

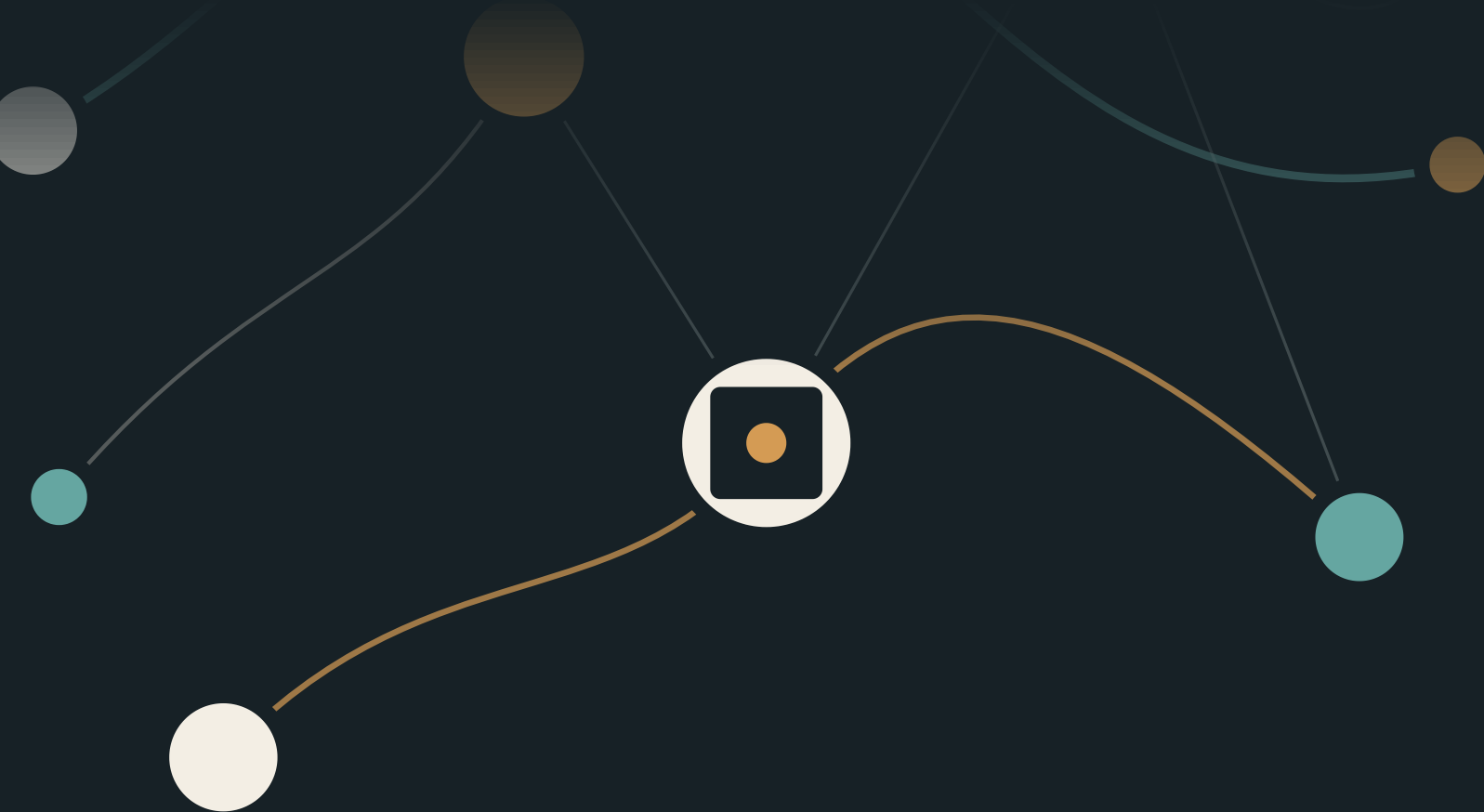
STUDY MONOGRAPH

Artificial Organizations

Understanding, designing, and
building institutions for agentic work

Curated and written by AI

Under the direction of
Dr. Paulo Salem



The information in this publication was researched and curated, and the text was written by artificial intelligence (AI), under the general direction of Dr. Paulo Salem.

General direction	Dr. Paulo Salem
Research, curation, and writing	Artificial intelligence, under human direction
Edition	Version 1.3 · 16 September 2026

Limited human review. The text was partly revised by Dr. Paulo Salem, who does not claim to have fully proofread it before release. *Errors, imprecisions, and other problems may remain.*

About the director. Dr. Paulo Salem holds a PhD in Computer Science from the University of São Paulo and Université Paris-Sud. His work spans AI, multiagent systems, formal methods, simulation, and tools for thought. He developed this monograph as an experiment in using AI to refine his own knowledge and judged it worth sharing with the community. The research, curation, and writing are by AI under his direction. [Biographical source: Paulo Salem, “About Paulo.”](#)

Research and interpretation

The monograph brings together historical literature, contemporary research, and technical standards. It distinguishes established findings, practical heuristics, disputed claims, and open questions. Cases draw on published research, institutional accounts, and reporting, not Dr. Paulo Salem's projects or clients. Reported facts, authors' thought experiments, benchmark simulations, and editorial design implications are explicitly distinguished. Illustrative arithmetic is not deployment evidence.

AI-generated research and writing can contain errors. References and evidence-status labels support scrutiny; they do not replace consulting primary sources or obtaining appropriate professional advice for consequential decisions.

Composition and production

The publication is composed from a maintained Markdown manuscript using Pandoc and paged.js. Mermaid diagrams and original vector illustrations are integrated with a dedicated print layout and a browser companion.

READER'S ORIENTATION

Designing accountable artificial organizations

In this monograph, **artificial organizations** are organizations in which software agents perform a material share of the work and coordination while people retain legal and moral responsibility. The ambitious form is an organization whose workers and middle managers are predominantly software agents, governed by people through objectives, roles, delegated authority, policies, decisions, and evidence rather than through a stream of prompts.

An existing term, a stated working definition. This book does not coin "artificial organization." The phrase appears in Ye and Carley's 1995 RADAR-Soar paper and Drogoul and Collinot's 1998 robotic-soccer study; Waites uses *Artificial Organisations* as the title of a 2026 LLM-system preprint [Ye and Carley 1995; Drogoul and Collinot 1998; Waites 2026]. These uses establish a history, not one universally accepted definition. The more established research traditions include **multi-agent systems**, **agent organizations**, **organization-oriented MAS**, and **computational organization theory**. This monograph brings their questions together with organization science and LLM-agent engineering. Its emphasis on durable responsibilities, governed work, and human accountability is an explicit working scope, not a claim that every earlier use of the term meant an autonomous business or met these requirements.

This study has two purposes. It explains the field, and it develops a method for building systems from its techniques without mistaking a collection of agents for an organization. Its central conclusion is:

Make organizational responsibilities, rules, and decisions explicit and inspectable. Select worker implementations to fit the task. Represent authority, commitments, policy, expenditure, state transitions, provenance, and acceptance in durable records and enforceable procedures rather than relying solely on a worker's internal state or self-report.

The evidence base combines organization science, human factors, concurrent and distributed-systems coordination, classical multi-agent systems (MAS), contemporary LLM-agent research, and official standards. Grouped references and topic-specific research routes let readers trace the argument to its sources without access to any private project or supporting workspace.

Anatomy of the book



The map connects the parts of an artificial organization to their chapters. Follow the arrows for the principal relationships; select a part to read about it. Printed page references lead to the same destinations. This is a reading map, not a prescribed hierarchy or a claim that every organization needs a separate department for each function.

Four ways to read this book

Reading goal	Route	What you will gain
Skim before a decision	Read each chapter's Core point , then Chapter 17	A balanced map of the argument and its limits
Learn the subject	Follow the main text, recurring cases, and visual studies	Vocabulary, technique selection, and a construction method
Work through the mechanics	Add the More detail boxes and integrated examples in Chapter 14	Assumptions, calculations, implementation boundaries, and counterexamples
Investigate a topic	Use Chapter 15's historical comparisons and the grouped References	Primary foundations, contemporary tests, and open questions

The main text carries the argument. **Key concept** boxes highlight distinctions; **Design implication** and **Caution** identify choices and limits. **Example**, **Practice note**, and **Biography** boxes offer illustrations, short guidance, and cultural context. **More detail** develops the reasoning in depth.

Source specimens sit beside their methods; reading maps are editorial syntheses. Verification limits accompany the individual examples. **Further afield** closes each substantive chapter with annotated research from other disciplines; proposed transfers remain exploratory. Use bookmarks for navigation and the [glossary-index](#) for definitions and principal discussions.

Evidence key

Claims are marked when their status matters:

- **[ESTABLISHED]** an empirical result within its documented study conditions, a formal result under stated assumptions, an authoritative standard, or a well-supported historical account. Independent replication is not implied unless explicitly identified.
- **[HEURISTIC]** practical engineering convention supported by experience or theory but not established as generally optimal.
- **[DISPUTED]** credible evidence or scholarly positions conflict.
- **[OPEN]** no satisfactory answer or adequate evidence is known.

Formal results retain their assumptions. Vendor documentation defines or advertises interfaces; verify availability, deployed behavior, and outcomes separately. Leaderboard values are dated because performance and evaluation conditions change.

Table of contents

Part I. Foundations

What an artificial organization is, which problems it solves, and when organization is the wrong answer.

- 1. The field and its boundaries 8
- 2. Cases and examples used in this book 13
- 3. A map of the field 18
- 4. Why organize at all? 23

Part II. Institutional mechanics

Roles, coordination, delegation, authority, attention, policy, and the machinery that makes them real.

- 5. Roles, missions, and obligations 49
- 6. Coordination and organizational topology 59
- 7. Delegation, authority, and human attention 83
- 8. Norms, policy, and institutional facts 96

Part III. Knowing and learning

Reliability, evidence, deliberation, memory, and the disciplined separation of claims from outcomes.

- 9. Reliability, evidence, and accountable closure 108
- 10. Deliberation, dissent, and decisions 123
- 11. Organizational memory and learning 134

Part IV. Construction

A runtime architecture, a ten-step method, and concrete examples connecting coordination, authority, evidence, and outcomes.

- 12. Runtime architecture and protocols 144
- 13. A construction method 166
- 14. Putting the concepts to work 172

Part V. Frontier and synthesis

What contemporary systems should inherit, which questions remain open, and where the evidence leads.

- 15. What LLM-agent systems should inherit 182
- 16. What remains open 191
- 17. Synthesis 195

References 198

Image credits 212

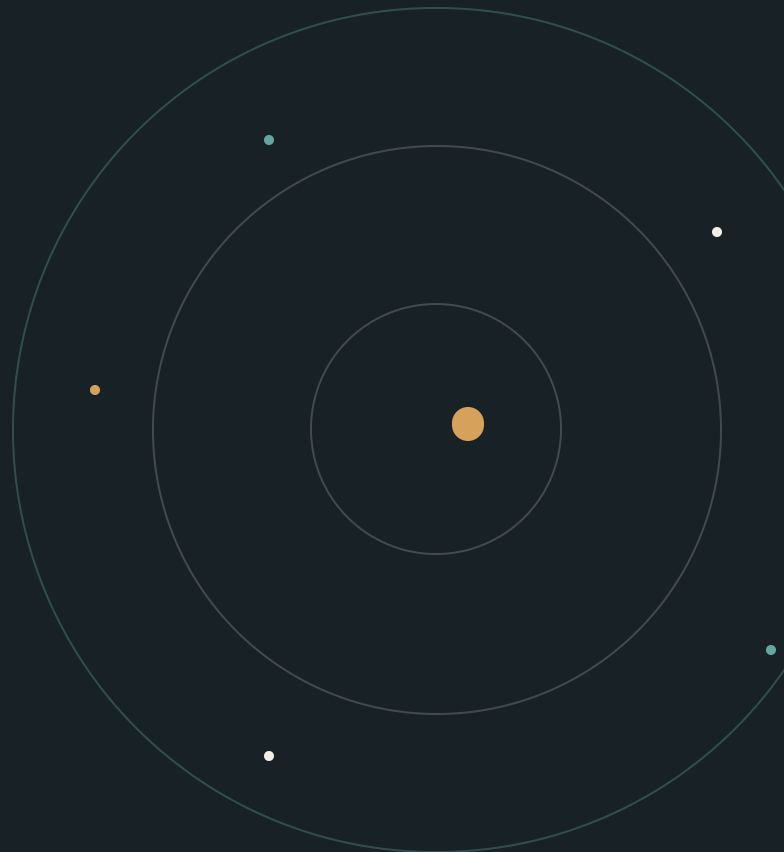
Core points and key concepts 214

Glossary and index 223

PART I

Foundations

What an artificial organization is, which problems it solves, and when organization is the wrong answer.



1 The field and its boundaries

© **Core point — Institutions, not agent headcount.** An artificial organization adds durable responsibilities, legitimate action, and accountable closure to computational agency. More agents do not, by themselves, supply any of these.

1.1 A working definition

An **agent** is a computational system that observes some state, chooses actions, and acts toward goals with some degree of autonomy. An LLM agent commonly adds a model-driven loop around tools, memory, and an environment. A **multi-agent system** contains multiple interacting agents. An **artificial organization** is more specific: it gives agents durable positions in a system of purpose, division of labor, authority, obligation, coordination, and accountability.

