HOLISTIC ORGANIZATIONAL INTELLIGENCE PLATFORM WITH DETERMINISTIC QUERY SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of priority under 35 U.S.C. § 119(e) to U.S. Provisional Application No. 63/904,593, filed October 24, 2025, entitled "Holistic Organizational Intelligence Platform with Unified Experience Tracking and Hallucination-Free AI Query System," the entire contents of which are incorporated herein by reference.

[0002] Subject matter disclosed in the aforementioned provisional application is entitled to the October 24, 2025 filing date. Subject matter first disclosed in the present application — including ambiguity-aware intent classification with margin-gated retrieval, machine-readable routing provenance, routing regression validation, deterministic enrichment at persistence keyed on verb-URI and object-type pairs, privacy-segregated ledger identity resolution, competency decay scoring, cross-domain readiness traversal, procedure drift detection, tacit-knowledge elicitation into a governed ontology, and multi-class actor attribution encompassing robotic, botanical, and zoological actors — is entitled to the filing date of the present application.

TECHNICAL FIELD

[0003] The disclosure relates to enterprise systems for organizational intelligence in which activity arising across learning, operations, maintenance, safety investigation, program management, human resources, and communications functions is recorded into a single auditable ledger carrying machine-navigable semantic context embedded at the moment of persistence; in which the identity of an acting entity is structurally separated from personally identifying information; in which acting entities are not restricted to human participants; and in which natural-language access to the ledger is mediated by a classification stage that does not produce factual assertions, an ambiguity evaluation that withholds retrieval on under-determined requests, and a deterministic retrieval stage that produces the factual content of every response.

BACKGROUND

[0004] Modern enterprises operate eight to twelve disconnected systems spanning learning management, project management, maintenance operations, document management, human resources, communications, incident tracking, and business intelligence. The consequence is not merely duplicated data entry but the loss of the relational context that gives records meaning. When a safety incident occurs, the investigation report resides in one system, the training records of involved personnel in a second, the equipment maintenance history in a third, and the photographic evidence in a fourth. The relationships among these points are represented nowhere, making it structurally impossible to identify recurrence patterns, to verify that a corrective action was effective, or to attribute an outcome to a training investment.

[0005] Learning management systems illustrate the defect. They capture course delivery and completion through content-packaging conventions, implement only the reaction and assessment tiers of four-tier evaluation models, and possess no mechanism for observing workplace behavior change or business result. More significantly they are unaware of the operational reality they exist to improve: an equipment failure that injures an operator generates no signal in the learning system, which therefore cannot determine whether the operator held current training, cannot enroll affected personnel in revised instruction, and cannot measure whether the instruction reduced recurrence. Organizations bridge the gap with extract-transform-load pipelines into warehouses, at integration cost frequently exceeding the license cost, and the context that made the data meaningful is discarded during transformation.

[0006] Activity-stream and experience-statement conventions were developed to address interoperability of learning records. Such conventions define a statement comprising an actor who performed an action, a verb describing the action, an object acted upon, a result, and a context. Two structural assumptions limit them. First, the actor is presumed to be a human learner, so activity originating from a machine, a robotic platform, a monitored biological asset, or an automated process either cannot be represented or must be misrepresented as human activity. Second, statements are conventionally deposited into a record store separate from the operational database and carry no semantic categorization beyond the identity of the instructional object, so any cross-domain grouping must be inferred after the fact by a classification pass over statement content — a pass whose errors are silent and whose results change whenever the classifier changes.

[0007] Retrieval-augmented generation has been proposed to provide natural-language access to enterprise data. Surveys of the field identify a persistent set of failure modes: retrieval of passages topically similar but factually inapposite; generation of assertions not entailed by retrieved context; opaque selection among retrieval strategies; and absence of any artifact permitting an auditor to determine, after the fact, why a particular retrieval path was taken. These failure modes share a structural cause. The generative component is permitted to produce the factual assertion. Where the assertion is wrong, no artifact identifies the step at which correctness was lost.

[0008] A related and less-examined failure mode arises in the selection of retrieval paths. Where a request is matched against a set of candidate retrieval targets and the first satisfied match is accepted, a generic match expression evaluated early will capture requests intended for a specific expression evaluated later. The resulting misroute is silent: the system returns a confident, internally consistent, fully formatted answer to a question that was not asked. No error is raised, no confidence degradation is observable, and the defect is undetectable from the response alone.

[0009] Prior Art Reference. U.S. Patent No. 12,272,265 B2 describes systems for generating and analyzing learning-experience data using activity-stream statement formats, in which learner

activity is recorded and machine-learning models operate over the recorded activity to characterize learner progress. That disclosure is directed to learners: its actor is a human undergoing instruction, its statements arise from instructional interactions, and its analysis is confined to the learning domain.

[0010] The present disclosure is distinguished from the aforementioned reference in at least the following respects, each of which is recited as a structural element of an independent claim rather than as an optional feature. First, the reference does not disclose embedding semantic categorization into a statement by deterministic rules at the moment of persistence, such that cross-domain grouping is available at query time without a post-hoc inference pass and such that embedded context is immutable against subsequent modification of the rules. Second, the reference does not disclose an actor registry in which software inference agents, automated processes, robotic platforms, botanical assets, and zoological assets emit statements into the same ledger as human participants and are resolved by a single audited resolution function, nor any reasoning that traverses two or more such actor classes in one operation. Third, the reference does not disclose separating actor identity from personally identifying information such that the analytical ledger contains no direct identifier and an audited resolution function is the sole computational path between them. Fourth, the reference does not disclose interposing an ambiguity evaluation between intent classification and data retrieval such that an under-determined request yields an enumerated clarification rather than a confident retrieval along an arbitrarily selected path. Fifth, the reference does not disclose computing an aggregate authorization score from confidence values carried by the individual statements evidencing each link of a qualification chain, as distinct from a confidence attribute stored on a credential record.

SUMMARY OF THE INVENTION

[0011] A system persists organizational activity as experience statements in a unified ledger. Each statement identifies an actor, a verb expressed as a governed uniform resource identifier, and an object having an object type, and carries a result structure whose extension fields hold the operational specifics that make the statement analytically useful — geospatial position and accuracy, work-record and document identifiers, consumed materials, labor duration, environmental conditions, and, where the statement was produced by an inference process, that process's own confidence value.

