examine.

Debugging. Controlled investigation of where and how behavior departs from intent, often by stopping, stimulating, substituting, varying, or replaying.

Delayed outcome. An outcome or label that becomes observable after the initial observation or action and must be joined back to the original case or cohort.

Diagnosability. The extent to which available observations and tests can distinguish relevant fault states or explanations within a stated model and set of conditions.

Diagnosis. Evidence-based identification and explanation of the state or mechanism that best accounts for observations.

Diagnostic analysis artifact. A preserved record of queries, transformations, selected evidence, alternatives, decisions, and verification produced during an investigation.

Diagnostic equivalence. In this synthesis, the inability to distinguish candidate conditions from the observations and tests available under stated conditions; another observation or test may separate them. Related terms in fault diagnosis include fault equivalence, ambiguity groups, and indistinguishability; their precise definitions depend on the model.

Diagnostic object. The entity and boundary to which a diagnostic claim applies: person, component, execution, model system, asset, or organization.

Dia|gram language. A visual notation used in Pattern-Oriented Observability to represent structures, events, relations, and transformations across memory, trace, and log evidence.

Digital twin. A maintained digital representation of an identified counterpart, updated with evidence for a declared diagnostic or predictive use.

Drift. Change over time in an instrument, population, data distribution, model relationship, process, or baseline.

Evidence. An observation or preserved artifact whose production path and relevance allow it to support or challenge a claim.

Evidence graph. A structure linking observations, states, versions, components, hypotheses, interventions, and outcomes with typed relationships.

Evidence independence. The degree to which two observations have different acquisition paths and failure modes rather than repeating one underlying signal.

Expected loss. The probability-weighted consequence of possible outcomes under a decision, used to compare tests and actions when errors have asymmetric costs.

Fault. An abnormal physical, design, data, model, or causal condition capable of producing error, damage, or failure.

Feature. A measured or computed property extracted from raw evidence for comparison, classification, estimation, or diagnosis.

First-failure data capture. Preservation of the earliest trustworthy evidence before retry, reset, failover, repair, or replacement changes system state.

Fitness-for-service. A structured engineering assessment of whether damaged or degraded equipment can continue operation under stated conditions and for a stated period.

Iconic Trace. A diagrammatic trace representation in which selected events and relations are expressed through visual icons and spatial organization.

Implementation pattern. A recurring way to realize a diagnostic or analysis operation in a particular tool, instrument, platform, or workflow.

Intervention. A deliberate change in treatment, load, configuration, component, input, model, route, or policy used to discriminate or correct.

Known Uses. Documented cases in which a pattern has been applied, providing evidence of recurrence, transfer, and limitations.

Likelihood ratio. The probability of a test result under the target condition divided by its probability under an alternative condition; it updates prior odds to posterior odds.

Memory–trace duality. The relation between preserved state and the event history through which it arose; each constrains the other, subject to capture timing and coverage.

Metrological traceability. The relation of a measurement result to a reference through a documented calibration chain, with uncertainty contributed by its links.

Missingness. Evidence that is absent, lost, uncollected, uninterpretable, or unavailable; it must not be represented as a negative or normal observation.

Monitoring. Repeated observation of selected conditions and rules to identify change requiring attention or action.

Observability. The designed capacity to infer relevant hidden behavior or state from obtainable outputs and preserved context.

Observation path. The sensors, specimen handling, acquisition, software, transformations, communication, and interpretation through which evidence is produced.

Pattern narrative. A structured composition of diagnostic patterns into an evidence-based explanatory sequence linking anomaly, context, hypothesis, intervention, and verification.

Pattern-oriented diagnostics. A diagnostic discipline that uses explicit repertoires of problem, analysis, implementation, and presentation patterns to structure evidence, interpretation, intervention, and learning.

Pattern-oriented observability. The observability component of pattern-oriented and systemic diagnostics, concerned with how observations are engineered, acquired, represented, related, and interpreted through patterns.

