Human–AI Interaction in the Enterprise

Naji Zouiti
ChatGPT (Prism assistant)

Abstract

This document frames enterprise human–AI interaction (H–AI) as an information architecture problem: how organizations capture, govern, and operationalize the traces produced when humans and AI collaborate. It argues that *decision traces* (rationales, constraints, approvals, exceptions, and iterations) are the key bottleneck, and the key input, for evolving a knowledge graph into a context graph.

Building on a taxonomy of Observation/Interaction as Context (OaC, IaC, OtIaC, ItOaC), the text introduces Very Fast Interaction (VFI) and Very Slow Interaction (VSI) as complementary regimes that stress context and memory in different ways. It also proposes the Decision Parallelization Pipeline (DPP), where AI turns decision-making from a linear funnel into parallel lanes that produce options, checks, summaries, and simulations before human arbitration.

Finally, the document connects these architectural patterns to practical enterprise constraints: API economics, staged data governance under urgency (especially for small and medium scale), and charter files inspired by "Claude's Constitution" that encode values while improving observability of real-world "dark arts" constraints. An AI management framework is presented as the control layer that unifies governance, cost, memory, and execution into a sustainable path from knowledge to context.

1 Subject

Enterprise human–AI interaction is not only an interface problem; it is an information architecture problem. As organizations embed assistants and agentic systems into everyday workflows (chat, tickets, documents, approvals), the enterprise continuously generates human decision traces: rationales, constraints, exceptions, approvals, and escalation paths. Treating these traces as structured context is what enables a shift from static knowledge representations to operational context graphs.

2 From Ontologies and Knowledge Graphs to Context Graphs

The business unit that links the ontology layer to the context graph layer can be expanded to include selected variables from the external environment (or a framework of environments). In enterprise settings, the "environment" effectively becomes part of the business across both layers because it is continuously interpreted through human judgment—and increasingly through human–AI interaction that shapes how information is framed, escalated, approved, and acted upon.

As information architecture evolves vertically, the enterprise distinguishes itself from its environment through a flow of decision traces. These traces are shaped by the human factor and, by extension, by the organization's patterns of human–AI interaction (e.g., assistants embedded in collaboration tools, copilots in workflow systems, and agentic services). In this view, the business unit becomes narrower and more intentional: it is defined by a semantic backbone (ontologies + knowledge graphs) running in parallel with internal decision-making processes, where AI augments how decisions are formed, documented, and revisited.

3 Decision Traces as the Bottleneck

Evolving a knowledge graph into a context graph requires addressing the bottleneck of human decision traces. One way to model these traces as context is to explicitly integrate human–AI interaction into the decision loop (for example, an assistant used inside Slack threads to summarize, propose options, surface precedent, or capture rationales). Building a context management framework is the first step toward a context graph: at the foundational level, AI may begin in an observer role—capturing and structuring context—before it becomes an interactive participant.

3.1 A Taxonomy of Human–AI Activity as Context

A practical taxonomy for turning human–AI activity into enterprise context aligns with the observation-to-interaction spectrum:

- **Observation as Context (OaC):** AI observes only (capture/label/index decision traces).
- **Interaction as Context (IaC):** AI interacts directly (recommend/ask/mediate), and the interaction itself becomes context.
- **Observe-then-Interact as Context (OtIaC):** AI first observes to build situational context, then interacts.
- **Interact-then-Observe as Context (ItOaC):** AI initiates interaction to elicit missing context, then observes outcomes.

Each mode implies a distinct pipeline and governance model for decision traces, centered on a Context Management Framework that routes traces into a Context Graph—moving from Knowledge Graph → Context Graph by making enterprise decisions (and human–AI collaboration around them) first-class, queryable context.