[0012] On occurrence of an entity-change event, a deterministic enrichment engine applies auditable rules keyed on the pair comprising the governed verb identifier and the object type, resolves the pair to one or more concept identifiers in a governed ontology, computes a mapping confidence and per-edge confidences, records a mapping source identifier, and embeds the resulting semantic context into the statement at the moment the statement is persisted. Because enrichment occurs at write time by reproducible rules, a later query navigates embedded context

directly with no separate enrichment pass, and the context embedded in an existing record is immutable against subsequent revision of the rules.

[0013] Actor identity is generalized and privacy-segregated. An actor may be a human participant, a software inference agent, an automated system process, a robotic platform, a botanical asset, or a zoological asset; every class registers in a common registry and is resolved by a single resolution function that logs each resolution. Statements carry only a stable identifier bearing no correspondence to a natural-person attribute. Personally identifying information, where it exists, resides in a schema physically and logically separate from the ledger, reachable only through an audited resolution function that verifies role authorization, logs each access, and denies or returns null-valued fields absent authority. The ledger is therefore structurally free of direct identifiers, and analytics, export, and inter-organizational sharing never traverse identifying data.

[0014] Natural-language access proceeds in two parts. A first part receives the request and produces a classification comprising an intent category, the implicated collections, the extracted constraints, and a methodology description; the first part does not produce a factual assertion. A second part selects a deterministic retrieval function, authenticates the requesting principal, executes retrieval under row-level access rules evaluated against that principal, aggregates by fixed computation, and returns structured results together with provenance comprising statement identifiers and concept mappings. Where data sufficient to answer is absent, the system records a structured gap and returns a data-missing indication; it is structurally configured to do so rather than to return a fabricated value.

[0015] Function selection is itself made auditable and ambiguity-aware. The classifier evaluates the request against every candidate retrieval target and collects all matches rather than accepting the first, scores each candidate, and applies a confidence floor and a runner-up margin threshold. Failing either test, the system executes no retrieval function and returns a clarification response enumerating the candidates, each described by a structured intent descriptor comprising a target label and exclusion terms. On receiving a selection, the system re-invokes the second part with the selected intent, bypassing re-classification. Every result carries machine-readable provenance identifying the target routed to, the reason for routing, the candidates considered, the routing confidence, and the structured intent descriptor. A stored regression corpus of requests paired with expected targets is evaluated against the routing table so that a newly added or broadened expression that shadows an existing one is reported as a regression rather than silently misrouting production requests.

[0016] Embedded semantic context supports inference operations impossible over uncategorized records. A cross-domain readiness traversal follows a chain from a worker, through a credential, a personnel qualification, a governing procedure, to an active permit, where each traversed edge carries its confidence in the statement that evidences that edge, and computes an aggregate

authorization score from those statement-borne confidences rather than from any confidence attribute on the credential record. A competency decay evaluation locates, via embedded edges, a credential issuance timestamp, a most-recent peer-endorsement statement, and a most-recent procedure-execution statement, computes time deltas, and generates a predictive decay score reported together with the statement identifiers from which it was derived. A procedure drift evaluation compares recorded executions against a governed step sequence and reports steps executed below a threshold count, out of order, or omitted. A tacit-knowledge elicitation captures expertise from a designated subject-matter expert through a structured interview protocol and persists it as concept nodes and graph edges in the same governed ontology traversed by the inference operations and queried by the deterministic query system, whereby expertise becomes queryable infrastructure rather than an unstructured transcript.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIG. 1 is a system architecture diagram showing entity-change events, the deterministic enrichment engine, the governed ontology, the unified experience ledger, the separated identity store, and the query and inference subsystems.

[0018] FIG. 2 is a block diagram of the experience statement structure, showing the actor identifier, governed verb identifier, object type, result structure with extension fields, and the semantic context embedded at persistence.

[0019] FIG. 3 is a flow diagram of deterministic semantic enrichment at persistence, showing rule application keyed on verb identifier and object type, concept resolution, confidence computation, embedding prior to commit, and the read path requiring no post-hoc enrichment.

[0020] FIG. 4 is a flow diagram of privacy-segregated identity resolution, showing the ledger read path, the separate identity schema, and the audited resolution function as sole computational path between them.

[0021] FIG. 5 is a diagram of cross-domain readiness traversal, showing the chain from worker to credential to personnel qualification to governing procedure to active permit, with the evidencing statement and its confidence shown for each traversed edge.

[0022] FIG. 6 is a flow diagram of competency decay scoring, showing location of credential, endorsement, and execution evidence via embedded edges, time-delta computation, score generation with provenance, and re-qualification flagging.

[0023] FIG. 7 is a flow diagram of the two-part deterministic query, showing classification without answering, deterministic retrieval under row-level access enforcement, result composition with provenance, and the structured-gap path returning a data-missing indication.

[0024] FIG. 8 is a flow diagram of procedure drift detection, showing comparison of recorded executions against the governed step sequence, the three drift conditions, and finding generation with provenance and review routing.

[0025] FIG. 9 is a flow diagram of tacit-knowledge capture, showing structured interview, extraction, concept-node and edge generation, persistence, and consumption by the inference and query subsystems.

[0026] FIG. 10 is a flow diagram of ambiguity-aware classification, showing candidate-set collection, threshold and margin evaluation, the clarification path with structured intent descriptors, selection receipt, re-invocation bypassing re-classification, and provenance emission.

[0027] FIG. 11 is a block diagram of the multi-class actor registry, showing six actor classes converging on a common ledger and a single audited resolution function, and cross-class grouping by semantic context.

Detailed Description

1. System Architecture

[0028] FIG. 1 shows the system. An entity-change event (110) arises when a domain record is created, updated, or deleted, or when an external system reports activity through an ingestion endpoint. A statement emission function (112) receives the event and invokes a deterministic enrichment engine (120), which applies auditable rule modules (122) against a governed ontology (130). The governed ontology comprises a concept scheme (132) defining a namespace and governance authority, concepts (134) each addressed by a uniform resource identifier, concept relationships (136) expressing typed edges between concepts, and concept mappings (138) associating external identifiers with governed concepts. The enrichment engine writes the resulting statement, with semantic context embedded, into a unified experience ledger (140) as a ledger record (142).