Post-test probability. The probability of a target condition after current test evidence has been combined with pre-test probability.

Pre-test probability. The estimated probability of a target condition before the current test, based on context, prevalence, and prior evidence.

Presentation pattern. A recurring way to render diagnostic evidence or conclusions so that relationships, uncertainty, and provenance remain inspectable.

Prevalence. The proportion of a defined population that has the target condition in a stated setting and period; it influences predictive values.

Probability of detection. The probability that a specified inspection system will detect a flaw of a given size or condition, usually estimated with uncertainty bounds.

Problem pattern. A recurrent diagnostic situation, evidence configuration, or failure manifestation that calls for analysis.

Prognosis. A conditional estimate of future condition, outcome, failure probability, or remaining useful life under stated assumptions.

Prognostics. The discipline of estimating future condition, failure probability, or remaining useful life under stated assumptions.

Provenance. The documented origin and transformation history of evidence, including identity, method, version, processing, and responsible operations.

Reference standard. The method or composite evidence used to establish the target state for comparison; no reference is universally infallible.

Reflexive diagnostics. The observation, diagnosis, and debugging of the diagnostic, observability, or debugging system itself, including sensors, telemetry paths, evaluators, workflows, and automated assistants.

Residual. A discrepancy between observed behavior and the value predicted by a model or expected relation.

Root cause. A causal mechanism or contributing condition at a stated level of analysis; complex systems may have several legitimate causal levels.

Shortcut learning. Use by a learned model of an easy but non-general causal or correlational cue instead of the intended task-relevant relationship.

Silent data corruption. An incorrect computation or stored value that escapes the available error detection and may become visible only through later checks or outcomes.

Sparse feature. A learned or engineered feature that is active for relatively few inputs or dimensions; in interpretability work, such features may be derived from sparse representations.

Stopping condition. A criterion for ending or narrowing an investigation because further expansion is unlikely to change action, responsibility, prevention, or evidential confidence enough to justify its cost.

Threshold. A decision boundary applied to a measurement, feature, probability, score, or risk estimate.

Trace. An ordered record of selected events, operations, states, or actions used to reconstruct behavior over time.

Training–serving skew. A mismatch between data, features, preprocessing, or conditions used during model training and those encountered during deployment.

Uncertainty. Quantified or stated doubt about observation, inference, coverage, model, consequence, or future condition.

Validation. Evidence that a method, model, instrument, workflow, or system is adequate for its intended real-world use.

Verification bias. Distortion of estimated diagnostic performance when receipt of the reference examination depends on the initial test result or related selection factors.

APPENDIX C

Worked cross-strand pattern cases

The following cases examine four proposed transfers. Each identifies context, required evidence, procedure, alternatives, risks, stopping conditions, and verification. We can relate these operations to the continuum in Table 5. The cases are illustrative comparisons rather than reports of actual patients or incidents. Hardware is examined explicitly in the fourth case.

Worked case 1. Observation-Path Diagnosis: NDE sensor drift and a dropped-span telemetry pipeline

Intent. Check whether the apparent change in the diagnosed object arose in the observation path.

Table 11. Observation-Path Diagnosis: NDE sensor drift and a dropped-span telemetry pipeline.

FIELD	Technical diagnostics: ultrasonic/NDE channel	Software diagnostics: distributed trace
Context	A structural-health or NDE channel reports increasing indication amplitude on an asset whose load regime appears unchanged.	A service trace shows a missing downstream span during an incident.
Required Evidence	Asset and sensor identity; calibration and reference-block results; coupling, temperature, procedure, operator, raw waveform, prior inspections, and independent modality.	Trace and span identifiers; propagation headers; instrumentation and collector versions; sampling policy; export and drop counters; service logs and metrics.
Transformation / Procedure	Re-run reference standard; compare raw and derived signals; swap or independently calibrate sensor; repeat under controlled geometry; compare with another method.	Inspect context propagation; compare logs and metrics for the supposed call; check sampler and collector loss; reproduce with controlled instrumentation; inspect an unsampled or locally captured request.
Alternatives and Counter-evidence	Real damage growth, load change, coupling change, material or temperature effect, processing change.	The downstream call never occurred; asynchronous work lost parentage; instrumentation was absent; the service failed before span creation.
Observer Effects and Risks	Repeated loading or invasive access may extend damage; recalibration can overwrite the failing state.	Extra tracing can change timing, cost, and cardinality; enabling full capture may expose sensitive data.