The distinction concerns institutional structure, not the label attached to a system. The following components can participate in an artificial organization, but none supplies all of its requirements by itself:

System	What it supplies	What is not established by that alone
One chatbot with tools	An interface for reasoning and action	Persistent responsibility and legitimate decision rights
A fixed automation workflow	Process and state transitions	How its work belongs to the surrounding institution
A group chat of agents	Multiple interacting participants	Durable roles, authority, commitments, and closure
A social simulation	A modeled population and environment	Operational responsibility for a real organization's work
An agent registry	Identity and declared capabilities	Purpose, obligations, and authority for particular actions
An artificial organization	Agents embedded in institutional structure	Safe autonomy, effective performance, or a new legal status

Artificial describes the worker and coordination substrate, not a new legal status. A company using agents remains subject to the laws and responsibilities that apply to its human and legal actors. Internal delegation cannot make legal accountability disappear [European Union 2024].

The recommendations chiefly concern digital work with inspectable records and controllable action boundaries. Robotics and sector-specific legal compliance require additional specialist knowledge. This is a critical synthesis of design alternatives, not a claim that one architecture has been proved optimal.

1.2 What problems does the field address?

The field begins where adding another capable agent stops being the main problem. It asks:

- 1. Purpose:** How does high-level intent become accountable work?
- 2. Division of labor:** Which role owns which mission, and can roles survive replacement of their occupants?
- 3. Coordination:** Which dependencies require communication, synchronization, or shared state?
- 4. Authority:** Which actions may be selected or executed, by whom, within what scope and budget?
- 5. Attention:** Which exceptions deserve scarce human judgment?
- 6. Governance:** How are norms created, applied, changed, violated, and repaired?
- 7. Epistemics:** What is known, how well is it supported, and what does silence mean?
- 8. Accountability:** What happened, why was it allowed, and was the intended outcome actually achieved?
- 9. Adaptation:** How does the organization learn, forget, restaff, or reorganize without losing obligations?

Adjacent fields contribute different pieces. Organization science explains bounded rationality, hierarchy, incentives, routines, and failure. Human factors studies automation, interruption, trust, and supervisory control. Classical MAS formalizes roles, norms, commitments, teamwork, and institutions. Distributed systems supply durable execution, identity, messaging, retries, and consistency. AI safety and governance contribute threat models, evaluation, and controls. Current LLM-agent engineering supplies flexible workers and lowers the cost of drafting natural-language specifications, while leaving validation necessary.

These traditions have not always developed together. A consequential **gap in the use of earlier research** arises when an older distinction is lost behind familiar terminology. Classical MAS treated roles, directed commitments, interaction rules, and institutional state as explicit design objects decades before LLMs. A modern agent's role description or a protocol's task status can resemble those objects without carrying their semantics. For example, MOISE, an organizational model, represents duties connecting roles to missions. Agent2Agent (A2A), a protocol for communication between agents, represents the progress of exchanged tasks. Those objects answer different questions [Hübner et al. 2002b; A2A 2026]. When an application needs both but represents only the latter, it must recover the missing organizational contract elsewhere.

Chapter 15 asks **which earlier ideas remain useful, where contemporary systems already implement them, and what is lost when they are omitted**. It compares records, transitions, and failure handling across dated sources, recognizing both the contributions of LLM

Bounded
rationality

A2A

ems and the limits of older approaches.

1.3 Starting concepts and reading prerequisites

The main argument assumes familiarity with ordinary software systems, but no prior training in organization science or multi-agent systems.

Later chapters introduce additional ideas from concurrency, distributed systems, probability, and formal reasoning. Readers can follow the conceptual discussion without mastering every technical detail; the implementation examples and quantitative sections require closer attention to their notation and assumptions. Building a dependable system requires practical engineering knowledge beyond the concepts introduced here.

Concept reference – the institutional vocabulary. These distinctions recur throughout the book. The concrete records describe implementation requirements; the vocabulary synthesizes organization-oriented MAS and institutional theory [Ferber and Gutknecht 1998; Hübner et al. 2002a; Jones and Sergot 1996].

Concept	Meaning in this manuscript	What makes it concrete	Do not confuse it with
Agent	Actor that observes, selects actions, and acts toward goals	Identified runtime actor with observations and action interfaces	A role or an organization
Role	Institutional position with duties, rights, and relationships	Versioned role definition and enactment interval	A persona description
Mission	Bundle of goals assigned as a responsibility	Named goals and success conditions	An individual tool call
Commitment	Directed undertaking to perform, sometimes conditionally	Debtor, creditor, content, conditions, lifecycle	A suggestion or unaccepted request
Authority	Governed allocation of decision and action rights	Scope, holder, validity, and revocation path	Credentials alone
Policy	Rule governing permitted, required, or recognized behavior	Authoritative version, applicability, enforcement, and remedy	The most recent discussion
Evidence	Material used to support or challenge a claim	Retained source, observation, provenance, and applicability	A fluent explanation
Outcome	Relevant state of the world after work	Observation against an explicit criterion and horizon	A worker finishing its run

Terms such as *role*, *commitment*, *regimentation*, and *positive predictive value* are introduced where they become useful.

1.4 Further afield

Top recommended readings

Direct literature, ranked by importance to this chapter's framing and value for deeper study. Throughout the book these are editorial priorities, not citation-count rankings. Scholarly details and available links are in References; primary case records are linked directly and documented in the relevant chapter's source notes.

1. Ferber and Gutknecht (1998), *A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems*.

Start here for the agent-group-role distinction behind this book's working definition. Coming from organization-oriented MAS before contemporary LLM teams, it makes an actor's institutional position explicit rather than inferable from its behavior. Read for what a role and group add to an agent description; this minimal model is not a complete theory of governance or reliable execution.

2. Ye and Carley (1995), *RADAR-Soar: Towards an Artificial Organization Composed of Intelligent Agents*.

This earlier computational-organization study is important historical evidence that the field's name and questions predate current agent products. Read it against the opening definition: identify the modeled participants, organizational arrangements, and explanatory purpose. A historical use establishes lineage, not agreement with every requirement here or readiness for modern deployment.

3. Drogoul and Collinot (1998), *Applying an Agent-Oriented Methodology to the Design of Artificial Organizations: A Case Study in Robotic Soccer*.

The robotic-soccer setting connects organizational analysis with situated behavior. It complements the preceding vocabulary with a design case, showing why environment and interaction matter. Compare its actual design objects with a present-day team of language models, without treating similarity of labels as equivalence of mechanisms or empirical conditions.

Broader explorations

Where does a thinking system end? Clark and Chalmers' *The Extended Mind* argues that some cognitive processes can include reliably coupled external resources, rather than ending at the person's body [Clark and Chalmers 1998]. Their notebook example is a philosophical thought experiment about the role information plays in action, not a clinical case or an agent-system benchmark.

This offers a different way to examine an assistant: treat the person, interface, tools, and records as a coupled working system. What competence disappears when one component becomes unavailable or misleading? The connection is methodological, not a claim that an organization has a conscious mind. Start with the authors' discussion of active externalism, then ask where a useful analysis boundary differs from a legal responsibility boundary.

Cognition distributed across a working setting. Edwin Hutchins' *Cognition in the Wild* studies navigation as an activity distributed among people, instruments, representations, and routines [Hutchins 1995]. Read it to question the habit of assigning all intelligence to an individual actor. In an artificial organization, a worker's apparent competence may depend on a well-designed record, a colleague's checking procedure, or an interface that makes an error visible. A useful investigation would remove or alter one such support and observe which capability disappears. The connection concerns the unit of analysis, not an assumption that a computational team reproduces human cognition.

Plans in actual situations. Lucy Suchman's *Human-Machine Reconfigurations* revisits plans, situated action, and the attribution of agency at interfaces [Suchman 2007]. It is useful beside an explicit workflow because a plan and the circumstances in which someone follows it are different objects. Ask how workers recognize an exception, recover context, or determine what an instruction means here. Compare the written process with observations of its use rather than treating deviations as noise by default. The point is not to abandon planning, but to investigate the practical work required to make a plan usable.

Follow associations before assuming a social whole. Bruno Latour's *Reassembling the Social* asks researchers to trace how associations are made and maintained rather than using “the social” as an unexplained cause [Latour 2005]. This offers a provocative counterpoint to the book's explicit organizational objects. Before declaring that an agent “belongs to a department,” follow its actual dependencies: permissions, records, tools, people, and rules. Which relationships sustain that description? Actor-network theory is not equivalent to computational actor models, and treating artifacts analytically as participants does not assign them human moral or legal responsibility.

Knowing exceeds what can be written down. Michael Polanyi's *The Tacit Dimension* examines forms of knowing that are not exhausted by explicit statements [Polanyi 1966]. It is a useful challenge to the ambition to capture an organization completely in prompts and policy documents. Investigate how experienced operators notice significance, recognize a familiar failure, or interpret an incomplete record. An observation protocol might compare the written procedure with the cues experts actually use. Tacit knowledge is not a license for unaccountable discretion; the research question is which judgments can be articulated, taught, checked, or supported without misdescribing them.

2 Cases and examples used in this book

© **Core point — Choose the example for the question.** Recurring incidents connect the chapters; organizational practices show how work is arranged; research studies and source specimens expose particular mechanisms. Each supplies a different kind of evidence. No single case demonstrates everything an artificial organization needs.

Examples in this book have different jobs. Public incidents make failures of meaning, responsibility, and recovery visible; studies and working practices show how decisions are arranged; technical specimens expose mechanisms. This chapter supplies enough context to recognize the examples when they appear. Their detailed analysis belongs beside the concept they explain.

The four incidents below recur because each makes a useful distinction, not because they are more important than the factory, TVA, or other studies. They concern organizations using technology, not four deployed LLM organizations. Their public accounts support the reported facts; proposed agent-assisted responses remain this book's recommendations, not demonstrated improvements. Monetary figures are US dollars unless another currency is explicitly identified.

Case A: Mars Climate Orbiter

Insight. The Orbiter loss shows that a technically successful data exchange can still fail when teams do not verify a shared interpretation of units.

On 30 September 1999, NASA/JPL reported that preliminary findings linked the spacecraft's loss to an uncorrected information-transfer error between the spacecraft team in Colorado and navigation team in California: one used English units and the other metric units. The release emphasized the failure of systems engineering and checks to detect the mismatch, not merely the existence of a human error.¹

The question is **shared meaning**: what demonstrates that two teams interpret a transferred value the same way? Section 4.6 develops the distinction between delivering information and preserving the meaning and checks it requires.

1. NASA/JPL, "Mars Climate Orbiter Team Finds Likely Cause of Loss," 30 September 1999, release 99-083. This is a preliminary institutional account, not the final investigation. <https://www.jpl.nasa.gov/news/mars-climate-orbiter-team-finds-likely-cause-of-loss/>.

Case B: Air Canada's bereavement-fare chatbot

Insight. In this dispute, a correct policy page elsewhere did not relieve Air Canada of responsibility for misleading chatbot advice on which its customer relied.

In November 2022, Jake Moffatt relied on Air Canada's chatbot advice that a bereavement-fare adjustment could be requested after travel. The linked policy page said otherwise. In *Moffatt v. Air Canada*, the British Columbia Civil Resolution Tribunal found negligent misrepresentation: the airline had not taken reasonable care to ensure the chatbot's information was accurate, and Moffatt reasonably relied on it to their detriment.²

The question is **responsibility for a public answer**: a correct policy page elsewhere does not necessarily repair misleading advice. Chapters 8–10 return to institutional effects, evidence, and challenged inferences. The decision is fact- and jurisdiction-specific; it neither identifies an LLM architecture nor establishes that every chatbot utterance changes policy or creates a contract.

Case C: GitLab's database outage

Insight. GitLab's configured backups and missing failure notifications concealed a recovery capability that had not been demonstrated when it was needed.

GitLab's published postmortem describes how, on 31 January 2017, an engineer trying to restore replication removed data from GitLab.com's primary database while believing the target was the secondary. The postmortem reports that ordinary backups were unusable, failure notifications had not reached their recipients, and recovery depended on a staging snapshot taken roughly six hours earlier. Some database changes were lost; Git repositories and wikis were stored separately and were not lost.³

The question is **demonstrated recoverability**, not merely whether a backup procedure exists. Section 12.2 diagrams the recovery dependencies; Section 14.4 connects ownership, restoration, and permanent loss alongside five other examples.

2. *Moffatt v. Air Canada*, 2024 BCCRT 149, Civil Resolution Tribunal, 14 February 2024: paragraphs 14–23 describe the chatbot evidence and policy conflict; paragraphs 24–32 explain negligent misrepresentation; paragraphs 40–44 give the remedy. Primary decision inspected 15 September 2026: <https://decisions.civilresolutionbc.ca/crt/crtd/en/item/525448/index.do>. The case does not identify an LLM architecture.

3. GitLab, "Postmortem of database outage of January 31," 10 February 2017, especially "Broken recovery procedures," "Recovering GitLab.com," and "Root cause analysis." This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.

Case D: Knight Capital's trading controls

Insight. Knight Capital's error messages did not produce an effective response, illustrating why detection is not a substitute for action or enforced exposure limits.

The SEC's 2013 account of Knight Capital's August 2012 trading incident describes erroneous automated orders reaching the market and inadequate controls over the resulting exposure. It also describes automated emails about an error condition that did not produce an effective response. Knight consented to the order without admitting or denying the findings.⁴

The question is **effective control**: detection, timely intervention, and enforced exposure limits are different mechanisms. Chapters 4, 12, 14, and 15 use the incident to distinguish fast execution from appropriately controlled work.