[0029] The ledger contains no personally identifying information. An identity store (150), physically and logically separate from the ledger, holds an actor registry (152) and mapping records (154) associating a stable ledger identifier with identifying information. An audited resolution function (156) is the sole computational path between the ledger identifier and the identifying information. A query subsystem (160) and an inference subsystem (170) read the ledger exclusively by stable ledger identifier and never traverse the identity store.

2. Experience Statement Structure

[0030] FIG. 2 shows the statement (200). The actor is represented solely by a stable non-identifying identifier (210). The verb is expressed as a governed uniform resource identifier (212) drawn from a verb registry, permitting a controlled and extensible vocabulary; in one implementation a registry of over three thousand verb identifiers is maintained. The object

comprises an object type (214), being an identifier of the domain entity class, and an object identifier (216).

[0031] The result structure (218) carries extension fields (232) holding operational specifics: geospatial latitude, longitude, and accuracy for field activity; a work-record identifier linking maintenance activity to a service request; a document identifier connecting activity to reference material; consumed materials; labor duration; environmental conditions; and, where the statement was produced by an inference process, a confidence value attributable to that process. A timestamp (234) records the instant of the underlying activity, distinct from the instant of persistence.

[0032] The semantic context (220) is the structure embedded at persistence. It comprises a categorical domain label (222) drawn from a controlled set including field operations, real-world event, knowledge management, incident investigation, and strategic planning; a set of concept identifiers (224) in the governed ontology to which the statement is mapped; a set of typed edges (226) each carrying an edge confidence (236); a mapping source identifier (228) recording which rule module produced the mapping; and a mapping confidence (230) for the mapping as a whole. Grouping and correlation by domain, concept, or edge type is therefore available at query time by direct field access.

[0033] It is emphasized that the semantic context is not an annotation added later. The record is not committed until the context is written. A consequence is immutability: revision of a rule module changes the enrichment of statements persisted after the revision and leaves earlier statements bearing the context assigned under the rules in force when they were written, so a historical query returns what the system understood at the time, which is the property an auditor requires.

3. Deterministic Semantic Enrichment at Persistence

[0034] FIG. 3 shows the enrichment path. An entity-change event (310) is dispatched (312) to the enrichment engine, which selects and applies deterministic auditable rules (314) keyed on the pair comprising the governed verb identifier and the object type. The keying is significant: enrichment is a function of what was done and to what class of thing, not of free text, so the mapping is reproducible from its inputs and reviewable by inspection of the rule module.

[0035] Rule application resolves the pair to one or more concept identifiers in the governed ontology (316) and to typed edges relating the statement to those concepts. A confidence computation (318) produces a mapping confidence for the mapping as a whole and an edge confidence for each edge, derived from the specificity of the matched rule and the completeness of the resolved concepts. A mapping source identifier records the rule module and its version tag (324).

[0036] The engine embeds the resulting context into the statement (320) and only then commits the record (322). No path exists by which a statement is persisted without semantic context. On the read side (326), query and inference operations access the embedded fields directly; there is no post-hoc enrichment pass, and consequently no class of error in which a read-time classifier mis-assigns a domain to a historical record.

[0037] Given identical inputs the rules produce identical outputs. A reviewer may therefore re-run a rule module against a recorded input pair and confirm that the embedded context is the context the rules produce, which is what makes the enrichment auditable rather than merely automatic.

4. Privacy-Segregated Identity Resolution

[0038] FIG. 4 shows identity handling. A request (410) arriving at the query subsystem is served entirely from the ledger read path (420), which resolves actors only by stable ledger identifier. A separate identity schema (450) holds identifying information. The audited resolution function (430) is the sole computational path between them.

[0039] On invocation the resolution function performs a role authorization check (432) against a designated authority for the class of information sought. Absent that authority the function denies (434), returning null-valued fields, and where the request originated from an inference agent, sets an attempted-access indicator and a blocked indicator. With authority present the function returns the identifying information (436). Every invocation, granted or denied, is written to an access audit log (442) recording the requesting principal, the ledger identifier sought, the fields requested, the outcome, and the timestamp.

[0040] The architectural consequence is that a retrieval returning many thousands of statements discloses no identity, because no identity is present in the returned data. Disclosure is a separate, individually authorized, individually logged act. Analytics, export, benchmarking, and inter-organizational sharing operate on the ledger and therefore cannot leak identifying data, because there is none to leak. Erasure obligations are satisfied by operating on the identity store, leaving the analytical record intact and unattributable.

[0041] This structure is distinguished from field-level masking, in which identifying values reside in the analyzed store and are suppressed at read time by policy evaluation. Under masking, a policy defect exposes data that is present. Under segregation, a policy defect exposes nothing, because the analytical store does not contain the data.

5. Cross-Domain Readiness Traversal

[0042] FIG. 5 shows readiness inference. A worker is identified by stable ledger identifier (510). A traversal (512) follows a chain of evidencing statements: worker to credential (514), credential to personnel qualification (516), qualification to governing procedure (518), procedure to active

permit to work (520). Each link is traversed not by reading a foreign key on a record but by locating the statement (s1 through s5) that evidences the link, each such statement carrying in its embedded semantic context the typed edge and that edge's confidence.

[0043] The credential record (514) carries a status and a visibility scope. It deliberately carries no per-credential confidence attribute. Confidence resides in the statements that evidence each link, not on the entity being linked. An aggregate authorization score (530) is computed from the confidences carried by the evidencing statements.

[0044] This placement is the substance of the mechanism. A confidence stored on a credential record asserts, without support, how certain the system is that a person holds that credential. A confidence carried by the statement evidencing the link records how the assertion arose, when, from what activity, and under what enrichment rules — and admits inspection of that provenance. Where four of five links are evidenced by high-confidence statements and the fifth by a low-confidence inference, the aggregate reflects the weak link and the traversal identifies which link is weak.

[0045] Where all required concepts are evidenced above threshold, the inference (532) proposes the qualification and routes it for human confirmation (534); the system does not self-authorize. Where they are not, a readiness gap output (536) enumerates the missing concepts and the links lacking sufficient evidence, converting an unmet requirement into a specific and actionable statement of what evidence is absent.

6. Competency Decay Scoring

[0046] FIG. 6 shows decay evaluation. Given a worker and a concept flagged in the governed ontology as safety-critical (610), the system locates three items of evidence by traversing embedded edges: the timestamp at which the governing credential was issued (612); the most recent statement evidencing a peer endorsement of the worker against that concept (614); and the most recent statement evidencing execution by the worker of a procedure governed by that concept (616).