FIELD	Technical diagnostics: ultrasonic/NDE channel	Software diagnostics: distributed trace
Stopping Conditions	Stop observation-path inquiry when independent calibration and modality agree that the channel is healthy, or when correcting the channel removes the anomaly and the asset result returns to baseline.	Stop when propagation, sampling, and collection are verified for a reproduced request, or when repairing the telemetry path restores the missing causal relation without changing service behaviour.
Verification	A known reference is measured correctly; an independent channel agrees; subsequent asset trend remains coherent.	Controlled requests yield complete linked spans; drop counters remain zero or quantified; incident interpretation agrees with logs and metrics.
Known Uses	Calibration checks, probability-of-detection qualification, sensor plausibility, repeat inspection.	Trace-context validation, instrumentation health checks, collector monitoring, loss and sampling audits.
Where the analogy breaks	The sensor interacts with material, geometry, coupling, and physical access; inspection can alter the asset.	The path is symbolic and software-defined; causality may be partial, and instrumentation can be changed without touching the physical service.

Evidence base: [13, 14, 18, 26, 50].

Worked case 2. Baseline Comparison: a personal medical trajectory and an ML holdout distribution

Intent. Compare current evidence with a qualified reference while preserving the reference's population, time, method, and decision purpose.

Table 12. Baseline Comparison: a personal medical trajectory and an ML holdout distribution.

FIELD	Medical diagnostics: within-person biomarker trend	ML/AI diagnostics: production distribution versus holdout
Context	A patient’s biomarker rises but remains inside a broad population reference interval.	A deployed classifier retains overall accuracy while confidence and subgroup performance change.
Required Evidence	Repeated results from comparable methods; specimen conditions; treatment and symptom timeline; population interval; analytical variation; intended clinical decision.	Training and holdout definitions; feature and label distributions; model and pipeline versions; calibration and subgroup metrics; serving inputs and outcomes.

FIELD	Medical diagnostics: within-person biomarker trend	ML/AI diagnostics: production distribution versus holdout
Transformation / Procedure	Estimate within-person change relative to analytical and biological variation; compare with population interval; update pre-test probability and consequences.	Compare production and holdout distributions; stratify cohorts; examine calibration and outcome frequencies; test whether drift is data, label, pipeline, or model related.
Alternatives and Counter-evidence	Method change, specimen handling, transient physiology, treatment effect, unrelated comorbidity, population interval inappropriate to the person.	Logging or labelling change, delayed outcomes, selection bias, feature pipeline defect, seasonal change, benign covariate shift.
Observer Effects and Risks	Repeat testing can trigger cascades, anxiety, cost, or treatment changes that alter the trajectory.	Monitoring choices can leak labels, influence retraining data, or optimise only measured cohorts.
Stopping Conditions	Stop when the available evidence changes or confirms the clinical action and further testing has lower expected value than follow-up or treatment.	Stop when the responsible layer is isolated sufficiently to choose rollback, recalibration, retraining, abstention, or continued monitored operation.
Verification	Follow-up, orthogonal test, treatment response, or outcome behaves as predicted.	Prospective production outcomes and calibration improve after the chosen intervention without unacceptable subgroup harm.
Known Uses	Serial biomarkers, personal baselines, delta checks, repeated imaging and vital signs.	Drift detection, calibration monitoring, holdout and shadow evaluation, subgroup surveillance.
Where the analogy breaks	The reference concerns a person with agency and care consequences; “normal” is not purely statistical.	The reference distribution is engineered and can be regenerated, but labels and environments may change after deployment.