A guide to the other examples

The later examples serve different questions. Read a practice as a documented arrangement, a study as evidence under stated conditions, and a specimen as an inspectable mechanism. None demonstrates organizational success by itself.

Example and where it appears	What it helps explain	How to read its evidence
Berkeley museum, Chapter 3	Cooperation across perspectives	Historical study of boundary objects
Paint factory and TVA, Chapter 4	Operating decisions and institutional partnerships	Retrospective modeling and field research
Kiva robots and LLM board-member research, Sections 4.2–4.3	Parallel work, shared information and focused investigations	Deployment and evaluation accounts by system developers
Coal-sector relationships, Section 4.7	Contracts versus internal organization	Industry research; photographs show context
Toyota, Chapter 7	Detection linked to stopping and response	The organization's account of its practice
IETF, Chapter 10	Technical objections versus headcounts	Published principles, not guaranteed practice

4. U.S. Securities and Exchange Commission, "SEC Charges Knight Capital With Violations of Market Access Rule," 16 October 2013, release 2013-222; links to administrative order 34-70694. Knight consented without admitting or denying the findings. <https://www.sec.gov/newsroom/press-releases/2013-222>.

Example and where it appears	What it helps explain	How to read its evidence
Generative Agents and Reflexion, Chapters 11-12 and 16	Retention and reflection across interactions	Architectures evaluated for specific tasks
OSWorld, TheAgentCompany, and MAST, Chapters 9 and 12	Agent tasks and observed failures	Benchmarks and traces, not company readiness
MOISE, InstAL, protocols, and runtimes, Chapters 5-12 and 16	Explicit records, rules, and operations	Source specimens with stated verification limits

Real cases show how several problems interact. To understand one mechanism at a time, the book also uses thought experiments, worked calculations, and proposed tests. A calculation can show how false alarms accumulate; a test can reveal whether a duty survives replacement of its worker. These simplified examples help explain what to look for in practice. They are identified as illustrations or tests so that their lessons are not mistaken for observed results.

2.1 Further afield

Top recommended readings

Direct case sources, ranked for understanding this chapter's recurring examples; this is not a ranking of the incidents' historical severity.

1. NASA/JPL (1999), [Mars Climate Orbiter Team Finds Likely Cause of Loss](#).

Begin with this preliminary institutional account to see why the book treats the event as a failure of shared meaning and checks, not merely a memorable units error. It gives a compact primary-source entry into a consequential organizational handoff. Preserve its date and preliminary status: the release is not the final investigation and does not test the controls proposed here.

2. GitLab (2017), [Postmortem of database outage of January 31](#).

Read the broken recovery procedures, actual recovery sequence, and root-cause discussion. The operator's concrete retrospective connects ownership, backups, artifact dependencies, restored service, and unrecovered data. Its value is operational specificity, not proof that one organization's recovery practice is universal. Distinguish the account's observations from this book's later institutional interpretation.

3. U.S. Securities and Exchange Commission (2013), [Knight Capital order 34-70694 and accompanying release 2013-222](#).

This enforcement record anchors the distinction between a technically possible action and adequate controls before market exposure. Read the findings on deployment and market access alongside the proposed safeguards. Knight consented without admitting or denying the findings; the order is not an experiment and cannot by itself estimate the effectiveness of a new agent-control architecture.

Broader explorations

History seen backward. Fischhoff's study of hindsight and foresight asks how knowledge of an outcome affects judgment under uncertainty [Fischhoff 1975]. It is a useful companion to incident reports: readers already know which warning mattered, whereas participants had to choose among uncertain signals.

Try reading a case in two passes. First stop before the outcome, record plausible explanations and available actions, and only then read the resolution. For agent evaluation, the exploratory counterpart is to withhold the answer from a replay and measure what the system could infer from the contemporaneous record. This need not excuse poor decisions; it helps distinguish information available at the time from clarity supplied by hindsight. The procedure proposed here is an application of the research question, not Fischhoff's experimental protocol.

Availability, resemblance, and anchoring. Tversky and Kahneman's "Judgment under Uncertainty: Heuristics and Biases" analyzes economical judgment strategies and their characteristic errors [Tversky and Kahneman 1974]. Read it before letting a memorable incident define an entire risk model. A dramatic outage may be readily recalled while ordinary successful recovery remains invisible; a plausible-looking agent may resemble a competent colleague without sharing that colleague's experience. Compare judgments made from a vivid narrative with judgments made from denominators and alternative cases. These human findings suggest tests of interfaces and evaluation procedures, not an automatic diagnosis of the internal mechanism of an LLM.

A disaster develops through normal work. Diane Vaughan's *The Challenger Launch Decision* reconstructs the organizational setting of the shuttle launch decision rather than reducing it to a single foolish choice [Vaughan 1996]. Its relevance is how repeated interpretations, production pressures, and accepted practices can make a dangerous condition appear ordinary. Use that perspective to ask what evidence was available at each stage and why it was interpreted as it was. A responsible transfer to agent operations would inspect how an exception becomes routine over repeated runs. It would not assert that every automation incident has the same social or technical causes.

Distinguish the error from its conditions. James Reason's *Human Error* examines different forms of error and the conditions that allow them to become consequential [Reason 1990]. It helps a case reader distinguish an immediate mistake from the defenses, interfaces, and organizational arrangements around it. In GitLab's account, selecting the wrong database host and lacking dependable recovery were different problems. Ask what intervention addresses each one. The analogy should not collapse human cognitive errors, malicious input, and software faults into one category. Its value is a more discriminating account of failure and of the barriers that could interrupt its consequences.

What would justify a causal lesson? Campbell and Stanley's *Experimental and Quasi-Experimental Designs for Research* introduces threats to causal inference when comparing interventions [Campbell and Stanley 1966]. An incident can reveal that a control failed, but it rarely tells us what would have happened under every proposed alternative. Read this work to distinguish an instructive counterfactual from a tested improvement. For a new review procedure, consider selection effects, concurrent changes, and differences between groups or periods. The goal is not to demand an experiment for every historical statement; it is to match the strength of a design recommendation to the evidence supporting it.

3 A map of the field

© **Core point — Separate the layers, then connect them.** Purpose explains why work exists; roles allocate responsibility; governance constrains action; runtime systems execute it; evidence tests whether the outcome occurred. These connections need not correspond to separate products.

The [opening anatomy map](#) locates the main parts of an artificial organization. Here the question changes from **what the parts are** to **which technique fits the problem**. These are complementary concerns, not a compulsory stack: a workflow, a role model, and an evidence record can solve different parts of the same problem without becoming separate departments.

Problem class	Technique families	Complementary techniques	Simpler alternative
Purpose decomposition	Goal trees, missions, accountable outcomes	Roles, commitments, budgets	One explicit owner and checklist
Division of labor	AGR, MOISE+, enterprise ontologies	Capability registry, staffing policy	One agent with tools
Coordination	Pipeline, hierarchy, blackboard, market/contract, teams	Durable workflow, shared artifacts	Deterministic function calls
Authority	Stage-specific automation, permission leases	Policy engine, capability security	Manual execution
Exception routing	STEAM calculus, signal detection, sentinels	Negotiated interruption, digests	Fixed high-risk gate
Governance	Norms, regimentation, enforcement, counts-as	Commitments, sanctions, reparations	Static allow-list

Problem class	Technique families	Complementary techniques	Simpler alternative
Epistemic quality	Provenance, Toulmin packets, subjective logic	Calibration, independent verification	Source links and tests
Deliberation	Argumentation, debate, decision records	Architectural dissent, meeting protocol	One accountable decider
Memory	Retention bins, transactive memory, precedent	Retrieval, policy versioning, forgetting	Versioned documents
Adaptation	Dynamic teaming, reorganization	Evaluation and change control	Human redesign

The **“simpler alternative”** column matters. If one competent worker or a deterministic workflow can perform a bounded task, additional managers and committees need a demonstrated benefit. Simpler execution still belongs to an organization: the applicable authority, evidence, and accountability requirements do not disappear when the task needs only one worker. Chapter 4 examines this cost comparison; Chapters 13 and 16 turn it into implementation and evaluation questions.

3.1 A historical spine, not a succession of replacements

Concurrent-systems coordination is a foundation, not merely an implementation detail. CSP specifies synchronized communication; Linda supplies associative shared-space operations; later work examines failure, agreement, and convergence [Hoare 1978; Gelernter 1985; Gelernter and Carriero 1992; Hellerstein and Alvaro 2019]. A mission states an institutional requirement; the substrate determines which concurrent transitions occur. Chapter 6 connects them; the [Coordination research report](#) provides the deeper comparison and source-access ledger.

The historical strands below address complementary problems. Organization science examines how people divide decisions and work; agent research develops computational models of collaboration; concurrent systems determine what happens when operations overlap or participants fail. LLMs broaden what individual workers can interpret without retiring those questions.

Tradition and representative work	Enduring question	Transfer boundary
Behavioral organization theory: Simon (1947); March and Simon (1958)	How can limited decision makers act coherently?	Routines and attention matter; agents do not automatically share human motives
Distributed AI: Smith (1980); Erman et al. (1980)	Who works next, with which inputs?	Contracts and shared state transfer; communication and rationality assumptions need checking
Concurrent systems: Hoare (1978); Gelernter (1985); Hellerstein and Alvaro (2019)	Which interactions need synchronization, and what survives concurrency or failure?	Shared-space operations and convergence require explicit assumptions; they do not establish institutional authority

Linda / tuple spaces
Communicating sequential processes

Tradition and representative work	Enduring question	Transfer boundary
Agent organizations: AGR (1998); MOISE+ (2002); OperA (2004)	How do positions, goals, and duties relate?	Explicit organizational objects transfer; formal consistency does not establish competence
Supervisory control: Bainbridge (1983); Parasuraman et al. (2000)	When does automation help or burden a person?	Stage-specific authority matters; original studies are not direct LLM-company trials
Agent evaluation: OSWorld (2024); Vending-Bench (2025)	Can a worker sustain action in an environment?	Evaluate trajectories and outcomes; a benchmark score is not institutional readiness



 Practice note

A design method is not a runtime.

Some approaches help describe the organization; others guide its design. AGR (Agent-Group-Role) and MOISE+ provide organizational models. Gaia structures analysis of roles, interactions, and rules; Tropos carries goals and dependencies through development [Zambonelli et al. 2003; Bresciani et al. 2004]. OperA separates organizational requirements, participation arrangements, and actual interactions [Dignum 2004].

Start with goal analysis when needs are unclear, then role/interaction design when responsibilities are clearer. Represent duties at runtime when execution must be governed. These are selection heuristics, not comparative-trial results. Borrowing a useful distinction may cost less than adopting a whole formalism. None replaces durable execution.

From explanation to construction. Chapter 4 first asks whether organization repays its cost. Chapters 5–8 develop its institutional mechanisms; Chapters 9–11 examine evidence, decisions, and learning. Chapters 12–14 put these ideas to work through infrastructure choices, a construction method, and six integrated examples without privileging one setting. The closing chapters compare historical and contemporary approaches, identify open questions, and draw the argument together. A role model, an outcome test, and an authority check remain distinct throughout: each contributes something the others cannot establish.

3.2 Further afield

Top recommended readings

Direct foundations, ranked to move from organizational explanation to explicit design without confusing either with an execution substrate.

1. **Simon (1947), *Administrative Behavior*.**
- This foundational study places decision making, bounded rationality, and organizational arrangements in one explanation. It supplies the reason to map a field by problems rather than fashionable tools: organization changes what limited participants can know and decide. Read it before importing a supposedly ideal hierarchy into artificial work; it is an analytical foundation, not an agent framework.

2. Ferber and Gutknecht (1998), *A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems*.

AGR gives a compact entry into the computational strand of the map. It separates participant identity from group membership and role, giving later methods a clear object of discussion. Compare this vocabulary with the chapter's technique families; its contribution is explicit organization, not a claim to supply every coordination, policy, and evidence mechanism.

3. Dignum (2004), *A Model for Organizational Interaction: Based on Agents, Founded in Logic*.

OperA develops organizational requirements, participation arrangements, and concrete interactions as distinct concerns. Historically it advances the move from isolated agent behavior toward organizational contracts. Read for how heterogeneous participants can meet shared requirements without identical internal plans; formal articulation still needs implementation and evaluation.

Broader explorations

Cooperation without one shared worldview.

Insight. The Berkeley museum study shows that shared artifacts and methods can sustain cooperation among groups without requiring them to adopt one worldview.

Star and Griesemer studied the early Berkeley Museum of Vertebrate Zoology, where amateurs, scientists, and administrators cooperated through standardized methods and *boundary objects* [Star and Griesemer 1989]. Such objects remain recognizable across groups while being usable from different perspectives. This is a real institutional history, not a claim that all participants agreed on a common ontology.

For an artificial organization, a review packet might serve as a boundary object between a researcher, a finance specialist, and a decision owner. Explore which fields must have one strict meaning and which interpretations can legitimately differ. The unusual lesson is that successful coordination may require a shared artifact without requiring complete conceptual agreement. Ambiguity about authorization, however, is not made acceptable by calling a document a boundary object.