[0047] Time deltas are computed for each (618) and combined into a predictive decay score (620). The three deltas measure different things and are not interchangeable. Credential age measures formal currency. Endorsement recency measures whether anyone competent has recently observed the worker and vouched. Execution recency measures whether the skill has recently been exercised at all. A worker whose credential is current but who has neither been observed nor performed the task in eighteen months is a materially different risk from a worker whose credential lapsed last week but who performs the task weekly under observation, and a scheme keyed on expiration date alone cannot distinguish them.

[0048] The score is emitted with the identifiers of the statements from which it was computed (622), so that a reviewer disputing a score inspects the evidence rather than the model. Where the

score exceeds a configured threshold (624) the concept is flagged for re-qualification (626), producing a work item routed to the qualification authority.

7. Two-Part Deterministic Query

[0049] FIG. 7 shows the query architecture. A natural-language request (710) is received. Part 1 (712) comprises a language model that produces a classification: an intent category; the collections implicated; the filters and constraints expressed in the request; and a description of the methodology by which the request would be answered. Part 1 is constrained not to emit a factual assertion responsive to the request. Its output is metadata about how to answer, never an answer.

[0050] Part 2 (714) selects a deterministic retrieval function from a maintained library, validates that an authenticated principal exists and terminates with an authorization error otherwise, constructs a data client initialized with that principal's permission context, executes retrieval under row-level access rules evaluated against the principal, and performs aggregation by fixed computation over retrieved values. Part 2 contains no language model. Given the same intent and constraints it returns the same result.

[0051] A sufficiency test (716) determines whether retrieved data answers the request. Where it does, the system composes a deterministic result (718) with provenance (720) comprising the identifiers of the statements from which the result derives and the concept mappings by which they were selected, and Part 1 renders the result in natural language. Where it does not, the system logs a structured gap (722) and returns a data-missing indication (724). Refraining from fabrication is not a behavior the model is instructed to exhibit; it is a property of the architecture, because the only fallback path available is the gap-logging path.

[0052] The gap record carries a proposed function name, a draft specification, a triage state, and the requesting role and department. Administrators triage the queue, implement the functions demand justifies (726), and re-run the originating request (728), which then resolves from retrieval. Capability expands along the axis of demonstrated demand and the boundary of what the system can answer factually is explicit rather than concealed.

[0053] It is not asserted that error becomes impossible. Retrieval may be executed against wrongly extracted constraints; a rule module may map a statement to an inapt concept. What the architecture provides is that the factual content of every response originates in retrieval rather than generation, and that provenance is emitted permitting identification of the step at which an error occurred. The failure surface is relocated to a layer that can be audited.

8. Procedure Drift Detection

[0054] FIG. 8 shows drift detection. A governed procedure and its step sequence are loaded from the governed ontology (810). Execution statements whose embedded edges evidence steps of that procedure are queried (812) and compared against the governed sequence (814).

[0055] Three conditions are evaluated. A step may be evidenced by fewer executions than a configured threshold count (816), indicating it is being skipped. Steps may be evidenced in an order inconsistent with the governed sequence (818). A step may be evidenced by no execution at all across a defined window (820), indicating omission.

[0056] A drift finding (822) is generated carrying the step identifier, a severity classification (824), and the identifiers of the statements from which drift was inferred (826), and is routed to the review authority designated for the procedure. Provenance matters here more than elsewhere, because a drift finding is an assertion about an operator's conduct and an operator is entitled to see the evidence.

[0057] The mechanism is bidirectional (828). Where many operators deviate consistently at the same step, the finding is treated as evidence that the authored procedure is defective — mis-sequenced, calling for unavailable tooling, or specifying a step that field conditions make impossible — rather than as evidence that many operators are non-compliant. The comparison is between recorded reality and authored intention, and either may be the thing that is wrong.

9. Tacit-Knowledge Capture

[0058] FIG. 9 shows knowledge elicitation. An actor is designated as a subject-matter expert (910) with recorded domains of expertise. A structured interview (912) is conducted according to a cognitive task analysis protocol, in one implementation the Critical Decision Method or the Critical Incident Technique, in which the expert is walked through a specific past decision and probed for the cues attended to, the discriminations drawn, the alternatives considered and rejected, the reasons for rejection, and the conditions under which the expert's ordinary approach fails.

[0059] Expertise elements are extracted (914). The extraction generates concept nodes in the governed ontology (916) representing the distinctions the expert draws, and graph edges (918) relating the expert to the concepts, procedures, and risk domains mastered. Nodes and edges are persisted (920) into the same ontology traversed by the readiness inference of FIG. 5 (922) and queried by the deterministic query system of FIG. 7 (924).

[0060] The result (926) is that expertise becomes queryable infrastructure rather than a transcript. A recorded interview or a written narrative is retrievable only by whoever knows it exists and searches for it. Expertise persisted as governed nodes and edges participates in readiness traversal, surfaces in response to natural-language queries that never mention the expert, and remains available when the expert has retired.

10. Ambiguity-Aware Classification

[0061] FIG. 10 shows the classification stage in detail. The classifier maintains a routing table in which each entry associates an intent with one or more match expressions, optional exclusion expressions, a resolution target, and a description. On receiving a normalized request the classifier evaluates every entry and retains as a candidate each entry whose match expression is satisfied and whose exclusion expressions are not, producing a candidate set (1010) rather than a first match. Each candidate is scored, weighted by the specificity of the satisfied expression and the extent of the request matched.

[0062] A threshold and margin decision (1012) is applied. The leading score must exceed a confidence floor, and must exceed the second-ranked score by a configured margin. Where either test fails, no retrieval function is executed and a clarification response (1014) is returned. No value is fabricated and no retrieval is attempted along a path the system cannot justify.

[0063] The clarification enumerates the candidates, each carrying a structured intent descriptor (1016) comprising a target label stating what the candidate resolves to and exclusion terms stating what it does not cover. Both halves are necessary. A label alone leaves an operator guessing at boundaries between similar-sounding options; exclusion terms make the boundary explicit, so the operator selects on the basis of what each option excludes as much as what it includes.

[0064] A selection is received (1018) and the retrieval stage is re-invoked with the selected intent (1020), bypassing re-classification. Bypass is required for correctness: re-classifying the original text would reproduce the same ambiguity and could resolve to a target the operator did not select.