Evidence base: [8, 10, 15, 16].

Worked case 3. Temporal Reconstruction: a crash dump and an agent trajectory

Intent. Join a snapshot to an ordered or partially ordered history to investigate the sequence that produced it.

Table 13. Temporal Reconstruction: a crash dump and an agent trajectory.

FIELD	Software diagnostics: crash dump plus telemetry	Agentic AI diagnostics: memory state plus tool trajectory
Context	A process crashes in an allocator with a corrupted heap; the dump preserves the terminal state but not the earlier write.	An agent produces an unsafe side effect; its final response does not reveal how context, tools, and memory produced the action.
Required Evidence	Exact binaries and symbols; dump; thread and heap state; logs, traces, profiles, deployment and configuration history; clock and loss information.	Model and policy identifier; system and task instructions; retrieved context; memory checkpoints; tool calls and results; retries, hand-offs, timestamps, side effects, and missing-event markers.
Transformation / Procedure	Reconstruct identity and terminal state; align prior events; search backward from corrupted structures; compare healthy runs; replay or instrument a minimized reproduction.	Join memory snapshots to trajectory events; establish partial order and causal links; locate context loss, loop, unsuitable tool choice, or policy failure; replay safely where possible.
Alternatives and Counter-evidence	Allocator defect, earlier overwrite, symbol mismatch, dump truncation, observer-induced schedule change.	Tool error, stale retrieval, policy misconfiguration, memory overwrite, missing trace step, external environment change, misleading generated explanation.
Observer Effects and Risks	Instrumentation and replay can change scheduling; dumps contain sensitive memory.	Full trajectory capture can expose sensitive prompts and data; replay may repeat side effects; generated rationales are not ground truth.
Stopping Conditions	Stop when a reproducible transition explains the corrupted state and a corrective change prevents recurrence, while remaining alternatives are bounded.	Stop when the evidence-supported trajectory identifies a controllable failure point and a safe intervention changes the predicted outcome; do not claim hidden reasoning beyond recorded behaviour.
Verification	Regression test, deterministic replay, sanitizer, or production outcome confirms prevention.	Sandbox replay, counterfactual prompt or tool restriction, and monitored deployment confirm safer behaviour.
Known Uses	Post-mortem dump analysis, record-and-replay, distributed incident timelines.	Agent observability, tool-call traces, memory audits, trajectory evaluation, side-effect review.

FIELD	Software diagnostics: crash dump plus telemetry	Agentic AI diagnostics: memory state plus tool trajectory
Where the analogy breaks	Program execution may be deterministically replayable and grounded in exact code and machine state.	Generative behaviour may be stochastic; model internals and natural-language rationale do not provide a stable, complete causal account.

Evidence base: [1, 4, 29–31, 45].

Worked case 4. Cross-layer correlation: silent computation error and a training regression

Intent. Test whether a behavioral anomaly in a training job has a hardware, software, data, or optimization mechanism, while keeping the evidence at each layer distinct. We use the Multimodal Correlation family in Table 8 to relate these records without treating agreement across layers as an established cause.

Table 14. Cross-layer correlation across hardware, software, and ML training.

FIELD	Hardware and software execution	ML training and outcome
Context	One host or accelerator is suspected after intermittent incorrect arithmetic; ordinary error reports are absent or inconclusive.	A training run sometimes diverges or produces a different checkpoint, but the loss curve alone does not identify a failing layer.
Required Evidence	Job, rank, host, device, core, firmware, driver, topology, ECC and machine-check records, workload, temperature, and first-failure capture.	Exact data and batch identifiers; code and model versions; seeds; optimizer state; checkpoints; intermediate results; loss curves and evaluation outcomes.
Transformation / Procedure	Align rank and device identity; repeat targeted correctness tests under recorded conditions; compare suspect and known-good resources; retain raw outputs and mapping.	Replay a controlled batch where feasible; compare intermediate computations and checkpoints; separate data, numerical, pipeline, and resource explanations.
Alternatives and Counter-evidence	Software defect, unstable kernel, network or storage corruption, reporting-path failure, or a workload-dependent error not reproduced by a general test.	Data or label change, optimizer instability, nondeterministic operations, changed preprocessing, or a harmless variation in training.