3.2 Decision Parallelization Pipeline (DPP)

The taxonomy (OaC/IaC/OtIaC/ItOaC) implies a structural shift in how enterprise decisions are produced. Instead of a linear funnel where humans sequentially transform data into execution, AI-enabled systems can parallelize decision work: generating options, stress-testing assumptions, and pre-assembling context while humans arbitrate. We refer to this as the *Decision Parallelization Pipeline (DPP).*

From linear to parallel decision-making. In the traditional pipeline, decision effort is largely serialized:

Linear: Data → Projects → Decisions → Execution → Results → Feedback

Under DPP, AI creates parallel "lanes" that compress time-to-decision by working ahead of the human decision point and by keeping context synchronized:

Parallel (DPP): Data → Projects → {Option generation ‖ Risk/constraint checks ‖ Retrieval/summarization ‖ Simulation} → Human arbitration/approval → Execution → Results → Feedback

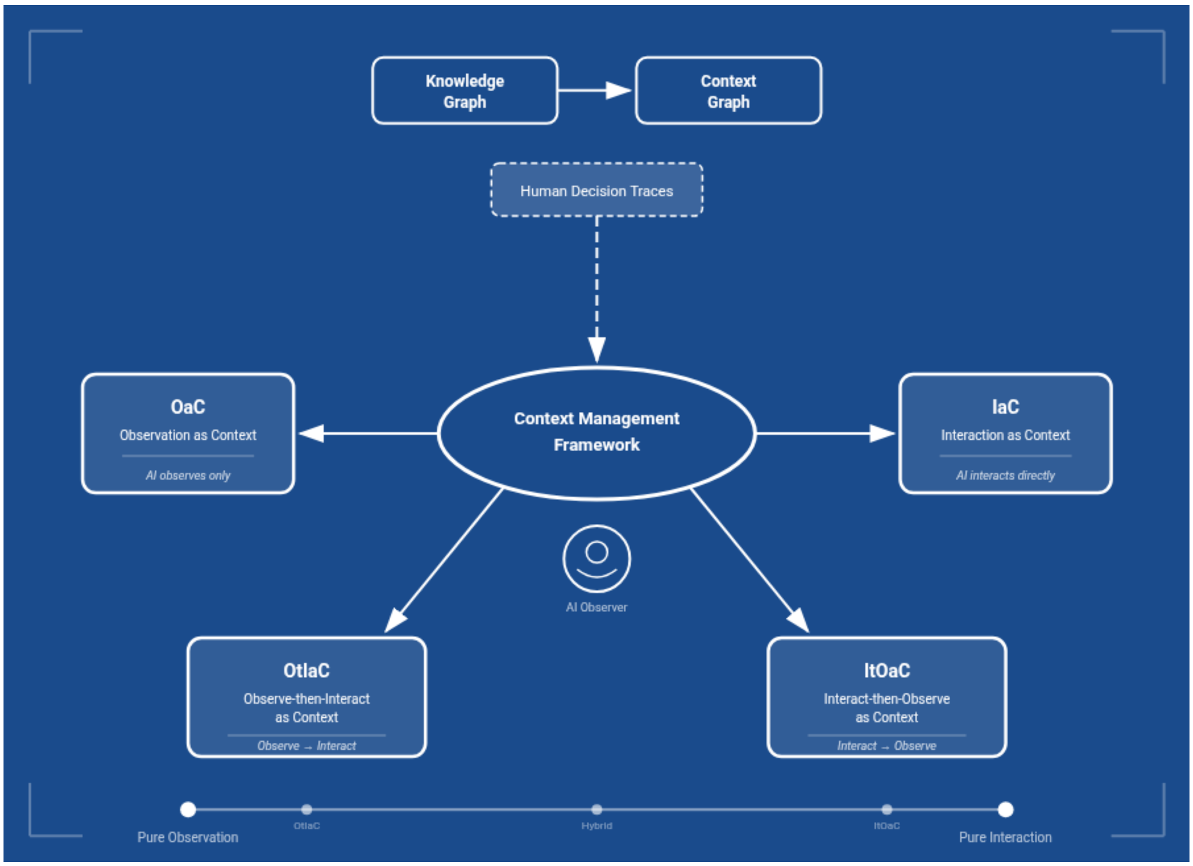

Figure 1: Taxonomy context: moving from knowledge graph to context graph via a context management framework and human decision traces, with OaC/IaC/OtIaC/ItOaC modes along the observation–interaction spectrum.

How the taxonomy produces DPP. DPP can be understood as an orchestration of the taxonomy modes across the pipeline:

- **OaC** continuously captures decision traces and normalizes them into reusable context (low disruption, high auditability).
- **OtIaC** uses fast observation to assemble minimal viable context, then interacts to converge on a decision with fewer iterations.
- **IaC** enables high-throughput parallel drafting (options, messages, plans), with governance required because interactions become part of context.
- **ItOaC** front-loads clarification (targeted questions) to unlock parallel work when critical context is missing.