[0065] Every response carries provenance (1022) comprising the target routed to, the reason for routing, the candidates considered, the routing confidence, and the structured intent descriptor. A regression corpus (1024) of representative requests paired with expected targets is evaluated against the routing table after any modification; divergence is reported as a regression, and where a first entry's expression captures a request expected to resolve to a second entry, a shadowing condition is identified specifically (1026).

[0066] A first-match-wins classifier is expressly disclaimed. Under first-match, a generic expression evaluated early shadows a specific expression evaluated later and the resulting misroute is silent — a confident, well-formed, entirely wrong answer, indistinguishable in the response from a correct one. The candidate-collection and margin architecture closes that failure mode by making the condition that produces it — two candidates of comparable score — the trigger for clarification rather than for arbitrary selection.

11. Multi-Class Actor Attribution

[0067] FIG. 11 shows the actor registry (1110). Six classes are admitted: human participant (1112); software inference agent (1114), identified by a machine address in a reserved agent namespace; automated system process (1116), identified by a process address; robotic platform (1118); botanical asset (1120); and zoological asset (1122). Activity from every class is persisted into a common ledger (1130) and every statement's actor is resolved, irrespective of class, by a single audited resolution function (1140).

[0068] Cross-class grouping (1150) is performed by semantic context, producing an attributable computed result (1160). Because class membership is a registry attribute rather than an architectural presupposition, a single query groups a robotic platform's completed inspection with a human technician's completed inspection of the same asset class, or correlates a monitored biological asset's recorded condition against the maintenance activity performed on the system sustaining it.

[0069] This generalization is the structural distinction over learner-centric activity-stream systems. Where the actor is presumed to be a person undergoing instruction, a robot's completed task, a monitored crop's growth interval, and a monitored animal's health observation are not representable as first-class activity, and reasoning across those classes and human activity in one operation is unavailable. Here the presupposition is removed at the registry, so the reasoning is available without special-case handling.

[0070] A statement produced by a software inference agent carries in its extension fields the confidence value that agent produced, and retrieval reports that confidence with any result derived from the statement. An inference is thereby traceable to the process that made it and qualified by that process's own stated certainty, on the same terms as any human-originated assertion.

12. Unified Media Architecture and Investigation Workflow

[0071] Media assets are cross-linked bidirectionally with each other and with operational records. A document record carries linking fields to an incident, a work order, a set of videos, a set of panoramic images, and a set of photographs; video, photograph, and panoramic-image records reciprocally carry document linking fields. Evidentiary assets additionally carry an evidence flag, a confidentiality classification, and a custody sequence recording every principal that captured, viewed, retrieved, or transmitted the asset with timestamp and action type.

[0072] Panoramic images carry an array of interactive markers, each specifying a marker type, angular position within the spherical projection expressed as yaw and pitch, title and body content, and a target identifier referencing a linked photograph, video, document, or adjacent panoramic image. An investigator constructs an immersive evidence package in which a reviewer stands virtually at a scene, looks in any direction, and reaches related evidence by selecting markers.

[0073] The full investigation workflow — incident registration, evidence capture with geolocation and condition classification, witness statement recording, selection among root-cause methodologies including five-whys, fishbone, fault tree, change analysis, Pareto, and barrier analysis, corrective and preventive action tracking through implementation and effectiveness verification, and formal handoff to training, change-management, and operations authorities with per-recipient completion tracking — is described in paragraphs [0015] through [0020] of the incorporated provisional application and is carried forward without modification.

13. Access Control, Agent Governance, and Audit

[0074] Row-level access rules are declared in each collection's schema as conditional expressions evaluated at query time against the requesting principal's role, application role, and record ownership. Enforcement occurs at the data layer, so a retrieval function holding service-role access nonetheless returns only records the invoking principal is entitled to see, preventing privilege escalation through function invocation.

[0075] Each inference agent is defined by a configuration enumerating exactly the collections it may access and, per collection, the operations permitted. An attempt to access a collection outside the enumeration, or to perform an operation outside the per-collection enumeration, is rejected. Where a first agent invokes a second, the second executes under the original principal's permission context, so agent composition cannot elevate privilege.

[0076] Every agent interaction is written to an audit record capturing the agent, the invoking principal and role, the request text, the classified intent, the collections accessed, the retrieval functions invoked, the count of records returned, the attempted-access and blocked indicators, the conversation identifier, the execution duration, success state and any error, the highest sensitivity classification of returned data, anomaly flags including cross-department access and sensitive-investigation access, and any peer agents invoked. Retrieval functions are prevented from initiating outbound requests to external services absent explicit principal authorization, so organizational data cannot be exfiltrated to external model providers or third-party services by agent action.

14. Standards Alignment, Governance, and Operational Intelligence

[0077] The platform implements learning-analytics metric families comprising outcome metrics measuring business impact, effectiveness metrics measuring learning achievement, and efficiency metrics measuring resource optimization, stored in a metric collection with a category, a stable metric key, a unit, a period, a plan flag distinguishing target from actual, and an owning role. Project governance implements stage-gated delivery with defined board roles, independent assurance roles, per-stage tolerances for cost, time, and scope, and manage-by-exception escalation when tolerances are breached. Risk management implements scored likelihood and

impact, response strategy classification, residual scoring after mitigation, and bidirectional linking between risks and captured lessons.

[0078] Real-time operational intelligence comprises geospatial visualization of personnel and asset positions derived from statement extension fields, one-action emergency beacon activation with responder notification, organization-wide shelter and evacuation alerting with per-person accountability confirmation, environmental and weather integration for work planning and post-incident analysis, and predictive maintenance alerting driven by sensor telemetry with predicted failure date and model confidence. These subsystems are described in paragraphs [0025] through [0031] and [0041] through [0046] of the incorporated provisional application and are carried forward without modification.

[0079] Figures 3, 4, 5, 6, 7, and 8 of the incorporated provisional application were annotated interface captures. They are not re-filed with the present application. Their subject matter remains disclosed by incorporation by reference and retains the October 24, 2025 date. Figure 1 of the incorporated provisional application is superseded by FIG. 7 of the present application as formal line art, substance unchanged.

15. Alternative Embodiments

[0080] An alternative embodiment implements native mobile applications providing offline-capable access to emergency beacon activation, work-order viewing, evidence capture, and message reading, with local persistence and queued writes reconciled on restoration of connectivity, biometric authentication, push notification, background position reporting under explicit consent and configurable interval, and machine-readable-code scanning for asset identification.