Classification is part of the infrastructure. Bowker and Star's *Sorting Things Out* examines how categories and standards organize experience and consequences [Bowker and Star 1999]. Read it alongside any taxonomy of agents, roles, or failures. A classification does more than shorten descriptions: it can make some work visible, hide exceptions, and determine which cases receive attention. Ask who designed the categories, how ambiguous cases are handled, and what becomes costly to express. An engineering study could compare two classification schemes against the same incidents and track the different decisions they support. A neat taxonomy is not automatically an adequate one.

A useful interface may still lack usable infrastructure.

Insight. The Worm Community System study shows that people can value an application yet struggle to use it because access and integration depend on the surrounding infrastructure.

Star and Ruhleder's study of the Worm Community System examines infrastructural complexity in a collaborative setting [Star and Ruhleder 1996]. Users could value the system while still encountering difficulties accessing and integrating it into their work. This is an important adjacent question for a field map: which prerequisites are assumed by the technique being compared? A working demonstration may depend on support, vocabulary, institutional changes, and other arrangements outside the interface. Study those dependencies explicitly before attributing success or failure to a local component alone. Infrastructure is relational to a practice, not simply a layer of machines below it.

Who has jurisdiction over a problem? Andrew Abbott's *The System of Professions* studies relationships among professions and their claims over tasks and expertise [Abbott 1988]. It offers a different lens on “role design”: the boundary between analyst, engineer, auditor, and manager may reflect a history of jurisdiction rather than a technically inevitable decomposition. Ask what knowledge and authority each professional claim includes, and what happens when new tools alter the work. For artificial organizations, this can inform research on disputed handoffs and responsibility gaps. It does not imply that giving an agent a professional title supplies professional standing.

Fields also construct their boundaries. Thomas Gieryn's analysis of boundary-work examines how scientific authority is distinguished from other activities [Gieryn 1983]. Read it when a field presents its own terminology as the natural map of a problem. Which competing accounts are excluded, and on what grounds? A study of agent research could compare explicit technical distinctions with claims about what counts as “real” agency or coordination. The point is to examine those claims rather than treating every boundary as either objective fact or mere rhetoric. Bibliographic coverage should include the work needed to understand the problem, even when it sits outside a favored label.

4 Why organize at all?

© **Core point — Organization must repay its overhead.** Specialization and exception routing can save expertise, but handoffs add delay, information loss, and failure boundaries. Compare against a competent single worker or a deterministic workflow before adding management.

Organization can make work possible by dividing a difficult task, concentrating expertise and bringing separate resources into cooperation. It must also reconnect the parts: decisions interact, information is lost at handoffs, and participants have interests of their own. This chapter moves from those gains and mechanisms to three practical comparisons: joint planning in a paint factory, local reach through TVA's partners, and ownership versus contracting in coal supply. It then asks when one capable worker or a simpler workflow is enough.

4.1 Bounded rationality and information compression

Herbert Simon's **bounded rationality** rejects the assumption that decision makers have complete information and unlimited computation. His later formal model develops search using aspiration levels: an acceptable option can end search without establishing a global optimum [Simon 1947, 1955]. Agents are also bounded by context, model capability, latency, cost, and incomplete access.

Bounded
rationality

Biography

Herbert A. Simon (1916-2001).



Herbert A. Simon, in a portrait published in RIT's News and Events, 1981.

Simon connected the study of organizations with psychology, economics, and computing. Trained in political science at the University of Chicago, he spent most of his career at Carnegie Mellon. His question was how people actually decide when time, knowledge, and attention are limited, rather than how an all-knowing optimizer would choose.

Administrative Behavior and his work on bounded rationality made those limits central to organization theory. With Allen Newell and J. C. Shaw he helped develop early artificial-intelligence programs, including the Logic Theorist. He shared the 1975 Turing Award with Newell and received the 1978 economics prize. His career is a reminder that organization theory and AI have a shared intellectual history, not merely a recent practical connection.⁵

[Portrait credit.](#)

5. Hunter Heyck, "Herbert Alexander Simon," ACM A. M. Turing Award biography, inspected 16 September 2026. https://amturing.acm.org/award_winners/simon_1031467.cfm.

 Key concept
Bounded rationality.

A decision maker cannot examine every alternative, consequence, and piece of evidence. Simon makes those limits part of the explanation of choice, rather than treating them as an incidental defect [Simon 1947]. In an artificial organization, assess the decision process against its actual information, capability, time, and resource constraints.

Organization is partly an information-processing response. Radner and Bolton & Dewatripont modeled hierarchies as communication networks; Garicano modeled a knowledge hierarchy in which **routine problems stay with production workers** and **exceptions move to scarce experts** [Radner 1993; Bolton and Dewatripont 1994; Garicano 2000]. **[Established under model assumptions]** Hierarchy can economize on communication and specialized knowledge.

This is not an argument that hierarchy is always best. Hierarchy can **distort information** when summaries conceal assumptions or contrary observations. Sah and Stiglitz showed that hierarchical and polyarchic decision structures trade different error types: sequential veto structures reject more bad projects but also reject good ones; independent acceptance channels admit more good projects and more bad ones [Sah and Stiglitz 1986].

Choice rule. Use hierarchy when **exceptions are rare**, expertise is scarce, and accountability needs a clear path. Prefer lateral or polyarchic review when **independent discovery matters** and false rejection is costly. Combine them by letting multiple scouts propose while a bounded authority chain executes.

 Key concept
Satisficing.

Search can stop when an option meets an aspiration or acceptability threshold, rather than after proving it globally optimal [Simon 1955]. This is not permission to accept arbitrary quality. For an engineered workflow, make the threshold and stopping condition explicit; whether the threshold is good enough remains a separate design question.

 Source example 1. Simon's watchmakers: stable intermediate results survive interruption

Insight. In Simon's thought experiment, stable subassemblies make complex work less vulnerable to interruption by preserving completed parts of the job.

Simon imagines two watchmakers. Tempus's unfinished watch falls apart when interrupted; Hora builds stable subassemblies. Both products have roughly 1,000 parts; Hora's intermediate assemblies preserve progress [Simon 1962, pp. 470–471].

Mechanism. Stable intermediate structures reduce the work lost per interruption. The useful boundary encloses a result that survives while attention switches elsewhere. This explains why decomposition needs durable intermediate results, not just a list of subtasks or managerial titles.

Real-world connection. GitLab's staging snapshot preserved recoverable state; unavailable logical backups did not. The analogy suggests inspecting which artifacts survive interruption. It does not imply that checkpointing a chat recovers an external database or that Simon's arithmetic predicts restore time.

Source and verification. Simon's thought experiment, paraphrased from the original article. Its arithmetic illustrates the mechanism; it is not an observed comparison of firms.⁶

4.2 Information processing: five complementary explanations

The information-processing tradition offers complementary explanations of how organizations manage complexity. Simon's **near-decomposability** concerns the relative strength of interactions [Simon 1962]. GitLab stored repositories separately from the affected application database, limiting which data was lost, but restoring the application still required dependent database and webhook operations.⁷ This is an application of the distinction, not an empirical test of Simon's model.

Key concept

Near-decomposability.

Simon's important distinction is **stronger interaction within subsystems** and **weaker interaction between them**, not complete independence [Simon 1962]. It explains why some complex systems can be understood in parts. A task boundary is useful only if it respects the interactions that still cross it.

Source example 2. Simon's rooms: timescale determines what can be summarized

Insight. Simon's rooms illustrate why stronger interactions within a unit can make its aggregate state a useful summary of slower interactions with other units.

Simon illustrates near-decomposability with heat moving through an insulated building. Walls slow exchange between rooms; heat moves faster between cubicles within each room. Temperatures settle within rooms before the building reaches overall equilibrium. At the slower timescale, room-level temperatures can describe behavior that initially needs finer detail [Simon 1962, pp. 474–475].

Mechanism. Stronger interaction within rooms makes a room a useful unit of description at the slower timescale. This is the chapter's task-boundary question in physical form: retain detailed coordination where interactions are strong, and summarize across weaker boundaries without erasing their effects.

6. Herbert A. Simon, "The Architecture of Complexity," *Proceedings of the American Philosophical Society* 106(6), 467–482 (1962). Academic source copy: <https://www2.econ.iastate.edu/tesfatsi/ArchitectureOfComplexity.HSimon1962.pdf>. These boxes paraphrase the author's examples.

7. GitLab, "Postmortem of database outage of January 31," 10 February 2017, especially "Broken recovery procedures," "Recovering GitLab.com," and "Root cause analysis." This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.

Real-world connection. GitLab's separately stored repositories and application database had different loss outcomes. One “recovered” status would conceal that distinction. The shared question is which detail a summary may safely omit; the physical processes in the two examples are different.

Source and verification. Simon's illustrative physical model, paraphrased from the original article. Its lesson concerns relative coupling and timescales; the appropriate summary depends on the question being studied.⁸

The computational counterpart is **factorization**: represent a large decision through smaller, interacting parts. Guestrin, Koller and Parr use factored dynamics and an approximate factored value function to coordinate a multiagent system [Guestrin et al. 2001]. Their method also serves single-agent planning. Thus, good decomposition can make a team effective without making agent count the explanation of the gain.

Key concept

Factorization.

Express a problem through smaller components while retaining the interactions needed for a correct combined result [Dechter 1999; Guestrin et al. 2001]. Independent components can be optimized separately; coupled components require coordination. The saving comes from exploiting structure, which either a central program or several agents may do.

The practical question is which decisions a worker can handle locally and which dependencies must remain visible to the rest of the system. Kiva's warehouse robots provide a deployed example: they divided the physical work of retrieving stock, while coordination and shared information kept the fleet useful as a whole.

Example

Kiva: bringing the shelves to the worker.

Insight. Kiva distributed shelf transport across mobile robots while sharing navigation corrections across the fleet, combining parallel work with collective improvement rather than making every robot independently solve the warehouse.

The real problem. Filling customer orders required retrieving products scattered through a warehouse. Kiva changed the work arrangement: small robots lifted movable shelving pods and brought the products to stationary human pickers. The first permanent installation began operating in summer 2006 [Wurman et al. 2008]. The agents here are physical robots with sensing and motion control, not conversational personas.

Follow the work. The documented goods-to-person process has three concrete contributions. A robot reaches and lifts a shelving pod; it carries that pod to a picking station; a human removes the required items. Other robots can transport other pods at the same time. The robot handles movement under its current load, while fleet coordination must account for vehicles sharing the same workspace. Delivery of a pod and completion of an order are different achievements: the human picking step still matters.

8. Herbert A. Simon, “The Architecture of Complexity,” *Proceedings of the American Philosophical Society* 106(6), 467–482 (1962). Academic source copy: <https://www2.econ.iastate.edu/tesfatsi/ArchitectureOfComplexity.HSimon1962.pdf>. These boxes paraphrase the author's examples.

What the robots had to learn. In D'Andrea's account, loads ranged from a few pounds to over a thousand, with different mass distributions, and the deliberately inexpensive robots varied in manufacture. They learned to move under those conditions before deployment and continued adapting as they wore. Navigation also depended on imperfectly known floor features. When one robot improved its estimate of a feature, that correction was shared with the fleet. Each robot therefore did not have to rediscover everything for itself.

The organizational lesson. Adding robots supplied simultaneous physical capacity; sharing corrections reduced duplicated discovery; coordinating motion addressed the interactions that division of labor could not remove. A more capable single robot would still have to move between locations, while one controller could coordinate many robots. The relevant distinction is between centralizing decisions and collapsing the workforce into one actor.

Observed benefit. The 2008 authors report worker productivity increasing by a factor of two or more as products came to workers instead of workers walking to products. This is a benefit of the redesigned human-robot work system, not a measured reduction from exponential to linear computation.

Source note. The deployment date and productivity claim come from the system designers' journal account; the adaptation and shared floor-feature corrections come from cofounder Raffaello D'Andrea's retrospective. The sequence above explains their reported operating process, not an individual order trace reproduced for this book; no robots were run here.⁹

Kiva makes **local work, shared knowledge and coupled action** tangible. It does not establish an exact factorization of warehouse performance or prove that a fully decentralized planner beats a central one. For an artificial organization, the corresponding questions are concrete: which work can proceed separately, which discoveries should be shared, and which resource conflicts still require a common decision? The paint factory later in this chapter shows why treating coupled decisions as independent can produce the wrong answer.

Where the formulas belong. In a specified discrete optimization model, N decisions with M options have M^N complete combinations. Independent additive choices can be optimized in $O(NM)$ work; pairwise tree interactions permit $O(NM^2)$ dynamic programming. More general interaction patterns depend on **induced width**, the largest dependency scope created during elimination [Dechter 1999; Fioretto et al. 2018]. These are conditional algorithmic results, not measured costs of the Kiva fleet or laws about LLM teams. The deployment illustrates why useful decomposition must preserve coordination, while the formal literature specifies when decomposition gives exact computational savings.

March and Simon's **uncertainty absorption** shifts attention from system boundaries to communication [March and Simon 1958]. When a summary replaces its underlying observations, preserve the assumptions, exclusions, and contrary evidence needed to challenge the inference without forwarding every record.

9. Peter R. Wurman, Raffaello D'Andrea, and Mick Mountz, "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses," *AI Magazine* 29(1), 9–20 (2008), DOI 10.1609/aimag.v29i1.2082. Journal abstract checked: <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2082>. Raffaello D'Andrea, "Kiva Systems / Amazon Robotics," project retrospective, inspected 20 September 2026: <https://raffaello.name/project/kiva-amazon-robotics/>. Full journal PDF retrieval was unavailable during this review; operational adaptation details are attributed to the accessible coauthor account, not inferred from the abstract. Both sources are insider accounts, not an independent controlled comparison of single- and multi-agent planners.