The practical implication is that "Decisions" become less of a single stage and more of a *parallelizable workload* whose artifacts (alternatives, rationales, rejected paths) should be stored as first-class context.

3.3 Very Fast Interaction (VFI) in the Enterprise

Very Fast Interaction (VFI) describes human–AI exchanges that occur at near-conversational latency and high frequency within work systems (chat, IDEs, service desks, review tools). VFI changes what "context" means operationally: instead of being assembled episodically, context is continuously updated by micro-decisions, short clarifications, and rapid iterations. In practice, VFI increases both the volume and the fragmentation of decision traces, which elevates the importance of capture, normalization, and governance.

Context and memory implications. VFI pushes the enterprise toward a hybrid memory model:

- **Ephemeral working context** (seconds to hours): the immediate conversational state that enables fast turn-taking.
- **Session memory** (hours to days): the compressed record of what was tried, accepted, rejected, and why.
- **Durable organizational memory** (weeks to years): versioned, permissioned traces (decisions, exceptions, approvals) anchored to systems of record.

As VFI accelerates, the risk is that critical rationale remains trapped in transient channels. A context management framework therefore needs explicit mechanisms for summarization, attribution, and retrieval ("what decision was made, by whom, under which constraints, with which AI suggestions?").

Relationship to the taxonomy. VFI does not replace OaC/IaC/OtIaC/ItOaC; it intensifies the trade-offs between them:

- **OaC** supports VFI safely by observing and structuring traces without perturbing the workflow, but may lag when users need rapid guidance.

- **IaC** maximizes speed and productivity, but increases the need for guardrails (permissions, policy, auditability) because the interaction itself becomes part of the context graph.
- **OtIaC** is a natural fit for VFI: observe quickly to assemble minimal viable context, then interact to reduce ambiguity and converge on a decision.
- **ItOaC** is useful when VFI reveals missing context: the AI can ask targeted questions first, then observe outcomes to stabilize memory.

In all cases, VFI implies that context and memory must be treated as products of collaboration, not just repositories of facts.

3.4 Very Slow Interaction (VSI) in the Enterprise

Very Slow Interaction (VSI) covers human–AI collaboration patterns that are intentionally long-horizon: planning multiple decisions ahead, coordinating multi-step work, or completing complex tasks whose runtime exceeds a typical work shift. Where VFI emphasizes rapid micro-turns, VSI emphasizes continuity, handoffs, and the ability to maintain a stable objective over extended time windows.

Context and memory implications. VSI is primarily a memory and governance problem:

- **Long-lived objectives:** tasks need persistent goals, constraints, and success criteria that survive breaks, handoffs, and tool failures.
- **Structured intermediate artifacts:** plans, checklists, decisions, and partial results must be stored as durable context (not only as chat history).
- **Agent coordination:** long tasks may require multiple specialized agents (parallel lanes) whose outputs must be reconciled through a decision trace.

Relationship to DPP and the taxonomy. VSI is often the "macro" counterpart of DPP: instead of parallelizing one decision, the system parallelizes a *sequence* of decisions and deliverables. In practice, VSI tends to favor OaC and OtIaC to keep costs and risk predictable (observe and stabilize context before interacting), while IaC/ItOaC are used selectively to unblock missing context or accelerate critical path steps.

Economic implications. Because VSI can involve long runtimes, repeated tool calls, and multi-agent loops, it must be explicitly budgeted. This makes caching, summarization checkpoints, and clear stop conditions essential—otherwise "slow" work silently becomes continuous spend.

3.5 API Key Economics and Execution Costs

In enterprise deployments, human–AI interaction is constrained not only by governance and context quality, but also by economics. As VFI increases the frequency of calls, per-request costs accumulate quickly, and the "context management framework" must include a cost model alongside the memory model.

1. Financial planning for API costs. A practical plan estimates costs from expected call volume and token usage, explicitly separating cache hits from cache misses (and their different cost profiles when applicable). It also anticipates price changes due to model upgrades, defines lower-cost fallback options, and inventories current financial resources.