FIELD	Hardware and software execution	ML training and outcome
Observer Effects and Risks	Moving a job changes load and temperature; a reset can remove the first-failure state; repeated tests may not exercise the triggering condition.	Replay and extra capture can be costly, expose training data, or change scheduling; a contaminated checkpoint can propagate an earlier error.
Stopping Conditions	Stop at a resource-level conclusion only when controlled comparison ties the incorrect computation to that resource within the tested conditions.	Stop when a discriminating test selects an actionable layer, or report the mechanism unresolved and contain the affected job or resource.
Verification	A known-good comparison and targeted test reproduce or exclude the suspected behavior; follow-up checks after isolation do not depend only on missing alarms.	The chosen correction changes the predicted intermediate result and prospective training outcome without a new regression.
Known Uses	Correctness testing of intermittently faulty processors, first-failure capture, fleet comparison, and returned-part investigation.	Checkpoint comparison, training-job diagnosis, batch replay, and correlation of runtime telemetry with learning behavior.
Where the analogy breaks	A physical fault may require laboratory proof and may appear only under narrow conditions; no counter proves its absence universally.	A loss curve is a derived behavioral signal with many possible causes; model quality also depends on data, task, and population.

Evidence base: the hardware and ML companion histories [2, 4]. This is a constructed diagnostic scenario, not a report of a particular fleet incident.

References

References are numbered by first citation. The five companion manuscripts are identified by their supplied version numbers; linked online copies may lag those versions. Papers, standards, specifications, and guidance support the more specific comparisons. An access date records consultation of a source and does not establish that a cited standard remains the current edition.

[1] Vostokov, Dmitry. History of Software Diagnostics, Observability, and Debugging. Online history; Version 10 companion manuscript supplied for this synthesis, 2026.

[2] Vostokov, Dmitry. History of Hardware Diagnostics, Observability, and Debugging. Online history; Version 14 companion manuscript supplied for this synthesis, 2026.

[3] Vostokov, Dmitry. History of Medical Diagnostics, Observability, and Debugging. Online history; Version 7 companion manuscript supplied for this synthesis, 2026.

[4] Vostokov, Dmitry. History of ML and AI Diagnostics, Observability, and Debugging. Online history; Version 7 companion manuscript supplied for this synthesis, 2026.

[5] Vostokov, Dmitry. History of Technical Diagnostics, Observability, and Debugging. Online history; Version 7 companion manuscript supplied for this synthesis, 2026.

[6] Kalman, R. E. “On the General Theory of Control Systems.” Proceedings of the First International Congress of the International Federation of Automatic Control, Moscow, 1960 (London: Butterworths, 1961), vol. 1, pp. 481–492. Accessed 2 August 2026.

[7] Sampath, M., R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. “Diagnosability of Discrete-Event Systems.” IEEE Transactions on Automatic Control 40, no. 9 (1995): 1555–1575. doi:10.1109/9.412626. Accessed 27 September 2026.

[8] U.S. Food and Drug Administration. Biomarker Qualification: Evidentiary Framework. Draft Guidance for Industry and FDA Staff, December 2018. Accessed 2 August 2026.

[9] Centers for Disease Control and Prevention. “ACCE Model Process for Evaluating Genetic Tests.” Office of Public Health Genomics, n.d. Accessed 26 September 2026.

[10] Fagan, T. J. “Nomogram for Bayes’s Theorem.” New England Journal of Medicine 293, no. 5 (1975): 257. Accessed 2 August 2026.

