Folding the People-Organization Layer Into Systems Engineering: Talents × High Reliability Organization Principles on a Foundation-Model Substrate

Alexander I. N. Derkatsch

Independent Researcher · bigBespoke LLC · Huntsville, Alabama, USA

ai.derkatsch@gmail.com

AI4SE & SE4AI Research and Application Workshop 2026 · Paper Presentation Track · Research Presentation

Primary Research Area: (AI4SE) Agentic AI and workflow integration

Secondary Research Area: (AI4SE) AI and cognitive assistants to support tasks from inference, decision-support and requirements trades to project management

Engineered systems emerge from the elegance of the people-organization substrate that produces them. The substrate has been parked outside the technical theory of systems engineering for sixty years, treated as a soft column of personalities, cognitive styles, sensemaking practices, and communication topologies, distinct from the engineering disciplines, despite recognition that any comprehensive theory of the field must include the study of human interactions that lead to the final result of the effort. The substrate developed below operationalizes that layer as the cross-product of the five principles of a high reliability organization and approximately thirty-five thinking talents drawn from cognitive-style research, yielding one hundred seventy-five named interaction primitives at the human-organization layer. The substrate is read not by asking contributors to declare their talents but by inferring them from natural behavior, an instrument that interprets organizational communication while inferring cognitive parameters from response patterns, now technically realizable through the present cohort of trillion-parameter pretrained transformer models wrapped in substantial engineering harnesses. Five demonstration systems exhibit distinct harness primitives the substrate composes: parallel-agent context management on a shared substrate; interface-construction-from-observation at any system boundary; schema-from-natural-interaction at usable cadence; longitudinal evidentiary substrate maintenance under multi-role contribution; and design-engine for deterministic-physics artifacts under simulator-verified acceptance. Composed, the primitives describe a collaborative-intelligence supersystem that observes organizational behavior, infers cognitive structure, drafts artifacts and recommendations, presents budget cases and schedules, and returns its outputs to the channels the organization already uses. The supersystem operates over multi-year programmatic cycles through a memory composed in three layers: current harness state, an evolving self-maintained ledger of the organization's substrate, and self-generated summaries that route back to the original sensor logs, meeting transcripts, design reviews, and decision artifacts when the substrate requires return to source. As the substrate operates, the organization's accumulated data fine-tunes capable open-source base models against the organization's specific work; the base model is fungible, the data is the asset, and the runtime compounds over years rather than degrading. The substrate opens the engineering organization to all available behavioral signal; the harness closes the loop on each interaction through the matrix's cell-activation read; the fine-tuning trajectory closes the loop on years of operation. Open system, closed loop: the structural form of the engineering organization that generates rather than consumes.

1. The gap at the human-organization layer

Complexity in aerospace system development is addressed through processes, architectures, and modelling techniques whose elaboration has occupied the field for sixty years. Model-Based Systems Engineering produced notation and tool support. Digital engineering and mission engineering produced process advances. The INCOSE Complexity Primer [1] acknowledges that the field has principles of complexity but lacks a theory of how an engineer comes to control it. Former NASA Administrator Michael Griffin examined this pattern and observed that failures characteristically emerge at interfaces between components thought to have no connection, with the consequence that finding them earlier in the design process requires holistic thinking rather than additional decomposition [2]. Modern systems engineering, on Griffin's account, is intrinsically a team effort whose final artifact reflects the organizational structures that produced it: any comprehensive theory of systems engineering must include the study of human interactions which lead to the final result of that effort.

Griffin's proposition has not been operationalized. The reason is structural: the human-organization layer of systems engineering (personalities, cognitive styles, organizational behaviors, sensemaking practices, communication topologies) has been parked outside the technical theory's scope, treated as a soft column distinct from the engineering disciplines. High Reliability Organization theory, developed by Weick and Sutcliffe and the broader Berkeley HRO program [3], drew organizational behavior closer to the engineering disciplines by formalizing the five behavioral principles distinguishing high-reliability operation, but the framework is descriptive of behavior rather than prescriptive of control. Two complementary directions have developed structural inputs toward the missing control layer: quantifying communication across the organizational hierarchy under HRO adherence to relate communication directly to the impact of unexpected events [4], and coupling Ancona's five-step sensemaking practice [5] to the thinking-talent taxonomy of Markova and McArthur [6] to produce a team-composition strategy across the four cognitive quadrants [7]. Both require a mediating instrument that has only recently become technically realizable.