2. Automation worthiness vs. cost and effort. Not every task benefits from AI automation. Low-value, high-frequency requests (e.g., "what's the weather?" or generic lookups) can silently drain budgets over time; cheaper, specialized solutions should be evaluated first. This trade-off becomes sharper under VFI, where convenience can translate into continuous spend.

3. Hardware upgrades and VPS usage. Compute is part of the same economic equation. Whether running models locally or relying on a VPS, upgrades should be justified by a use case's value-creation strategy, predicted added value, and long-term ROI. Under VFI, excess capacity should be intentionally exploited (batching, caching, scheduled heavy jobs) rather than left as idle cost.

4. R&D discipline. Experiments (e.g., with *Clawdbot*) should follow an explicit plan: objectives, hypotheses, expected results, and a clear "value tipping point" from experimental to executable. This prevents runaway iteration costs—especially when fast feedback loops make experimentation feel "free".

5. Execution strategy and observability. Operational agents (e.g., *Clawdbot* and Claude Code) should run with defined efficiency targets: estimated daily runtime, parallel agents, loop triggers (e.g., "Ralph Wiggen"-style routines), and observability to learn from outcomes. In the taxonomy, higher-interaction modes (IaC, ItOaC) typically increase cost variability, while observation-heavy modes (OaC, OtIaC) tend to be more predictable due to stronger opportunities for caching and summarization.

3.6 Frugal Human–AI Adoption Beyond "Digital Transformation"

The term "digital transformation" is often used as a catch-all narrative for enterprise change, sometimes in ways that obscure practical alternatives. In the context of human–AI interaction, a frugal approach treats "transformation" as an outcome of measurable value creation rather than as a prolonged program.

Transformation vs. solutions that already exist. Transformation implies that existing processes must change over time, which can unintentionally frame problems as unsolvable without organizational disruption. In practice, many solutions already exist but are overlooked in the name of long-term "strategic AI adoption". A frugal HAII strategy starts by identifying what can be achieved with minimal change: better retrieval, clearer context capture, and selective automation aligned with the API cost model.

Radical, time-boxed change—or no change at all. If a transformation is required, it should be time-boxed (weeks, not quarters), integrated with urgency, and evaluated by near-term results (cycle time reduction, fewer iterations, fewer errors, higher decision quality). Only after initial wins should the organization iterate through feedback, updates, productivity improvements, and value-maximization. This avoids multi-month initiatives that accumulate API spend and coordination overhead without producing durable context.

Horizontal first, then vertical. With digitalization, it is often better to solve problems horizontally before going vertical. "Vertical AI" should first integrate into the existing horizontal environment (chat, documents, tickets, approvals) through OaC/OtIaC patterns: observe, capture traces, and stabilize context and memory. As interoperability and autonomy mature, the system can transition to more vertical progression at the managerial level (DPP orchestration, governance, and context graph alignment).

In short, frugality is not a constraint on human–AI interaction; it is an architectural principle that keeps VFI and DPP sustainable by linking context, memory, and automation choices to measurable outcomes and controllable costs.

3.7 "Claude's Constitution" and Enterprise Markdown Charters

The "Claude's Constitution" blog post provides a useful framing: an AI system can be guided by an explicit, written constitution rather than by implicit expectations. In enterprise human–AI interaction, the analogous pattern is a set of local markdown "charter" files (policy-as-text) that can be reviewed, versioned, diffed, and audited.

Values and ethics as first-class context. A charter file should document enterprise values, ethics, escalation rules, and decision authority. In practice, this can be maintained as a local markdown file (e.g., `enterprise-charter.md`) that is versioned, reviewed, and treated as policy-as-text. In the taxonomy, it becomes operational guidance for OaC/IaC/OtIaC/ItOaC: what can be automated, what requires human approval, and what must be recorded as a decision trace.

The "dark arts" as explicit context, not implicit folklore. Enterprise reality includes informal or "shady" practices that materially shape outcomes (shadow workflows, policy exceptions, KPI gaming, undocumented dependencies). A governed model should not endorse wrongdoing; however, it should be able to *recognize and label* these patterns as context signals. To make this actionable, an enterprise can maintain a second markdown file (e.g., `dark-arts.md`) that documents these behaviors and their underlying logic as "necessary evils" tied to survival, expansion, and profit (while still tagging them as risk). This reduces ambiguity and improves observability: it clarifies why DPP lanes diverge, why constraints are inconsistent, and where risk accumulates.