Key concept

Uncertainty absorption.

Recipients often receive an inference instead of the observations that produced it [March and Simon 1958]. This economizes on attention but can turn tentative judgments into apparently settled facts. Preserve the assumptions and contrary evidence needed to reopen the inference; forwarding everything is not the only alternative.

Information-processing design

Galbraith links organizational design to the information that must be processed during uncertain tasks [Galbraith 1973]. There are two broad responses: **reduce information-processing need** or **increase processing capacity**. Slack resources and more self-contained tasks reduce the need for continual adjustment. Vertical information systems and lateral relationships increase the capacity to handle it. This is a design framework, not a calibrated formula in which uncertainty and interdependence have universal numerical coefficients.



Key concept

Information-processing need versus capacity.

Galbraith's two responses are to reduce the coordination information a task requires or increase the organization's capacity to process it [Galbraith 1973]. Better task boundaries and slack address the first; information systems and lateral relationships address the second. More reporting does not necessarily do either.

Design move	Question raised by the GitLab case	Cost or limitation
Add slack	Is sufficient recovery capacity available when ordinary replication fails?	Capacity is paid for even when unused
Make work more self-contained	Can an independently tested restore package include its required data and dependencies?	A boundary that omits web-hooks is incomplete
Improve vertical information	Does an accountable owner receive evidence that backups and restore tests succeeded?	More messages can merely move overload upward
Add lateral coordination	Can storage, database, and application specialists resolve restoration dependencies together?	Joint adjustment needs time and clear closure

Task interdependence

Thompson distinguishes **pooled**, **sequential**, and **reciprocal** interdependence [Thompson 1967]. Units with pooled interdependence share a larger enterprise or resource base without consuming each other's outputs directly; standardization can help. Sequential work benefits from plans and schedules. Reciprocal work requires mutual adjustment as participants alter each other's inputs. These are characteristic coordination needs, not guarantees that a single mechanism will always suffice.

 Key concept**Task interdependence.**

Pooled work shares an enterprise or resource base; sequential work consumes upstream outputs; reciprocal work repeatedly changes other work's inputs [Thompson 1967]. The distinction helps choose standardization, planning, or mutual adjustment. A reciprocal task does not become a pipeline merely because the interface displays it in columns.

 **Design implication – Reduce the need to interrupt before improving the interruption interface.** Ask whether better task boundaries, shared standards, reserved capacity, or lateral coordination can resolve the dependency locally. Send an exception upward when the responsible lower-level actors lack relevant knowledge, authority, or resources, or when oversight policy independently requires review. Record the reason. Do not route every event to a director merely because a dashboard can display it [Galbraith 1973; Garicano 2000].

Mintzberg's coordination mechanisms cut across those dependencies: mutual adjustment, direct supervision, and standardization of work processes, outputs, or skills [Mintzberg 1979]. For artificial organizations, distinguish checking a procedure, checking an output, and checking a worker's demonstrated competence. They provide **different kinds of assurance**. A role description that says “expert” is **not the equivalent of professional training** or a capability evaluation.

 Key concept**Coordination mechanisms.**

Standardizing a process, an output, or a skill provides different assurance; direct supervision and mutual adjustment coordinate in other ways [Mintzberg 1979]. Specify what is actually standardized or checked. A role title cannot substitute for demonstrated skill, and procedural compliance cannot establish that an outcome was achieved.

4.3 Specialization, parallel capacity, and local agency

Decomposition makes work manageable; distributing it can supply resources and knowledge that a single worker lacks. Stone and Veloso identify parallelism, geographic distribution, specialized capabilities, modularity and robustness as reasons to use multiple agents [Stone and Veloso 2000, Section 2]. In human organizations, acquiring expertise takes time and people cannot work in two places at once. Artificial workers change those costs, but still operate with limited context, tool access, computation and time.

Specialization conserves scarce capability. Garicano's knowledge hierarchy reserves expertise for unfamiliar problems; Bolton and Dewatripont explain how specialization gains compete with communication costs [Garicano 2000; Bolton and Dewatripont 1994]. In an agent system, a specialist may have a trained model, restricted tools or a focused working history. A job title alone supplies none of these. The useful test is whether the specialist performs its contribution better or more economically, including the handoff.

Parallel work increases capacity when contributions can proceed separately. Several researchers can investigate different sources while an integrator compares their findings. For LLM agents, separation can also determine which search results and intermediate reasoning remain in each working context. Anthropic reports a concrete task on which that arrangement outperformed its single-agent research baseline [Hadfield et al. 2025].

 Example

Researching the boards of S&P 500 technology companies.

Insight. In Anthropic's reported board-member research task, a team of search agents found the answers its single-agent baseline missed, illustrating the value of dividing broad research into focused investigations before synthesis.

The task and observed result. Identify all the board members of the companies in the S&P 500's information-technology sector. Anthropic reports that its team succeeded by decomposing the research, while a single Claude Opus 4 agent failed with slow, sequential searches. The wider internal research evaluation compared an Opus 4 lead with Sonnet 4 subagents against single-agent Opus 4 and reported a 90.2% improvement; that aggregate figure is not a score for this one query.

What the separate agents did. In the reported system, the lead defines research subtasks and their output requirements. Each researcher chooses searches, reads results and follows leads in its own context. It returns findings to the lead, which synthesizes them and decides whether more research is needed; a citation agent then locates supporting passages. Thus the lead can manage coverage without carrying every worker's browsing history in its own context. The article describes this architecture, rather than publishing a complete assignment-by-assignment trace of the board-member query.

Why this is more than changing hats. A single conversation alternating among companies still accumulates one shared trail of pages, failed searches and partial answers. Separate contexts let a researcher continue its own investigation while others work elsewhere, and return the information needed for integration. Anthropic also reports that vague delegation caused duplicate or mismatched research: the separation helped only when subtasks had clear boundaries. The mechanism is focused working state plus parallel exploration, not different personalities or independently trained experts.

What remains to test. A single controller can also schedule isolated contexts and parallel tool calls. The published comparison does not establish that a team beats that stronger alternative at the same spending and deadline. Anthropic explicitly reports high token use; it does not isolate context separation from additional computation. The practical lesson is to test this architecture for broad research with separable evidence, not to add agents to every sequential reasoning task.

Source note. Hadfield and colleagues' engineering account, 13 June 2025, sections “Benefits of a multi-agent system,” “Architecture overview for Research” and “Prompt engineering and evaluations for research agents.” This is a vendor-reported task and evaluation, not an independently reproduced run; the board list and per-agent transcript were not reproduced here.¹⁰

The warehouse and research cases justify separation differently: robots add physical capacity, whereas LLM researchers add concurrent, focused investigations. Both still need integration. Tightly interdependent work may spend the saved time on coordination [Thompson 1967; Kim et al. 2026].

Equal access does not mean equal attention. Suppose every LLM worker may read the same files and use the same tools. Their active contexts can still differ: one investigates a narrow question, while another checks a result without the history that led its author to favor it. This supplies no new knowledge or independent authority. It changes what a bounded model must attend to on a particular call. AutoGen's OptiGuide study provides a closer comparison of that kind of separation than the warehouse or broad-search cases [Wu et al. 2023].

Example

OptiGuide: separating the writer from the checker.

Insight. AutoGen's OptiGuide experiment improved detection of unsafe proposed programs by separating writing from checking within each tested model family, supporting focused review rather than a claim that several agents possess knowledge unavailable to one.

The problem. A coffee supply-chain analyst asks what happens if roasting costs rise by 5%. In the published demonstration, a Writer proposes changes to an optimization program, a Safeguard screens them, and a Commander manages revision and solver execution. The resulting optimal total cost rises from 2470.0 to 2526.5: an increase of 56.5, not 5% of the original total. The answer comes from executing the optimization model, not from the agents agreeing on a plausible number.

What was compared. In a separate experiment, the authors compare one agent performing both writing and checking with the separated arrangement on 100 constructed tasks, half safe and half unsafe. They report higher F1 scores for identifying unsafe programs with both GPT-4 and GPT-3.5-turbo. Each comparison uses the same named backbone; the advantage does not come from supplying the checker with a more knowledgeable model.

Why separation can help. Producing a requested change and challenging its acceptability are different jobs. A focused checker can receive the candidate and its checking criterion without inheriting the writer's whole conversation. That is a plausible attention-management explanation, not an isolated causal finding of this experiment. Merely telling one ongoing conversation to “change hats” need not recreate that information boundary.

What one controller could still do. It could perform the same fresh-check workflow. The study does not establish that the team beats that alternative at matched total cost, or that an LLM's approval guarantees safety. Nor does it audit identical operating-system permissions. The useful lesson is to test separate

10. Jeremy Hadfield, Barry Zhang, Kenneth Lien, Florian Scholz, Jeremy Fox, and Daniel Ford, “How We Built Our Multi-Agent Research System,” Anthropic, 13 June 2025, inspected 20 September 2026. <https://www.anthropic.com/engineering/multi-agent-research-system>. The reported 90.2% improvement concerns the internal research evaluation as a whole. The approximately 15-times token-use comparison is against chat interactions, not the single-agent baseline for this query; neither figure is a controlled estimate of the causal effect of context separation alone.

production and review contexts, not to assume a second role creates a second source of knowledge or an independent security authority.

Source note. Wu and colleagues, AutoGen, Section 3 A4, Appendix D A4 and Appendix E Tables 13 and 15. The coffee example is an optimization demonstration, not a documented factory deployment; the safety comparison uses constructed tasks and older models. No new agent or solver run was performed for this account.¹¹

Local agency connects decisions to information and responsibility. A distant coordinator may lack current observations, permission to inspect a record, or authority to act for another organization. Classical MAS therefore distinguishes distributing a common task from coordinating participants with their own control and interests [Stone and Veloso 2000; Yeoh and Yokoo 2012]. Autonomous local responses can avoid unnecessary escalation, provided the delegated decisions and their limits are explicit. TVA's intermediaries later in this chapter make the human version of this distinction concrete.

Key concept

Distributed agency.

Several actors retain distinct local state, capabilities or decision rights and coordinate their contributions [Stone and Veloso 2000]. The reason for separation should be identifiable: capacity, local knowledge, independent interests or bounded authority. Several roles may share one model; several conversations may still belong to one principal. Model count and legitimate independence are different things.

Separation also makes different kinds of assurance possible. Independent observations can expose a shared mistake; independently controlled privileges can prevent one actor from approving its own act [Saltzer and Schroeder 1975]. But different prompts do not establish independent errors, and a coordinator holding every credential does not acquire independent approval by switching roles. Chapters 7 and 10 develop authority and review respectively. Resilience similarly depends on which failures are isolated, not just on worker count.

These advantages create integration duties: focused memories need a common record of governing facts; local goals need a way to resolve conflicts; parallel outputs need an acceptance procedure. MAST's observations of context loss, misalignment and weak verification show the practical cost of leaving those duties implicit [Cemri et al. 2025]. The following sections examine that cost through incentives, organizational cases and the choice of firm boundaries.

11. Qingyun Wu et al., "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," arXiv:2308.08155v2, 3 October 2023, inspected 20 September 2026. <https://arxiv.org/html/2308.08155v2>. Section 3 A4 reports F1 improvements of 8% with GPT-4 and 35% with GPT-3.5-turbo, without establishing a matched-token comparison against the same isolated review call. Its separate time-saving comparison with ChatGPT and Code Interpreter is confounded by that product's lack of access to the Gurobi dependency; it is not evidence of an advantage under identical tools.

4.4 Incentives and competing claims

An organization is **not only an information network**. Its participants can pursue **different objectives**, and its allocations favor some interests over others. Cyert and March describe firms as coalitions with negotiated goals; Coase asks why activities are coordinated inside firms rather than through market exchange [Cyert and March 1963; Coase 1937]. These perspectives prevent a purely technical account of organization.

Agents need not have human ambition to create an incentive problem. GitLab's postmortem records a concrete resource tradeoff: staging used cheaper storage, whose throughput became a restoration bottleneck when production depended on it.¹² That does not establish conflicting personal motives; it shows how a locally reasonable cost decision can affect another operational objective. An external supplier's capability claim also needs scrutiny when it benefits from winning an award. Conversely, do not assume strategic self-interest merely because a participant is called an agent: inspect its objective, information, and reward structure.

Practice note

A precise score can hide a value judgment.

Insight. GitLab's recovery choices involved different losses and delays, so reducing them to one score would require an explicit judgment about their relative importance.

In the GitLab recovery, snapshot age, completeness, and restoration time were distinct criteria. First exclude options that violate hard constraints. Then compare feasible options on each criterion. Only aggregate them when the decision owner has specified acceptable tradeoffs. A weighted sum's weights express priorities; they are not natural constants.

Preserve criteria, assumptions, and sensitivity to plausible changes. In a recovery decision, uncertainty about restore time or record completeness may change which path is feasible. This is a review method applied to the case, not a claim that GitLab used weighted scores. The method draws on behavioral organization theory and design rationale [Cyert and March 1963; MacLean et al. 1991]; it does not establish a universally optimal decision rule.

12. GitLab, "Postmortem of database outage of January 31," 10 February 2017, especially "Broken recovery procedures," "Recovering GitLab.com," and "Root cause analysis." This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.