[11] Welch, H. G., and W. C. Black. “Overdiagnosis in Cancer.” Journal of the National Cancer Institute 102, no. 9 (2010): 605–613. doi:10.1093/jnci/djq099. Accessed 27 September 2026.

[12] Avižienis, A., J.-C. Laprie, B. Randell, and C. Landwehr. “Basic Concepts and Taxonomy of Dependable and Secure Computing.” IEEE Transactions on Dependable and Secure Computing 1, no. 1 (2004): 11–33. doi:10.1109/TDSC.2004.2. Accessed 2 August 2026.

[13] World Wide Web Consortium. Trace Context, W3C Recommendation, 23 November 2021. Accessed 2 August 2026.

[14] OpenTelemetry. OpenTelemetry Specification 1.61.0. Accessed 26 September 2026.

[15] Sculley, D., et al. “Hidden Technical Debt in Machine Learning Systems.” Advances in Neural Information Processing Systems 28 (2015): 2503–2511. Accessed 2 August 2026.

[16] Guo, C., G. Pleiss, Y. Sun, and K. Q. Weinberger. “On Calibration of Modern Neural Networks.” Proceedings of the 34th International Conference on Machine Learning, PMLR 70 (2017): 1321–1330. Accessed 2 August 2026.

[17] Isermann, R. “Fault Diagnosis of Machines via Parameter Estimation and Knowledge Processing—Tutorial Paper.” Automatica 29, no. 4 (1993): 815–835. Accessed 2 August 2026.

[18] National Aeronautics and Space Administration. NASA-STD-5009C: Nondestructive Evaluation Requirements for Fracture-Critical Metallic Components. 2023. Accessed 2 August 2026.

[19] API 579-1/ASME FFS-1. Fitness-For-Service. 4th ed., 2021. Accessed 2 August 2026.

[20] DICOM Standards Committee. Digital Imaging and Communications in Medicine (DICOM), current edition. Accessed 2 August 2026.

[21] ASTM International. ASTM E2339-21: Standard Practice for Digital Imaging and Communication in Nondestructive Evaluation (DICONDE). 2021. Accessed 2 August 2026.

[22] Vostokov, Dmitry. Theoretical Software Diagnostics: Collected Articles. 4th ed. OpenTask, 2024. “Five Golden Rules,” p. 29, for the 4WS questions. Supplied manuscript consulted. Accessed 27 September 2026.

[23] Vömel, S., and F. Freiling. “Correctness, Atomicity, and Integrity: Defining Criteria for Forensically-Sound Memory Acquisition.” Digital Investigation 9, no. 2 (2012): 125–137. doi:10.1016/j.diin.2012.04.005. Accessed 27 September 2026.

[24] Joint Committee for Guides in Metrology. International Vocabulary of Metrology—Basic and General Concepts and Associated Terms (VIM). 3rd ed., 2008 version with minor corrections, JCGM 200:2012. Definition 2.41, metrological traceability. Accessed 27 September 2026.

[25] Begg, C. B., and R. A. Greenes. “Assessment of Diagnostic Tests When Disease Verification Is Subject to Selection Bias.” Biometrics 39, no. 1 (1983): 207–215. doi:10.2307/2530820. Accessed 27 September 2026.

[26] Parker, P. A., et al. Guidebook for the Design and Analysis of a NASA Standard Nondestructive Evaluation Probability of Detection Study. NASA/TM-20220013822, 2022. Accessed 2 August 2026.

[27] de Kleer, J., and B. C. Williams. “Diagnosing Multiple Faults.” Artificial Intelligence 32, no. 1 (1987): 97–130. Accessed 2 August 2026.

[28] Pattern-Oriented Diagnostics and Observability as a Philosophy of Engineering. DumpAnalysis.org, n.d. Accessed 2 August 2026.

[29] Pattern-Oriented Observability (Part 1): The Base Slab and Foundation of Observability Pillars. DumpAnalysis.org, n.d. Accessed 2 August 2026.