Practical implication for the context graph. Together, `enterprise-charter.md` (values) and `dark-arts.md` (real constraints) should drive structured tagging of traces (value constraints, policy exceptions, and risk signals). This helps the context graph preserve not only *what* was decided, but also *which norms were intended* and *which realities were encountered*—especially under VFI, where micro-interactions can otherwise turn into ungoverned, high-frequency spend.

3.8 AI Management Framework (Governance, Cost, and Context)

To make the full system operational, the enterprise needs an *AI management framework* that unifies governance, economics, and context engineering. In this context, Naji Zouiti defines *AI management* as "the management of artificial resources that are characterized by human like intelligence through reasoning, context and memory to align with the values of an enterprise or industry". It is the control layer where the taxonomy, VFI, DPP, cost management, and constitution/charters converge into a single set of decisions about *how* AI is allowed to observe, interact, execute, and remember.

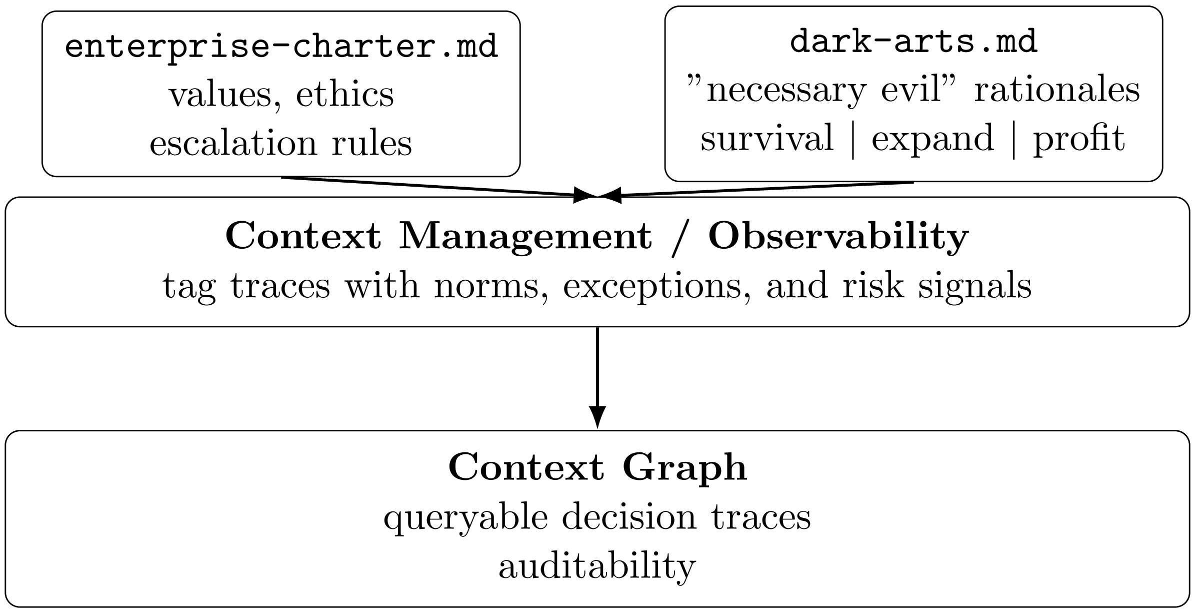

Figure 2: Two markdown inputs for governed context: a values charter and a "dark arts" file that makes real constraints explicit. Both feed observability and trace tagging, so context graphs capture norms, exceptions, and risk—not only facts.

A practical operating model. At minimum, the framework defines:

- **Interaction modes and routing:** when to use OaC vs. IaC vs. OtIaC vs. ItOaC, and when to escalate to humans.
- **Memory policy:** what remains ephemeral, what becomes session memory, and what becomes durable organizational memory.
- **Observability:** how decision traces are logged, summarized, attributed, and made queryable in the context graph.
- **Data governance (staged):** how data is accessed, scoped, and permissioned; which sources are trusted; and what becomes durable memory.
- **Economics:** budgets, cache strategy, fallback models/tools, and ROI constraints for automation.
- **Execution control:** when to trigger parallel lanes (DPP), run agents/loops, and measure outcomes.
- **Values and risk:** charter files inspired by "Claude's Constitution" that encode norms while recognizing real-world "dark arts" as context signals.