[0081] An alternative embodiment implements federated deployment in which multiple organizations operate independent instances with controlled cross-organization sharing for industry-wide safety benchmarking or shared instructional content. The privacy-segregated ledger of § 4 makes this practicable: because the analytical ledger contains no identifying data, cross-organization sharing of ledger records requires no anonymization step and no assurance that anonymization succeeded.

[0082] An alternative embodiment writes cryptographic hash fingerprints of statements and of critical record transitions to an append-only distributed ledger, storing the resulting transaction identifier on the statement, permitting a party to re-hash a statement and compare against the recorded hash to demonstrate the statement has not been altered since creation.

[0083] An alternative embodiment implements a marketplace in which organizations publish domain-specific agent configurations, retrieval functions, ontology extensions, and rule modules for installation by other deployments, with purchased components executing in isolated namespaces having access only to the installing organization's data.

16. Use Cases

[0084] Representative applications in manufacturing safety, healthcare clinical competency, military and defense training and readiness, oil and gas operations across distributed assets, educational institutions and accreditation, aviation maintenance and safety management, government and public-sector records administration, and higher-education research compliance are described in paragraphs [0073] through [0080] of the incorporated provisional application and are carried forward without modification.

[0085] One application illustrates the claimed mechanisms operating together. A manufacturer experiences repetitive strain injuries at a production line. Each occurrence is registered, evidence captured with geolocation and condition classification, and root cause analyzed. The enrichment engine of § 3 has already mapped every relevant statement — injury registrations, instructional completions, maintenance actions, behavioral observations — to concepts in the governed ontology at the moment each was written. A single deterministic retrieval per § 7 therefore correlates injury incidence against instructional completion, observed behavior, and equipment age without any post-hoc classification pass, and returns the correlation with the statement identifiers supporting it. The competency decay evaluation of § 6 identifies operators whose credentials are current but whose observed practice and execution recency indicate elevated risk. The drift detection of § 8 reveals that operators consistently omit a step of the authored procedure, and the bidirectional evaluation concludes the step specifies a tool the line does not have. The readiness traversal of § 5 identifies which operators lack sufficiently evidenced qualification for the revised procedure and enumerates the specific missing evidence. Throughout, no query result discloses any operator's identity, because per § 4 the analytical ledger contains none.

Claims

What is claimed is:

1.A method of persisting organizational activity, comprising:

(a) detecting an entity-change event arising from creation, modification, or deletion of a domain record, or from an activity report received from an external system;

(b) forming an experience statement identifying an actor, a verb expressed as a governed uniform resource identifier, and an object having an object type;

(c) applying deterministic auditable rules keyed on a pair comprising said governed verb uniform resource identifier and said object type, said rules producing identical output for identical input;

(d) resolving, by application of said rules, said pair to one or more concept uniform resource identifiers in a governed ontology and to one or more typed edges relating said statement to said concept uniform resource identifiers;

(e) computing a mapping confidence score for the resolution and an edge confidence score for each said typed edge, and recording a mapping source identifier identifying the rule producing the resolution;

(f) embedding into a semantic context structure of said experience statement, at the moment said experience statement is persisted, said concept uniform resource identifiers, said typed edges with their respective edge confidence scores, said mapping confidence score, said mapping source identifier, and a categorical domain label;

(g) representing said actor in said experience statement solely by a stable identifier bearing no correspondence to any natural-person attribute; and

(h) persisting said experience statement to a unified experience ledger containing no personally identifying information;

whereby said semantic context is machine-navigable at query time with no separate post-hoc enrichment pass, and said embedded semantic context is immutable against subsequent modification of said deterministic auditable rules.

2.The method of claim 1, wherein said unified experience ledger is persisted as native records of an operational data store such that a single query joins experience statements to operational records without extraction to a separate analytical store.

3.The method of claim 1, wherein said governed ontology comprises a concept scheme defining a namespace and governance authority, concepts each addressed by a uniform resource identifier, concept relationships expressing typed edges between concepts, and concept mappings associating external identifiers with governed concepts.

4.The method of claim 1, wherein a modification of said deterministic auditable rules is recorded with a version tag, and wherein experience statements persisted prior to said modification retain the semantic context embedded under the rules in force at their respective times of persistence, whereby a later read of such a statement encounters the same embedded semantic context irrespective of said modification.

5.The method of claim 1, wherein said categorical domain label is drawn from a controlled set comprising field operations, real-world event, knowledge management, incident investigation, and strategic planning.

6.The method of claim 1, further comprising storing in extension fields of a result structure of said experience statement at least one of a geospatial position with an accuracy value, an

identifier of an associated work record, an identifier of an associated document, a set of consumed materials, a labor duration, an environmental condition, and a confidence value produced by an inference process that produced said experience statement.

7.The method of claim 1, further comprising recording, for each retrieval executed against said unified experience ledger, an audit record identifying the invoking principal, the request text, the collections accessed, the retrieval functions invoked, a count of records returned, an attempted-access indicator, a blocked indicator, an execution duration, and a classification of the highest sensitivity of returned data.

8.The method of claim 1, wherein an evidentiary media asset associated with said experience statement carries a custody sequence recording each principal that captured, viewed, retrieved, or transmitted said asset, together with a timestamp and an action type for each such recording.

9.A method of responding to a natural-language request against an organizational data store, comprising:

(a) receiving said request at a classification stage comprising a language model;

(b) producing, by said classification stage, a classification comprising an intent category, an identification of implicated data collections, a set of constraints extracted from said request, and a description of a methodology for answering said request, said classification stage being constrained not to emit a factual assertion responsive to said request;

(c) selecting, on the basis of said classification, a deterministic retrieval function from a stored plurality of deterministic retrieval functions;

(d) authenticating a requesting principal and terminating without retrieval where authentication fails;

(e) executing said selected deterministic retrieval function against said organizational data store under row-level access rules evaluated against said authenticated requesting principal, and performing any aggregation by fixed computation over retrieved values, said selected deterministic retrieval function comprising no language model such that a given classification yields a given result;

(f) determining whether data sufficient to answer said request was retrieved;

(g) where said data is sufficient, composing a deterministic result and provenance comprising identifiers of experience statements from which said result derives and concept mappings by which said experience statements were selected, and rendering said result in natural language by said classification stage; and

(h) where said data is insufficient, recording a structured gap record and returning a data-missing indication, the method being structurally configured such that returning said data-missing indication is the sole available alternative to said deterministic result;

whereby factual content of a rendered response originates in step (e) and not in said language model.