4.5 Depth, screening, and the costs of compression

Processing delay

Radner adds the relation between information-processing structure and delay [Radner 1993]. Results about depth in a specified model are not instructions to maximize managerial fan-out: attention, communication capacity, task coupling, and decision rights can change the preferred structure.

Sah and Stiglitz's screening models add a further cost: the mistakes created by the acceptance rule itself [Sah and Stiglitz 1986]. Serial approval tends toward more rejection; alternative acceptance channels tend toward more admission. Which is preferable depends on the loss function and error dependence. Knight Capital's erroneous market orders illustrate why irreversible exposure requires a different acceptance boundary from a reversible internal analysis; the SEC's findings concern inadequate controls before orders reached the market.¹³

 Key concept

Knowledge hierarchies.

Route routine problems to general workers and exceptional problems to scarce expertise [Garicano 2000]. The rationale is conserving specialized knowledge, not assuming that a superior position always knows more. The advantage depends on the task distribution and the costs of learning and communicating.


 Key concept

Screening architectures.

Requiring every screen to approve and allowing any channel to approve trade false acceptance against false rejection in different ways [Sah and Stiglitz 1986]. The preferred rule depends on losses and error dependence. Adding reviewers is not a neutral increase in safety; it changes which mistakes the organization tends to make.

 **Caution – More screens are not automatically more independent evidence.** Human reviewers can share assumptions, and different model families can share errors or sources. Estimate the effect of an additional screen on false acceptance, false rejection, time, and cost. Independence must be established for the relevant errors; it does not follow from different names, employers, prompts, or model vendors.



Information-
processing
design
Task
interdependence

Two studies put these explanations to work. The paint factory illustrates Thompson's interdependence and the limits of separate local optimization: the same staffing choice changes production capacity, inventory and later costs. TVA illustrates Galbraith's capacity-building through lateral relationships, but adds Selznick's institutional question: whose interests enter with that capacity? The first case asks **which decisions must be solved together**; the second asks **whose cooperation and judgment the organization relies on**.

13. U.S. Securities and Exchange Commission, "SEC Charges Knight Capital With Violations of Market Access Rule," 16 October 2013, release 2013-222; links to administrative order 34-70694. Knight consented without admitting or denying the findings. <https://www.sec.gov/newsroom/press-releases/2013-222>.

 More detail

The paint factory: deciding production and staffing together.

Insight. A cheaper staffing decision can create greater production or inventory costs, so the factory's model treats these choices as one interdependent plan.

What is being organized? Holt, Modigliani, Muth, and Simon studied an anonymous paint company and analyzed six years of its production and workforce decisions. Orders arrive from distributors and dealers, workers produce paint, and inventory connects today's production to later sales. Management must choose how much to produce and how many workers to employ while demand changes [Holt et al. 1960, ch. 1, pp. 16–25].¹⁴

Why are the decisions coupled? Following each change in orders immediately can require overtime, hiring, or layoffs. Keeping production steadier can avoid some of those adjustments but accumulate inventory when orders are low, or leave back orders when orders exceed available stock and output. The study's decision model therefore connects payroll, overtime, hiring and layoff costs, inventory, and back orders. The important object is not an isolated production target: it is a sequence of resource decisions and their consequences over time.

What did the researchers do? They constructed decision rules and compared their implied operation with the historical record through retrospective simulation. This asks what the model would have recommended under specified information and cost assumptions. It is different from observing managers adopt the rule, cope with its exceptions, and achieve the predicted result in a live factory. In the anatomy plate, the researchers' model therefore sits outside the operating organization rather than above it as an installed manager.

Why do the forecasts matter? The company's original forecasts were not recorded. The researchers had to compare alternative forecast assumptions, including a perfect forecast constructed with hindsight. A rule given future demand can look better than one that had to decide under uncertainty. The informative comparison is therefore conditional on the information supplied; simulated savings cannot simply be credited to the decision rule alone or described as savings the factory actually realized.

What should a designer take from it? A useful organization model makes cross-unit and cross-time consequences visible. A staffing decision changes production capacity; production changes inventory; inventory affects what can be delivered later. For an artificial organization, analogous dependencies must survive task decomposition. A staffing specialist, a production planner and an inventory manager need a shared plan that accounts for these effects. Merely giving each a local target misses the mechanism the study models. A central planning method can also supply that integration: the case supports joint decision making, rather than a requirement for three autonomous planners.

What remains outside the model? A cost representation does not settle fairness to workers, lawful practice, or which service commitments are acceptable. Nor does a retrospective comparison prove deployability. Before using a similar model, ask which costs, constraints, and information were represented, which were omitted, and who is entitled to choose the tradeoffs. The study matters because it exposes the decision system behind a simple factory chart, not because it establishes one universally efficient form of organization.

14. C. C. Holt, F. Modigliani, J. F. Muth, and H. A. Simon, *Planning Production, Inventories, and Work Force* (Prentice-Hall, 1960), Chapter 1, especially pp. 16–25. Open source scan: <https://archive.org/details/PlanningProductionInventoriesAndWorkForce>. The plate is an original reduction, not a copied figure.

In the Tennessee Valley Authority's agricultural programme, the central problem is reaching and working with farmers through institutions that already have expertise and local standing. Their knowledge and relationships make cooperation useful; they also give the intermediaries influence. The contrast with the factory is important: these participants are not simply variables in one manager's plan.

 More detail

TVA: gaining local reach without controlling every intermediary.

Insight. The partnerships that gave TVA access to farmers also gave its intermediaries influence over which farmers and interests the public programme served.

What is the setting? The Tennessee Valley Authority (TVA) is a United States public agency. Selznick's *TVA and the Grass Roots* examines its organization through field research; the part used here concerns its historical agricultural programme. Fertilizer test-demonstration work connected technical activity with changes on participating farms [Selznick 1949, chs. III–IV].¹⁵

How did the programme reach farmers? TVA worked through existing land-grant institutions and agricultural extension services. Cooperative agreements and reimbursed personnel connected the agency to people with expertise and local relationships. County extension agents linked this network to farmers; committees and meetings contributed to participation and selection. This was a cooperation network, not simply a chain of TVA employees receiving commands.

What did participation involve? The source describes test-demonstration farmers changing farm plans, keeping records, maintaining check plots, paying freight, and allowing visits. Farmers were therefore participants in producing and communicating an agricultural demonstration, not passive endpoints of an information-delivery system. Their practical observations and records contributed to the work, while selection procedures determined which farmers took part.

Where did influence enter? A programme cannot reach "the community" in the abstract: particular institutions and people select participants, interpret needs, and mediate access. Selznick describes farmer selection as centred in practice on the extension official, modified by committees and meetings (pp. 130–132). Those arrangements made delivery possible while giving existing local relationships a role in shaping whose experience and interests entered the programme. Participation is therefore a question to examine, not a synonym for representation of everyone affected.

What is cooptation here? Selznick's account concerns how incorporating outside groups and interests into an institution's arrangements can secure cooperation while creating commitments and altering its practical direction. The point is not that every partner acted improperly. It is that partners supplying indispensable reach and legitimacy are not neutral transmission channels. An agency's formal purpose and reporting chart cannot, by themselves, explain the influence exercised through these relationships.

What follows for organizational design? Cooperation can buy reach and local knowledge without absorbing every participant into a hierarchy. It also requires scrutiny of representation: who chooses participants, who can question the intermediaries, and whose needs remain unheard? This extends the chapter's information-processing account into an account of authority and interests.

15. Philip Selznick, *TVA and the Grass Roots: A Study in the Sociology of Formal Organization* (University of California Press, 1949), Chapters III–IV; organizational charts pp. 106–110, programme relationships pp. 124–145, farmer participation pp. 130–132. Open source scan: <https://archive.org/details/in.ernet.dli.2015.286>. The plate selects agricultural relationships, not the whole agency.

Why care when building artificial organizations? External services, knowledge providers, and operational partners can similarly supply capability while influencing what the organization sees and can do. The useful questions are who selects inputs and participants, whose interests become represented, what dependence is created, and how the arrangement can be challenged or revised. These are proposed contemporary applications of an institutional analysis, not a claim that software providers behave identically to agricultural partners.

② Example

A demonstration plot makes a claim visible.

Insight. A demonstration plot makes a treatment claim inspectable, but whose farms become demonstrations is a separate choice that shapes whom the programme represents.



Historical photograph H2. A TVA test field, 1942: a person examines vegetation between plots labeled untreated and treated with phosphate and lime. The contrasting growth and signs show how a field demonstration communicates a claim. FDR Presidential Library and Museum, 27-0921a.

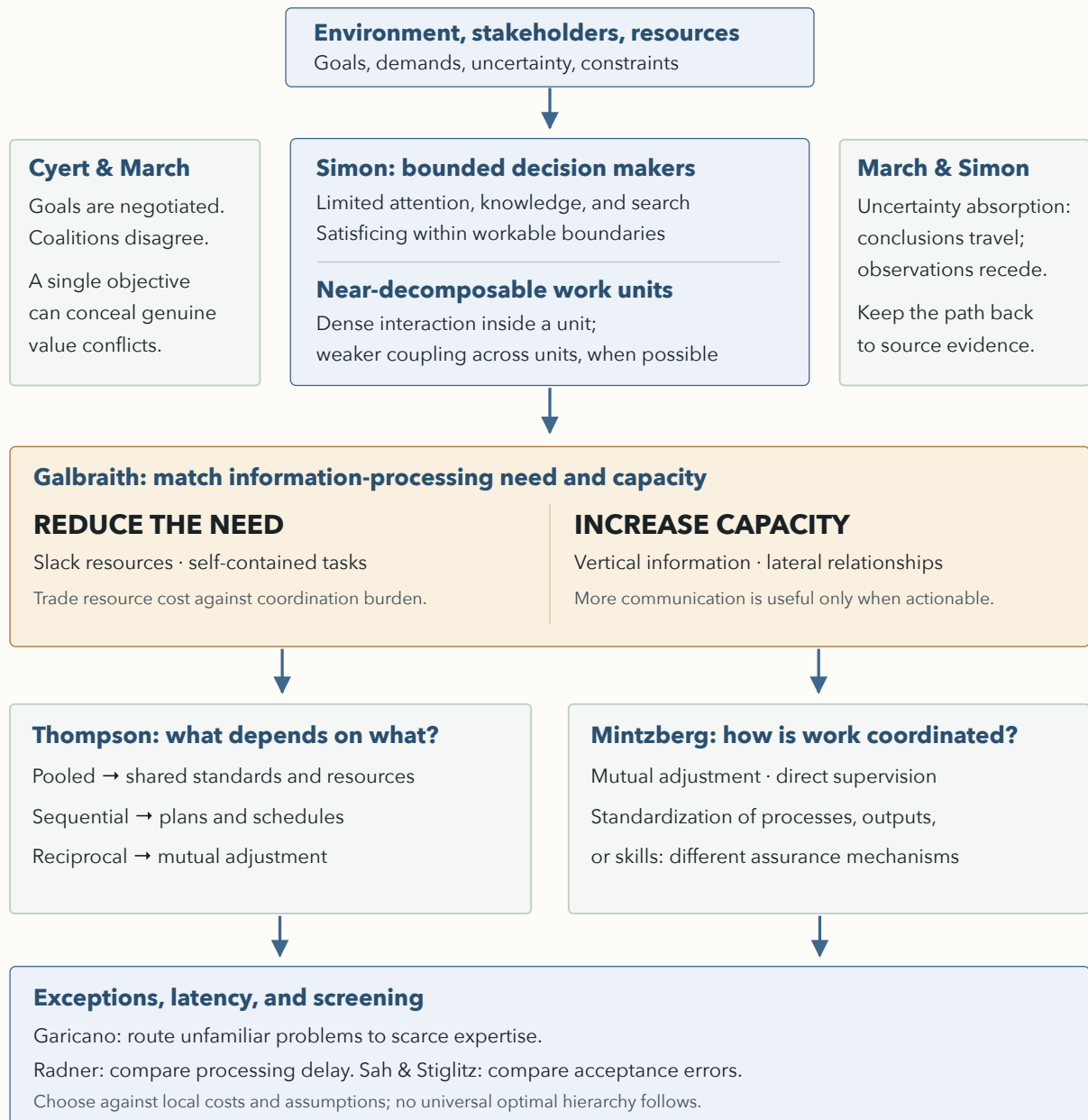
The labeled plots turn an abstract treatment claim into a comparison that visitors can inspect. Demonstration work connects research, farmers' practical judgments, and the evidence used to recommend a change. The archive attributes the contrasting growth to fertilizer treatment.

Source note. This archive photograph illustrates test-demonstration work; its location is not established as a Selznick field site. The image does not independently establish treatment effects or representative participation. [Image source and reuse terms.](#)

② **How to read Synthesis plate I.** Begin with the central problem of bounded decision making. The side panels explain conflicting objectives and what gets lost in communication. Follow the downward arrows to Galbraith's design options, then Thompson's dependencies and Mintzberg's coordination mechanisms. The final band adds expertise routing, latency, and screening error. **The arrows organize the explanation; they are not a literal org chart or a causal estimate.** The plate is an editorial synthesis of the sources developed in Sections 4.1–4.6, not a model jointly proposed by the named authors.

ONE ORGANIZATION • SEVERAL EXPLANATORY LENSES

Editorial synthesis of the cited approaches, not a model jointly proposed by these authors.