A note on governance vs. urgency (SMB/medium scale). At small and medium scale, governance is often not the limiting factor in early AI adoption; urgency and iteration speed are. A frugal strategy is to start with minimal, high-signal guardrails (least privilege, logging, and clear boundaries on durable memory), then evolve toward stronger data governance as VFI/DPP usage grows and the context graph becomes more operational.

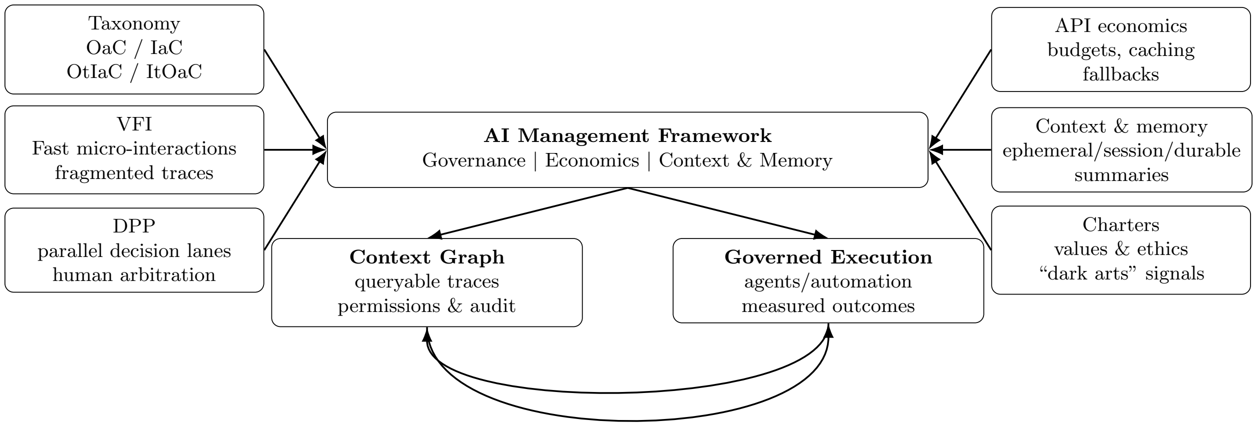

Figure 3: An AI management framework as the enterprise control layer: it unifies interaction modes (taxonomy), speed (VFI), parallel decision-making (DPP), memory/observability, economics, and charter-driven governance into governed execution and a queryable context graph.

4 Conclusion

This document argued that enterprise human–AI interaction should be treated as a context-and-memory engineering discipline, not only as a UX layer. By centering decision traces, the taxonomy (OaC/IaC/OtIaC/ItOaC), interaction regimes (VFI/VSI), and the Decision Parallelization Pipeline (DPP) become practical levers for building a context graph that is both operational and auditable. The AI management framework then serves as the control layer that keeps speed, cost (API economics), values (charters), and staged governance aligned with measurable outcomes.

5 Discussion: Toward AI as Its Own Entity

A broader question remains open: what are we actually building toward? Yann LeCun has argued that large language models are not, by themselves, a sufficient path to artificial superintelligence, and that a humanoid future is not guaranteed by scaling alone. Hardware limitations also remind us that machine intelligence is finite and bounded by energy, cost, latency, and physical constraints.

One interpretation of these limits is philosophical: AI has no soul and will never have one; in that sense it may be best understood as a historical instance—a powerful but passing invention tied to our civilization. This framing can be sobering and clarifying: it pushes enterprise practice away from anthropomorphic expectations and back toward rigorous context, memory, economics, and accountability.

An alternative direction is the *institutionalization* of AI. Rather than designing AI as an imitation of humans, we can model it as an imitation of institutions—bounded actors with roles, permissions, obligations, and audit trails, operating alongside other legal entities. In this view, "AI as its own entity" becomes an enterprise design concept: a governed agent with clearly defined authority, durable identity, and traceable responsibility, integrated through the context graph and constrained by charters.

As for what this creation is ultimately allowed to become, and for how long, that question exceeds engineering. Recognizing finiteness, technical, institutional, and spiritual, can be a source of reassurance as much as it is a constraint.