10.The method of claim 9, wherein said structured gap record comprises a proposed function identifier, a draft specification for a deterministic retrieval function, a triage state, and a role and department of said requesting principal.

11.The method of claim 10, further comprising deploying a deterministic retrieval function derived from said proposed function identifier and said draft specification, and re-executing said request such that said request thereafter resolves under step (g).

12.The method of claim 9, wherein said deterministic retrieval function is executed by a runtime distinct from said language model and is prevented from initiating an outbound request to an external service absent explicit authorization by said requesting principal.

13.The method of claim 9, wherein an inference agent performing step (c) is restricted to an enumerated set of data collections and, for each said collection, an enumerated set of operations, and wherein an attempt by said inference agent to access a collection outside said enumerated set of collections or to perform an operation outside said enumerated set of operations for a collection is rejected.

14.The method of claim 13, wherein a first inference agent invokes a second inference agent, and said second inference agent executes under the permission context of said requesting principal, whereby invocation of said second inference agent does not elevate privilege.

15.The method of claim 9, wherein step (e) joins experience statements of a unified experience ledger with records of at least three of an instructional enrollment collection, an incident collection, an asset collection, and a behavioral observation collection, and returns a computed association together with a sample count.

16.A method of responding to a natural-language request against an organizational data store, comprising:

(a) receiving said request and normalizing said request;

(b) evaluating said normalized request against each of a plurality of stored routing entries, each said routing entry associating an intent with at least one match expression and with a resolution target, and collecting as a candidate every said routing entry satisfied by said normalized request, rather than terminating said evaluating upon a first satisfied routing entry;

(c) assigning to each said candidate a score weighted by a specificity of the satisfied match expression and by an extent of said normalized request matched thereby;

(d) interposing, prior to execution of any resolution target, an ambiguity evaluation comprising comparing a highest said score against a confidence threshold and comparing a difference between said highest score and a second-highest said score against a margin threshold;

(e) where either comparison of step (d) is not satisfied, returning a clarification response enumerating competing said candidates, executing no resolution target, and fabricating no value responsive to said request, each enumerated candidate being accompanied by a structured intent descriptor comprising a target label and one or more exclusion terms;

(f) receiving a selection of a single intent from among said enumerated candidates;

(g) re-invoking a deterministic retrieval stage with said selected intent, bypassing re-classification of said request; and

(h) returning a result together with machine-readable provenance identifying the resolution target routed to, a reason for said routing, the candidates considered, a routing confidence, and said structured intent descriptor;

whereby a silent-misroute failure mode, in which a match expression of one routing entry captures a request intended for another routing entry and a confident response is returned to a request not asked, is closed.

17.The method of claim 16, wherein where both comparisons of step (d) are satisfied, the method proceeds to execute the resolution target of the highest-scored candidate without returning said clarification response.

18.The method of claim 16, further comprising evaluating a stored corpus of requests paired with expected resolution targets against said plurality of stored routing entries, and reporting as a regression any request of said corpus whose resolved resolution target differs from its paired expected resolution target.

19.The method of claim 18, wherein said reporting identifies a shadowing condition in which a match expression of a first said routing entry captures a request paired with an expected resolution target of a second said routing entry.

20.The method of claim 16, wherein a said routing entry further comprises at least one exclusion expression, and wherein said routing entry is excluded from said candidate collecting where said exclusion expression is satisfied by said normalized request.

21.The method of claim 16, wherein said clarification response is recorded to an audit trail together with said candidate set and said scores, whereby a request resolved following clarification is distinguishable in said audit trail from a request resolved without clarification.

22.A system comprising one or more processors and storage holding instructions that, when executed, cause the system to:

(a) maintain an actor registry in which each registered actor is associated with an actor class selected from a plurality of actor classes comprising a human participant, a software inference agent, an automated system process, a robotic platform, a botanical asset, and a zoological asset;

(b) receive activity reports originating from actors of at least three distinct said actor classes and persist each said activity report as an experience statement in a common experience ledger, each said experience statement identifying its actor, a verb, an object, and a semantic context embedded at a time of persistence of said experience statement;

(c) resolve the actor of any said experience statement, irrespective of said actor class, by a single resolution function that records each resolution to an audit trail; and

(d) execute an analytical retrieval that groups experience statements across at least two distinct said actor classes by said embedded semantic context and returns a computed result attributable to the grouped experience statements.

23.The system of claim 22, wherein an activity report originating from a robotic platform comprises telemetry and a completed-task indicator, and wherein the experience statement formed therefrom is grouped under step (d) with experience statements of human participants performing a corresponding task.

24.The system of claim 22, wherein an experience statement originating from a software inference agent comprises, in an extension field of a result structure thereof, a confidence value produced by said software inference agent, and wherein said analytical retrieval reports said confidence value together with any result derived from said experience statement.

25.A system comprising one or more processors and storage holding instructions that, when executed, cause the system to:

(a) maintain a unified experience ledger storing experience statements, each said experience statement identifying an actor solely by a stable identifier bearing no correspondence to any natural-person attribute and comprising a semantic context embedded at a time of persistence, said unified experience ledger containing no personally identifying information;

(b) maintain an identity schema physically and logically separate from said unified experience ledger, said identity schema comprising an actor registry associating said stable identifier with personally identifying information;

(c) provide an audited resolution function constituting the sole computational path between said stable identifier and said personally identifying information;

(d) cause said audited resolution function, upon invocation, to verify a role authorization of an invoking principal for a requested class of personally identifying information, to write an access record to an access audit log identifying said invoking principal, said stable identifier, the fields requested, and an outcome, and to return said personally identifying information where said role authorization is verified or to deny and return null-valued fields where said role authorization is not verified; and

(e) execute analytical retrieval, export, and inter-organizational sharing operations exclusively against said unified experience ledger by said stable identifier;

whereby said operations of step (e) do not traverse said personally identifying information.

26.The system of claim 25, wherein where said audited resolution function is invoked by a software inference agent and said role authorization is not verified, said access record further comprises an attempted-access indicator and a blocked indicator.