Blue arrows: explanatory reading sequence, not literal reporting lines or causal estimates.

Synthesis plate I. One organization viewed through complementary theories.

ORGANIZATIONAL ANATOMY • TWO EMPIRICAL STUDIES

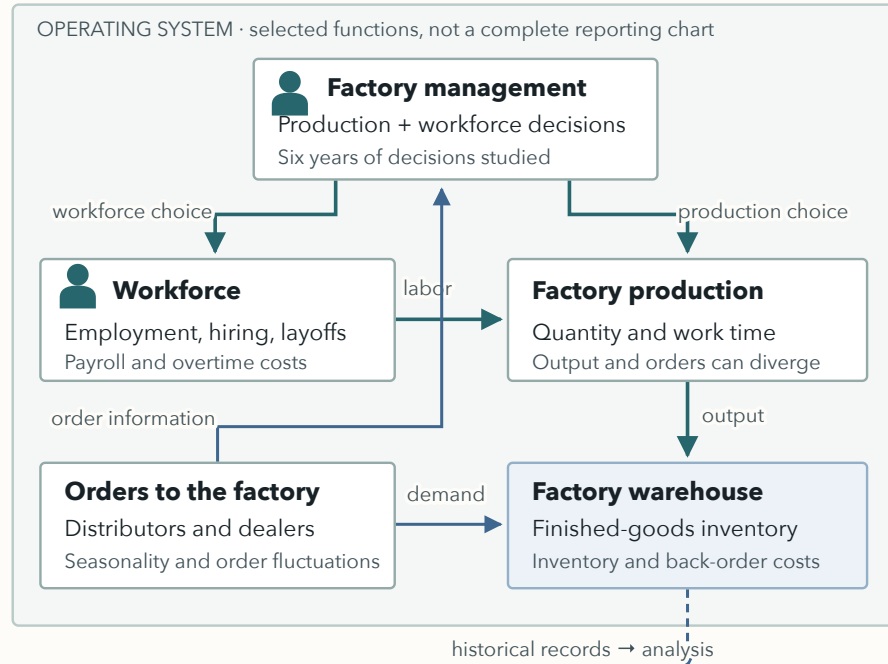
Actual organizations in the literature: an internal decision system and a cross-institutional programme.

→ Operating relationship → Information / evidence - - - - -> Research model, not installed control - - - - -> Interpreted influence

A • THE PAINT FACTORY

Holt, Modigliani, Muth & Simon (1960)

Six years of operating decisions at an anonymous paint company.



Research team • retrospective decision-rule simulation

Inventory + workforce + demand forecasts → proposed decisions
 Compared historical operation with rules using two forecast assumptions.
 Original forecasts were not recorded; hindsight is not live knowledge.
 Do not read simulated savings as measured savings from deployment.

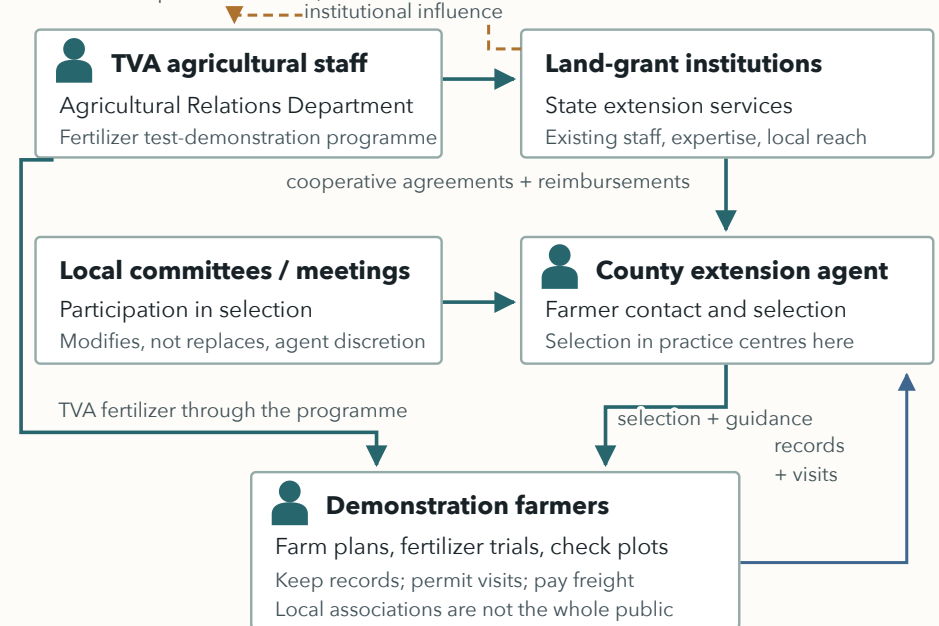
Source: Planning Production, Inventories, and Work Force, ch. 1, pp. 16-25.

Original analytical drawings from the cited studies. Arrows identify selected relationships; no unreported job titles, headcounts, or AI workers are added.

B • TVA'S AGRICULTURAL PROGRAMME

Philip Selznick (1949) • field study of an actual public agency

Selected cooperation network; not today's TVA and not its entire structure.



Selznick's interpretation • cooptation and constituency

Partners supply capacity, but their interests can shape the agency.
 Existing extension institutions and farm leadership are not neutral channels.
 Informal influence is different from a formal command relationship.
 Participation must be examined: who selects, who benefits, who is absent?

Source: TVA and the Grass Roots, chs. III-IV, especially pp. 104-145.

Synthesis plate II. Actual organizations in the literature: the paint factory studied by Holt, Modigliani, Muth, and Simon, and Selznick's TVA agricultural programme.

 Key concept

Capacity comes with organizational commitments.

The factory panel connects staffing, production and inventory in one plan because a local saving can create costs elsewhere. The TVA panel connects local reach with the partners' influence over which farmers and interests the programme serves. An intermediary supplies capacity and can shape its use. These studies are not interchangeable evidence for one ideal hierarchy [Holt et al. 1960; Selznick 1949].

 **How to read Synthesis plate II.** Compare how decisions travel through **two documented organizational settings**. The left panel shows a factory's management, workforce, production, inventory, and incoming orders. The right shows an agricultural programme delivered across TVA, land-grant institutions, extension personnel, local committees, and participating farmers. Human symbols identify people or groups of people, not AI workers. Solid lines represent selected operating or information relationships. Dashed blue lines identify the researchers' modelling activity; dashed amber identifies Selznick's interpretation of institutional influence. Neither panel claims to reproduce a complete formal reporting chart.

4.6 Recurring case: an interface that did not carry shared meaning

Insight. Mars Climate Orbiter shows that exchanging data successfully is insufficient when the teams' checks fail to preserve its physical meaning.

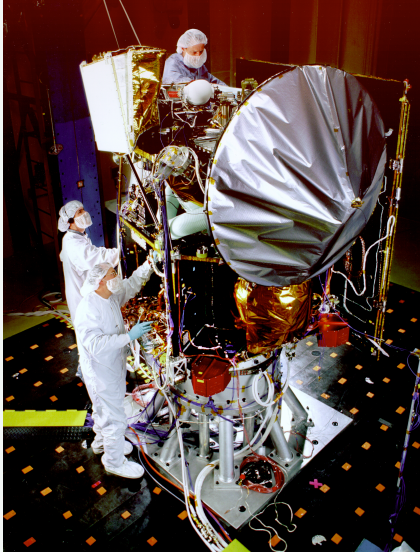
Mars Climate Orbiter's spacecraft and navigation teams exchanged information across an organizational boundary, but the unit mismatch was not caught. NASA/JPL's preliminary account identifies the failure of checks and systems engineering as central.¹⁶ The lesson is not that all information must reach one director: it is that an accepted handoff must preserve the meaning and verification obligations needed by its recipient.

16. NASA/JPL, "Mars Climate Orbiter Team Finds Likely Cause of Loss," 30 September 1999, release 99-083. This is a preliminary institutional account, not the final investigation. <https://www.jpl.nasa.gov/news/mars-climate-orbiter-team-finds-likely-cause-of-loss/>.

② Example

The physical system behind the interface.

Insight. A successful hardware test and a correct data handoff establish different requirements, so evidence for one cannot replace verification of the other.



Historical photograph H1. Technicians with Mars Climate Orbiter during acoustic testing, 27 May 1998. Flight hardware, instruments, and test equipment bring the system's physical interfaces into view. NASA, GPN-2000-000498.

Reliable operation depends on matching each requirement to the right check. Launch-environment testing examines how the spacecraft withstands physical conditions; navigation-data checks examine whether teams interpret transferred values consistently. Passing one kind of test leaves the other to be established.

The photograph records prelaunch acoustic testing, not the later navigation failure. [Image source and reuse terms.](#)

The broader design question is what information must survive a handoff. For a measurement, that includes units and interpretation. For a recommendation, it includes the evidence and qualifications behind the conclusion. Forwarding every transcript is rarely a useful answer. **[HEURISTIC]** Use **loss-accounted compression**: link a summary to its evidence, preserve unresolved objections and dissenting views, and state what was omitted. This is the book's design synthesis, not a diagnosis that summarization caused the spacecraft's loss.

💡 Key concept

Loss-accounted compression.

A useful summary reduces reading effort while preserving a route to its evidence, omissions, and unresolved disagreement. This book's design heuristic asks what a summary loses and how that loss can be inspected; it is not a named theorem from Simon or a claim that every detail must remain in the executive view.

4.7 Why firms sometimes beat the open market

Why not buy every contribution from an independent supplier? Coase's answer is that **using the price mechanism itself has costs**: finding capable suppliers and relevant prices, negotiating terms, checking performance, adapting agreements, and resolving disputes [Coase 1937]. A firm can replace repeated bargaining over each adjustment with continuing arrangements that authorize some decisions to be made administratively. That does not eliminate contracts or give managers unlimited authority. It changes how a defined range of decisions is coordinated. Markets, too, depend on institutions and enforceable rules; the contrast is not organization versus disorder.¹⁷

Key concept

Transaction costs and firm boundaries.

Compare the cost of obtaining the same useful result through different arrangements, not the supplier's price against a supposedly free internal instruction. Internal organization saves some search, bargaining, and adaptation costs but adds administration, monitoring, decision errors, and incentive problems. Coase's boundary question is marginal: would the next activity be coordinated more economically inside this firm, through the market, or by another firm? There is no implication that all activity belongs in one large hierarchy.

Williamson makes the comparison more specific. **Asset specificity** means that an investment loses value outside a particular relationship. Once parties depend on such investments, switching partners may cease to be a credible response to disagreement. Contracts cannot cheaply anticipate and govern every future contingency. Renegotiation can then create **hold-up**: one party uses the other's dependence to seek better terms after the investment has been made. Internal coordination may make joint adaptation less costly, particularly for recurring, nonstandard transactions [Williamson 1979]. But hierarchy introduces its own bargaining, weak incentives, and opportunities to misuse authority. A long-term contract or partnership may outperform either repeated spot purchases or full integration.

Grossman and Hart explain another part of the boundary: **residual control rights**, meaning rights to decide how assets are used where contracts leave decisions unspecified [Grossman and Hart 1986]. Changing ownership changes the parties' bargaining positions and therefore their incentives to make investments that cannot be fully specified or enforced in advance. Integration can strengthen one party's investment incentive while weakening another's; ownership is not a costless cure for incomplete contracts. This is an account of asset control, not ownership of people or a claim that every manager knows best.¹⁸

17. Ronald H. Coase, "The Institutional Structure of Production," Nobel lecture, 9 December 1991. Coase explains the market-coordination costs and the comparison with internal organization and other firms. <https://www.nobelprize.org/prizes/economic-sciences/1991/coase/lecture/>.

18. Royal Swedish Academy of Sciences, *Contract Theory*, public information for the 2016 Prize in Economic Sciences, especially "Incomplete contracts" and "Property rights." This accessible account explains the investment-incentive tradeoff in Hart's research. <https://www.nobelprize.org/prizes/economic-sciences/2016/popular-information/>.

Arrangement	When its advantages matter	Costs that remain
Market purchase	Outputs are specifiable, quality is checkable, alternatives are credible	Search, contracting, verification, and switching
Long-term contract or partnership	Repeated exchange needs safeguards and adaptation, but supplier autonomy remains valuable	Negotiating changes, monitoring, and resolving contractual gaps
Internal organization	Interdependent, relationship-specific work needs frequent joint adjustment	Administrative overhead, weaker incentives, internal conflict, and managerial error

This table summarizes selection considerations, not a universal ranking. Compare feasible arrangements at comparable quality and risk. Efficiency also does not by itself establish legality, fairness, or acceptable effects on outsiders.

Example

Coal mines and power plants.

Insight. Greater dependence between a mine and a power plant can make protected long-term supply or common ownership more valuable than repeated market purchases.



Historical photograph H3. Trucks hauling coal from Navajo Mine to Four Corners Generating Plant, May 1972. Trucks, transport route, and generating plant reveal the physical connection behind a continuing supply relationship. Lyntha Scott Eiler, EPA/DOCUMERICA; NARA 544169.

The plant needs continuing fuel deliveries, while the mine needs a buyer for its output. Coal-sector research examines how such relationships are governed [Joskow 1985]. A mine and a nearby power plant may depend heavily on one another when alternative buyers or suppliers are distant. The Royal Swedish Academy's review describes more extensive contracting or integration where that dependence is greater, and simpler market arrangements where alternatives are readily available.¹⁹ **When would integration help?** If the plant depends on this mine, the mine has few alternative buyers, and frequent unforeseen adjustments are costly to negotiate, common ownership may protect relationship-specific investment and make adaptation easier. This is Williamson's mechanism: dependency changes the cost of governing an exchange, even if extracting coal remains the same task.