2. The 175-cell substrate

The composition of the two directions at the human-organization layer is a matrix at the cross-product of the five HRO principles and approximately thirty-five Markova thinking talents:

$$\mathrm{M} = \mathrm{HRO} \times \mathrm{T}, \quad |\mathrm{M}| = 5 \times 35 = 175 \text{ interaction primitives.}$$

Each cell M[h, t] names the interaction pattern between HRO principle h and thinking talent t. The cell *Wanting-to-Win* × *Reluctance-to-Simplify* denotes the operational pattern in which a contributor whose dominant talent is wanting-to-win refuses premature closure on problems framed as competitive, but only on those framed that way. The cell *Sensitivity-to-Operations* × *Connection* denotes the pattern in which a contributor whose dominant talent is connection notices interaction-level anomalies between subsystems but may overlook intra-subsystem operational drift. Each of the 175 cells names a distinct interaction primitive at the human-organization layer, recognizable in operational practice to anyone who has worked in an engineering organization long enough to observe how talent-and-principle pairs interact.

The cardinality is the central structural claim. Complex systems are controlled by a small number of underlying parameters: typically two or three, rarely six or seven; beyond approximately ten the system is no longer complex in the controllable sense but chaotic [8]. At the human-organization layer, talents and HRO principles are the two controlling parameters. Personality is not at this layer; it is a state-driven emergent surface, unstable across context, and not a design surface against which composition decisions can be made.

Stability is at the substrate, not at the behavior. A contributor whose dominant talent is Love-of-Learning and one whose dominant talent is Wanting-to-Win may both arrive at risk-seeking behavior in the same situation but for distinct and stably-predictable reasons: the first energized by the prospect of novel learning, the second by competitive advance. The substrate captures the stable reasons; the behavior is the contingent surface. The matrix is therefore not a human-resources instrument. It is the upstream control parameter for whatever the organization intends to build, the substrate whose elegant composition yields elegant emergent systems regardless of target complexity.

3. Reading the substrate from behavior

The substrate is operationally usable because it is read from observed behavior rather than from explicit assessment. An organization deploying the framework does not pause for contributors to introspect on their talents or for the program to administer questionnaires. The matrix is the schema against which the harness layer's foundation model interprets the natural artifacts the organization already produces: meeting transcripts, design-review minutes, requirements records, code commits, change requests, and the contributor's own response patterns within those artifacts.

The mechanism is exhibited by the CopApp.ai demonstration described in §4. A three-agent architecture (a Psyche agent maintaining a working hypothesis about the user's top-five talent profile, a World agent maintaining scenario state at the granularity Psyche's analysis specifies, and a Dialog agent producing the response under Psyche-and-World direction) refines the talent hypothesis through subtle probing of how the user responds. The non-player character may phrase a line as "I just want everyone to make it out okay" to test whether the user's reply pattern aligns with a Wanting-to-Win signature; alternatively as "I'm not sure I'm going to be okay tonight" to probe for a Mentor pattern; or as "Just tell me what's going on" to probe for Thinking-Logically. Within a few turns the hypothesis stabilizes, and subsequent interaction is parameterized by the inferred profile.

The behavior-focused reading is what makes the framework deployable. The substrate accumulates from the work the organization already does. Contributors continue their tasks; the harness observes; the matrix's cells activate against the artifacts. No introspection step is required. This is the architectural feature that distinguishes the framework from psychometric-instrument deployments, and it is the feature on which the framework's practical viability rests.

4. The runtime and the demonstration composition

The mediating instrument the matrix requires has become technically realizable through trillion-parameter pretrained transformer models (open-weight DeepSeek-V3 reporting 671 billion total parameters with 37 billion active under Mixture-of-Experts routing [9], Meta Llama 4, Mixtral 8×22B, and the closed-weight frontier of Anthropic Claude Opus, xAI Grok, Google Gemini, and OpenAI GPT-class models at vendor-claimed comparable scale [10]) wrapped in substantial engineering harnesses. The harness is the engineering substrate around a foundation model that gives the model operational reach into a specific deployment context: file-system access, tool-use scaffolding, context management, code-execution sandboxes, retrieval over the working repository, monitoring and retry logic, conversation memory. Cursor and Claude Code at production scale exemplify the pattern. Substrate-construction work (designing, populating, and maintaining the product-system's knowledge graph from the organization's data sources) has crossed the threshold of feasibility through recent LLM-driven knowledge-graph construction methods [11], operating over standardized agent-tool protocols including the Model Context Protocol [12] and Agent-to-Agent Protocol.

Five demonstration systems exhibit distinct harness primitives the substrate composes. **T-MINUS** (eight LLM console operators under a Flight Director command loop on a fourth-order Runge-Kutta physics simulator) demonstrates parallel-agent context management on a shared substrate. **Universal Interface** (observing a new pairwise integration boundary, generating a connector implementation, synthesizing adversarial tests, iterating to convergence) demonstrates interface-construction-from-observation; the harness functions as the Interface Control Document, with the property that the ICD updates in flight when the underlying systems change. **CopApp.ai** demonstrates schema-from-natural-interaction as described in §3. **PawTrek** (voice-first capture of K9-handler reports structured against court-admissibility precedent under *Florida v. Harris* [13]) demonstrates active-interlocutor longitudinal substrate maintenance. **FletchGNC** (a foundation model proposing control-law modifications gated by simulator-verified acceptance criteria) demonstrates design-engine for deterministic-physics artifacts with prose-and-code co-evolution.