27.The system of claim 25, wherein an obligation to erase personally identifying information of an actor is satisfied by operating upon said identity schema, whereby experience statements of said actor persist in said unified experience ledger in an unattributable state.

28.A method of evaluating currency of a competency, comprising:

(a) identifying a worker and a concept designated in a governed ontology as safety-critical;

(b) locating, by traversing typed edges embedded in semantic contexts of experience statements of a unified experience ledger, (i) a timestamp of issuance of a credential governing said concept, (ii) a most recent experience statement evidencing a peer endorsement of said worker with respect to said concept, and (iii) a most recent experience statement evidencing execution by said worker of a procedure governed by said concept;

(c) computing a respective time delta for each of (i), (ii), and (iii) of step (b);

(d) generating a predictive decay score as a function of said respective time deltas;

(e) outputting said predictive decay score together with identifiers of the experience statements from which said predictive decay score was computed, whereby evidence underlying said predictive decay score is traceable; and

(f) flagging said concept for re-qualification of said worker where said predictive decay score exceeds a configured threshold.

29.The method of claim 28, wherein said predictive decay score distinguishes a first worker having a current said credential and time deltas for (ii) and (iii) exceeding respective thresholds from a second worker having a lapsed said credential and time deltas for (ii) and (iii) below said respective thresholds.

30.The method of claim 28, wherein said flagging of step (f) generates a work item routed to a qualification authority designated for said concept.

31.A method of determining authorization readiness, comprising:

(a) identifying a worker by a stable identifier bearing no correspondence to any natural-person attribute;

(b) traversing a chain of evidencing experience statements from said worker, through a credential, through a personnel qualification, through a governing procedure, to an active permit to work, wherein each link of said chain is traversed by locating an experience statement evidencing said link, each said evidencing experience statement carrying in a semantic context embedded therein a typed edge corresponding to said link and an edge confidence for said typed edge;

(c) wherein said credential comprises a status and a visibility scope and comprises no per-credential confidence attribute;

(d) computing an aggregate authorization score from the edge confidences carried by said evidencing experience statements of step (b), rather than from any confidence attribute stored on said credential; and

(e) where all concepts required for said active permit to work are evidenced above a configured threshold, inferring said personnel qualification and routing said inferred personnel qualification for human confirmation.

32.The method of claim 31, wherein said aggregate authorization score reflects a lowest said edge confidence among said evidencing experience statements, and wherein said method further comprises identifying the link of said chain corresponding to said lowest edge confidence.

33.The method of claim 31, further comprising, where not all said concepts are evidenced above said configured threshold, outputting a readiness gap enumerating the concepts not so evidenced and the links of said chain lacking sufficient evidence.

34.The method of claim 31, wherein said inferring of step (e) does not confer said active permit to work absent said human confirmation.

35.A method of detecting drift between an authored procedure and its execution, comprising:

(a) loading a governed procedure and an ordered step sequence thereof from a governed ontology;

(b) querying experience statements of a unified experience ledger whose embedded typed edges evidence steps of said governed procedure;

(c) comparing said queried experience statements against said ordered step sequence;

(d) evaluating whether a step of said ordered step sequence is evidenced by fewer said experience statements than a configured threshold count, whether steps are evidenced in an order inconsistent with said ordered step sequence, or whether a step is evidenced by no said experience statement within a defined window; and

(e) generating a drift finding comprising an identifier of an affected step, a severity classification, and identifiers of the experience statements from which said drift finding was inferred, and routing said drift finding to a review authority designated for said governed procedure.

36.The method of claim 35, further comprising determining that a plurality of distinct workers deviated consistently at a same step of said ordered step sequence, and generating an indication that said governed procedure rather than the execution thereof is defective.

37.The method of claim 35, wherein a step of said ordered step sequence specifies required evidence and a visual success criterion, and wherein said evaluating of step (d) further comprises determining whether evidence captured for said step is inconsistent with said visual success criterion.

38.A method of capturing tacit expertise as queryable infrastructure, comprising:

(a) identifying an actor designated as a subject-matter expert and one or more domains of expertise recorded therefor;

(b) conducting a structured interview of said subject-matter expert according to a cognitive task analysis protocol, said structured interview eliciting cues attended to, discriminations drawn, alternatives considered and rejected, and conditions under which an ordinary approach of said subject-matter expert fails;

(c) extracting expertise elements from said structured interview;

(d) generating, from said expertise elements, concept nodes in a governed ontology;

(e) generating graph edges relating said subject-matter expert to said concept nodes and to procedures and risk domains mastered by said subject-matter expert;

(f) persisting said concept nodes and said graph edges into said governed ontology; and

(g) wherein said persisted concept nodes and graph edges are traversable by a cross-domain inference operation and queryable by a deterministic query system operating over said governed ontology;

whereby said expertise is persisted as queryable infrastructure rather than as an unstructured transcript.

39.The method of claim 38, wherein said governed ontology of step (f) is the same governed ontology against which deterministic auditable enrichment rules resolve concept identifiers at persistence of experience statements.

40.The method of claim 38, wherein said persisted concept nodes and graph edges are returned in response to a natural-language query that does not identify said subject-matter expert, and remain so returned after said subject-matter expert ceases to be an actor of the organization.

Abstract

Organizational activity across learning, operations, maintenance, safety, and management is persisted as experience statements in a unified ledger. On each entity-change event, deterministic rules keyed on a governed verb identifier and object type resolve the event to concepts in a governed ontology and embed the resulting semantic context, with per-edge confidences, into the statement at persistence time, so cross-domain grouping needs no post-hoc enrichment and embedded context is immutable against later rule changes. Actors may be human participants, software agents, automated processes, robotic platforms, or biological assets, all resolved by one audited function. Statements carry only non-identifying identifiers; personal information resides in a separate schema reachable solely through an audited, role-verified function. Natural-language access uses a classification stage emitting no factual assertion, a margin-gated ambiguity check, and a deterministic retrieval stage that returns a data-missing indication rather than a fabrication.

Inventor Information

Inventor 1.	Randy Stewart Miller
Assignee	Ripplemesh Corporation 304 Mica Trl, Maxwell, TX 78656
Email:	randystewartmiller@gmail.com
Attorney Docket:	10-03-2026
Priority Claim:	U.S. Provisional Application No. 63/904,593, filed 2025-10-24
Claim Count:	40 total, 9 independent

NO MORE FOLLOWS ———————————————————————————