[30] Pattern-Oriented Observability (Part 2): Semiotics of Memory, Trace, and Log Analysis. DumpAnalysis.org, n.d. Accessed 2 August 2026.

[31] Pattern-Oriented Observability (Part 3): Observation Spaces. DumpAnalysis.org, n.d. Accessed 27 September 2026.

[32] Pattern-Oriented Observability (Part 4): Diagnostics–Observability Square (DOS). DumpAnalysis.org, n.d. Accessed 2 August 2026.

[33] Alexander, C., S. Ishikawa, M. Silverstein, et al. A Pattern Language: Towns, Buildings, Construction. New York: Oxford University Press, 1977. Accessed 2 August 2026.

[34] Gamma, E., R. Helm, R. Johnson, and J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison-Wesley, 1995. Accessed 2 August 2026.
[35] Fowler, M. Analysis Patterns: Reusable Object Models. Reading, MA: Addison-Wesley, 1997. Accessed 2 August 2026.
[36] Vostokov, Dmitry. Trace, Log, Text, Narrative, Data: An Analysis Pattern Reference for Information Mining, Diagnostics, Anomaly Detection. 5th ed. OpenTask, 2023. Supplied manuscript consulted. Accessed 27 September 2026.
[37] Reiter, R. “A Theory of Diagnosis from First Principles.” Artificial Intelligence 32, no. 1 (1987): 57–95. Accessed 2 August 2026.
[38] Gertler, J. Fault Detection and Diagnosis in Engineering Systems. New York: Marcel Dekker, 1998. Accessed 2 August 2026.
[39] NASA Goddard Space Flight Center. GSFC-HDBK-8004: Guideline for Failure Modes and Effects Analysis and Risk Assessment. 2024. Accessed 2 August 2026.
[40] Pauker, S. G., and J. P. Kassirer. “The Threshold Approach to Clinical Decision Making.” New England Journal of Medicine 302, no. 20 (1980): 1109–1117. doi:10.1056/NEJM198005153022003. Accessed 26 September 2026.
[41] Bainbridge, L. “Ironies of Automation.” Automatica 19, no. 6 (1983): 775–779. Accessed 2 August 2026.
[42] Parasuraman, R., and V. Riley. “Humans and Automation: Use, Misuse, Disuse, Abuse.” Human Factors 39, no. 2 (1997): 230–253. Accessed 2 August 2026.
[43] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1, 2023. Accessed 2 August 2026.
[44] Parasuraman, R., and D. H. Manzey. “Complacency and Bias in Human Use of Automation: An Attentional Integration.” Human Factors 52, no. 3 (2010): 381–410. doi:10.1177/0018720810376055. Accessed 26 September 2026.
[45] Higher-Order Pattern Narratives (Analysing Diagnostic Analysis). DumpAnalysis.org, n.d. Accessed 2 August 2026.
[46] Schwartz, R., et al. Towards a Standard for Identifying and Managing Bias in Artificial Intelligence. NIST Special Publication 1270, 2022. Accessed 2 August 2026.
[47] Kelsey, J., J. Callas, and A. Clemm. RFC 5848: Signed Syslog Messages. Internet Engineering Task Force, May 2010. Accessed 2 August 2026.
[48] Biggio, B., B. Nelson, and P. Laskov. “Poisoning Attacks against Support Vector Machines.” Proceedings of the 29th International Conference on Machine Learning, 2012. Accessed 2 August 2026.
[49] Unified Computer Diagnostics: Incorporating Hardware Narratology. DumpAnalysis.org, n.d. Accessed 2 August 2026.
[50] OpenTelemetry. “Sampling.” OpenTelemetry Documentation, updated 16 October 2025. Accessed 2 August 2026.