The demonstrations are pieces. Composed, they describe a collaborative-intelligence system that captures organizational behavior through PawTrek-style natural-interaction interfaces, infers cognitive structure through CopApp-style multi-agent decomposition, coordinates across specialized roles through T-MINUS-style monitor-gated multi-agent operation, integrates and decides across heterogeneous systems through Universal Interface-style harness self-modification, verifies command-loop integrity through FletchGNC-style simulator-gated proposal-acceptance, and returns its outputs to the organization's existing communication tools, drafting meeting recommendations, product notes, presentations, budget cases, and work schedules. The composition is not yet built as a single deployed system. Each primitive is exhibited in its own deployment context. The composition is the architectural target the substrate is structured to admit, and the demonstrations are evidence that each piece is realizable today.

5. Pattern language and the closed feedback loop

The substrate is, in the system-theoretic sense, a pattern language. Product-systems have patterns: components, subsystems, environments, the failure modes those produce. People have patterns: the talents that determine where each contributor draws energy and where each contributor repels. Lifecycle phases have patterns: design, integration, qualification, sustainment, each with its own substrate-activation signature. Pattern enumeration belongs ideally to Concept-of-Operations and requirements-derivation work; most programs do not perform it, and the omission is a major source of fault discovery late in the cycle when correction is expensive. The matrix names the patterns explicitly and at the cardinality the substrate's control requires, which is what makes the substrate deployable at any point in the lifecycle rather than only at conception.

Engineering organizations typically operate as closed systems with open loops. Inputs arrive locked down: hostile contracts, fixed requirements, schedule pressure. Outputs depart: mediocre product on firm-fixed-price, or never-ending engagement on cost-plus. The loop never closes on feedback that would change the system; the system never opens to fresh signal that would change the loop. The result is entropic. The framework inverts the pattern. The substrate opens the system to all available behavioral signal: contributors, instruments, communication, telemetry. The harness closes the loop on each interaction through the matrix's cell-activation read and the substrate's continuous maintenance. The result is generative rather than entropic: an organizational structure that learns from its own work as the work proceeds and steers toward the system it is trying to become. The path is from quantifying communication in high-reliability organizations to orchestrating it through an AI substrate whose interpretations are traceable to structural axes rather than to model intuition alone.

References

1. Sheard, S., et al., *A Complexity Primer for Systems Engineers*, Complex Systems Working Group, International Council on Systems Engineering, Wilmington, DE, 2015.
2. Griffin, M. D., "How Do We Fix Systems Engineering?" Keynote Address, 61st International Astronautical Congress, Prague, Czech Republic, Sept.–Oct. 2010.
3. Weick, K. E., and Sutcliffe, K. M., *Managing the Unexpected: Resilient Performance in an Age of Uncertainty*, 2nd ed., Jossey-Bass, San Francisco, CA, 2007.
4. Derkatsch, A. I. N., Maa, M. E. I., and DeTurris, D. J., "Managing Complexity through Quantifying Communication in High Reliability Organizations," AIAA Paper 2022-0994, Jan. 2022. https://doi.org/10.2514/6.2022-0994
5. Ancona, D., "Sensemaking: Framing and Acting in the Unknown," *The Handbook for Teaching Leadership*, edited by S. A. Snook, N. Nohria, and R. Khurana, SAGE Publications, Thousand Oaks, CA, 2012, pp. 3–19.
6. Markova, D., and McArthur, A., *Collaborative Intelligence: Thinking with People Who Think Differently*, Spiegel & Grau, New York, 2015.
7. Maa, M. E. I., Derkatsch, A. I. N., and DeTurris, D. J., "Managing Complexity through Collaborative Intelligence," 31st INCOSE International Symposium, July 2021.
8. Ashby, W. R., *An Introduction to Cybernetics*, Chapman & Hall, London, 1956.
9. DeepSeek-AI, "DeepSeek-V3 Technical Report," arXiv preprint, arXiv:2412.19437, Dec. 2024. https://doi.org/10.48550/arXiv.2412.19437
10. Bommasani, R., et al., "On the Opportunities and Risks of Foundation Models," Stanford Center for Research on Foundation Models, Stanford University, arXiv:2108.07258, Aug. 2021.
11. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., and Wu, X., "Unifying Large Language Models and Knowledge Graphs: A Roadmap," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 36, No. 7, July 2024, pp. 3580–3599. https://doi.org/10.1109/TKDE.2024.3352100
12. Anthropic, "Model Context Protocol Specification," Anthropic, Version 2024-11-05, Nov. 2024. URL: https://modelcontextprotocol.io [retrieved 5 June 2026].
13. *Florida v. Harris*, 568 U.S. 237 (2013).