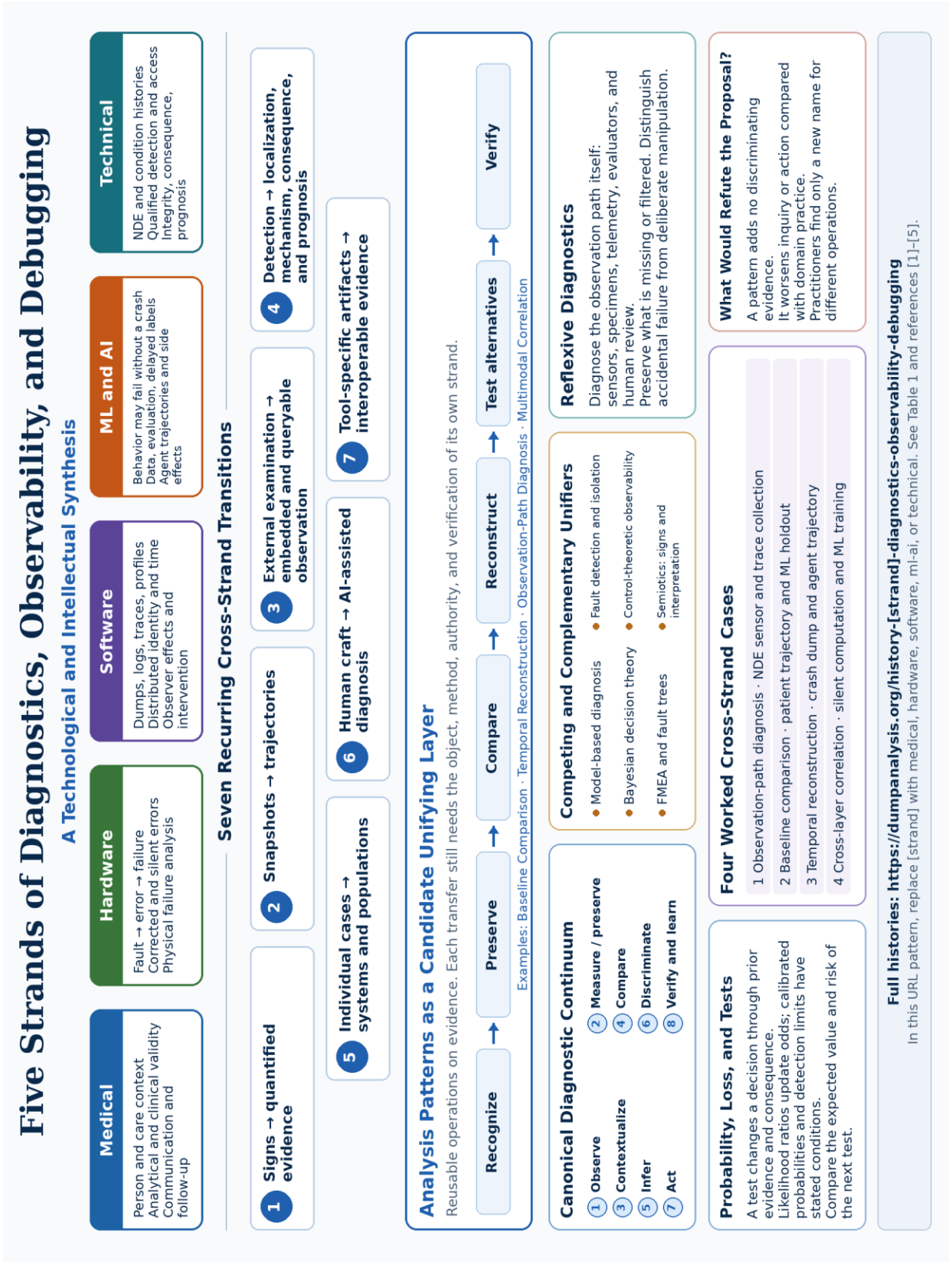


Figure 1. Five strands: seven transitions, analysis patterns, unifiers, reflexive diagnostics, continuum, and four cases.