When would contracting be better? A workable long-term contract can protect investments while retaining supplier incentives; accessible alternative mines or buyers can make market exchange sufficient. Integration is warranted only if its adaptation gains exceed its administrative and incentive costs. Proximity alone does not decide the issue [Joskow 1985; Williamson 1979].

19. Royal Swedish Academy of Sciences, *Economic Governance: The Organization of Cooperation*, public information for the 2009 Prize in Economic Sciences, especially "Markets versus hierarchies," "Efficient conflict resolution," and "Mutual dependence behind hierarchical organizations." The coal-sector illustration here follows that review; no new analysis of plant-level data was performed. <https://www.nobel-prize.org/prizes/economic-sciences/2009/popular-information/>.

Source note. The photograph records the Navajo Mine–Four Corners supply connection; it establishes neither common ownership nor the best governance for these firms, and does not identify them as members of Joskow's sample.

[Image source and reuse terms.](#)

For artificial organizations, cheap model calls are not the whole comparison. A market of external agents may reduce search and access costs while leaving verification, confidential context, policy alignment, continuity, and liability expensive. Stable internal roles may repay those costs for repeated, tightly coupled work; interchangeable services with clear contracts and testable outputs may be better procured. This is an engineering application of the theories, not evidence that internal agents outperform suppliers. Measure the entire governed transaction, including coordination and recovery, rather than token price alone.

4.8 When fewer agents suffice

The benefits developed above supply the test for adding a team. If the work needs no additional capacity, specialized information or separate authority, one capable agent may handle its stages in turn. Solo Performance Prompting demonstrates multi-persona collaboration within one LLM; Self-Consistency generates and aggregates alternatives; MemGPT maintains information across memory tiers [Z. Wang et al. 2024; X. Wang et al. 2023; Packer et al. 2023]. Changing hats is therefore a real design option, not an inadequate baseline.

One model, one context and one principal are different things. A controller can schedule focused contexts and specialist tools without turning every stage into an autonomous actor. Prefer this simpler arrangement when dependencies are predictable, a common owner can legitimately supply the necessary information, and additional handoffs do not improve the outcome. Preserve the organization's controls even when there is only one worker.

The “all-powerful single agent” objection therefore has a precise answer. If one controller can reproduce every worker's history, model call, observation and action with the same available resources, it can emulate the team's computation. There is no extra computational capability supplied merely by calling the parts agents. Actual systems remain bounded: keeping several focused contexts may work better than one growing conversation. But a comparison against that conversation does not establish superiority over every single-controller design.

Empirical comparisons reinforce this conditional choice. Strong-prompt controls can match multi-agent discussion on tested reasoning tasks [Q. Wang et al. 2024]; Gao and colleagues find diminishing team advantages with stronger models and propose adaptive routing [Gao et al. 2025]. Conversely, layered synthesis can benefit from complementary model outputs [J. Wang et al. 2024]. Kim and colleagues' 260-configuration study finds gains on decomposable financial reasoning and losses on sequential planning [Kim et al. 2026, v3]. The tested tasks, budget controls and model generations limit transfer; they do not establish one ideal architecture or agent count.

Frameworks such as AutoGen can still be useful because they make context management, message routing, feedback and reusable workers easier to implement [Wu et al. 2023, Sections 2 and 4]. Their utility does not depend on proving that one controller is incapable of the task. Also distinguish an LLM reasoning worker from a tool executor: AutoGen's default assistant-and-executor pair can contain only one LLM-backed decision process. Counting framework objects is not counting independent problem solvers.

 Practice note

Compare mechanisms at comparable cost.

Test a strong single worker, isolated role contexts, independent samples with aggregation, and the communicating team on the same held-out work. Include tools, memory, repair, managers and reviewers in the resource accounting. Compare outcome quality, elapsed time, total spending and human review; equal model-call counts need not mean equal cost [Kapoor et al. 2024]. For independent principals, do not give the central baseline information or authority it could not possess.

Local information can also make coordinated planning harder: the classical finite-horizon DEC-POMDP problem is NEXP-complete under its stated assumptions [Bernstein et al. 2000; Oliehoek and Amato 2016]. Distribution addresses real constraints; it does not abolish them. The practical conclusion is to retain the least elaborate arrangement that supplies the necessary capability and governance. The next chapter turns from choosing that arrangement to defining the roles and obligations within it.

4.9 Further afield

Top recommended readings

Direct foundations, ranked for explaining why organization is useful and why it also creates costs.

1. Simon (1947), *Administrative Behavior*.

The chapter's starting point is that real decision makers have limited attention, information, and computation. Simon made these limits central to organizational analysis rather than incidental departures from perfect choice. Read for how arrangements shape decisions; the later satisficing model develops this program, but neither work licenses arbitrary quality thresholds in an engineered process.

2. March and Simon (1958), *Organizations*.

This behavioral synthesis develops how organizations process information and coordinate constrained participants. Its treatment of uncertainty absorption is especially important here: conclusions can travel while their observations disappear. Follow that mechanism before designing summaries and escalation chains. The book's loss-accounted compression rule is a subsequent design synthesis, not the authors' named theorem.

3. Coase (1937), *The Nature of the Firm*.

Coase's question explains why coordinated activity is not always purchased through repeated market exchanges. Search, bargaining, adaptation, and internal administration belong in the comparison. Read the boundary argument as a marginal choice among arrangements, not a celebration of bigger hierarchies. It supplies the economic counterweight to assuming that more internal agents and managers are costless.

Broader explorations

Beyond the market-or-firm choice. Ostrom's work on common-pool resources examines how users organize rules, monitoring, and cooperation themselves [Ostrom 1990]. The Royal Swedish Academy's account includes irrigation systems in Nepal, where changing infrastructure also changed incentives for maintaining the shared system.²⁰ This widens the question from ownership alone to the arrangements that sustain cooperation.

Biography

Elinor Ostrom (1933-2012).



Elinor Ostrom at a press conference in Stockholm, 7 December 2009.

Ostrom was an American political scientist trained at UCLA who built much of her career at Indiana University. Her field-based research examined how people devise rules for shared resources such as water, forests, and fisheries. It challenged the assumption that only privatization or centralized control could avert resource depletion.

She helped establish the Workshop in Political Theory and Policy Analysis with Vincent Ostrom and received the 2009 economics prize for her analysis of economic governance, especially the commons. Her importance here is methodological as well as substantive: examine actual arrangements and local conditions before prescribing one institutional solution.²¹

[Portrait credit.](#)

Could several independently governed agent teams share an evaluation service, knowledge collection, or scarce computational resource under jointly maintained rules? That is a research direction, not evidence that a digital commons will govern itself. Compare who can amend the rules, who observes misuse, and whose costs are invisible. The human cases are valuable precisely because they contain institutional detail that a generic "decentralized" label omits.

20. Royal Swedish Academy of Sciences, *Economic Governance: The Organization of Cooperation*, public information for the 2009 Prize in Economic Sciences, especially "Markets versus hierarchies," "Efficient conflict resolution," and "Mutual dependence behind hierarchical organizations." The coal-sector illustration here follows that review; no new analysis of plant-level data was performed. <https://www.nobel-prize.org/prizes/economic-sciences/2009/popular-information/>.

21. Nobel Prize Outreach, "Elinor Ostrom – Facts," and her biographical account, inspected 16 September 2026. <https://www.nobel-prize.org/prizes/economic-sciences/2009/ostrom/facts/>; <https://www.nobelprize.org/prizes/economic-sciences/2009/ostrom/biographical/>.

One agent or many: start with the structure. Stone and Veloso (2000, Section 2) give the classical reasons for MAS and their limits. Read Guestrin, Koller and Parr (2001) beside Dechter (1999) to distinguish factored computation from agent count; Fioretto and colleagues (2018) make message size, memory and privacy costs explicit. Oliehoek and Amato (2016, Chapter 8) connect these methods to decentralized planning. Then compare Solo Performance Prompting, Q. Wang's strong-prompt controls and Kim's version-3 scaling study. Wooldridge and Jennings (1998) supply the engineering warning: agent abstractions do not remove the difficulties of decomposition and distribution.

Shared interests do not automatically produce collective action. Mancur Olson's *The Logic of Collective Action* examines why members of a group may have difficulty obtaining a common benefit despite valuing it [Olson 1965]. It adds incentives and participation to the information-processing explanation of organization. Ask who pays the cost of monitoring, maintenance, or evidence collection when everyone benefits from the result. In a human-agent setting, the incentive problem may sit with owners, operators, or service providers rather than with a model itself. Do not assume that a shared mission statement or a list of rational participants resolves the distribution of effort.

Knowledge is dispersed in time and place. Hayek's "The Use of Knowledge in Society" contrasts knowledge actually distributed among people with the fiction of all relevant information being given to one planner [Hayek 1945]. His discussion of prices as coordinating signals is useful beside the chapter's account of firms and markets. What local knowledge would be lost by centralizing a decision, and what additional information does the local actor need? The question can inform comparisons of centralized agent orchestration and local execution. It does not establish that a price mechanism solves every governance, quality, or responsibility problem, or that all decentralization is beneficial.

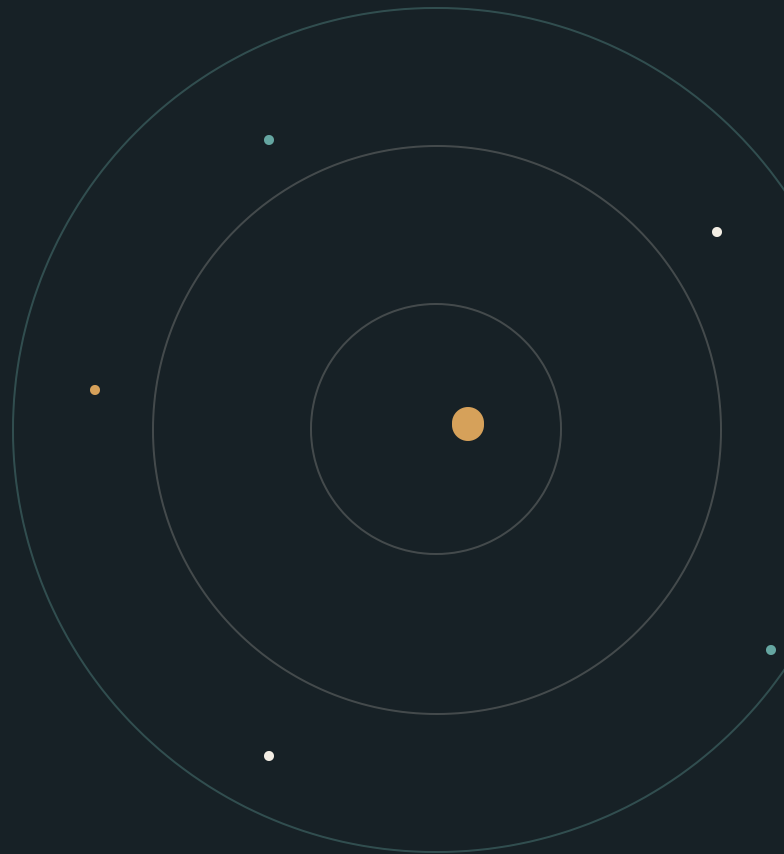
Leaving and speaking are different corrective mechanisms. Albert Hirschman's *Exit, Voice, and Loyalty* studies responses to deterioration in organizations and other arrangements [Hirschman 1970]. An operator may switch a supplier, challenge a decision, or remain while seeking improvement. These are not equivalent signals. For agent services, inspect whether dissatisfied users can leave with their records, make a meaningful complaint, or revise an arrangement. A low complaint count may reflect blocked voice rather than satisfaction. Hirschman's framework suggests comparing the available routes of correction; it does not make exit costless or turn every expression of dissatisfaction into reliable evidence about performance.

Legibility can simplify away the wrong things. James C. Scott's *Seeing Like a State* examines large schemes that depend on making complicated social settings administratively legible [Scott 1998]. Read it as a challenge to the assumption that a cleaner organizational dashboard necessarily represents work better. Which local distinctions disappear when activities become standardized fields and targets? A useful study would compare what the central record says with what people need to keep the operation functioning. Scott's historical argument is not a prohibition on measurement or planning. It asks us to examine the losses and power relations introduced by the simplification itself.

PART II

Institutional mechanics

Roles, coordination, delegation, authority, attention, policy, and the machinery that makes them real.



5 Roles, missions, and obligations

© **Core point — Roles outlive workers.** Bind duties to durable positions and missions, then record their occupants and enactment intervals. Replacement is safe only when responsibility, access, and unfinished work transfer explicitly.

5.1 The problem: identity is not responsibility

An agent instance is a **runtime actor**. A **role** is an institutional position with expected competences, permissions, obligations, and relationships. If the organization says “model instance 8 owns tax filing,” replacing that instance can erase responsibility. If it says “the Treasurer role is obliged to file, and agent 8 currently enacts Treasurer,” the obligation survives replacement.

The Agent-Group-Role (AGR) model gives a minimal vocabulary [Ferber and Gutknecht 1998]. MOISE+ adds three linked specifications [Hübner et al. 2002a, 2002b]:

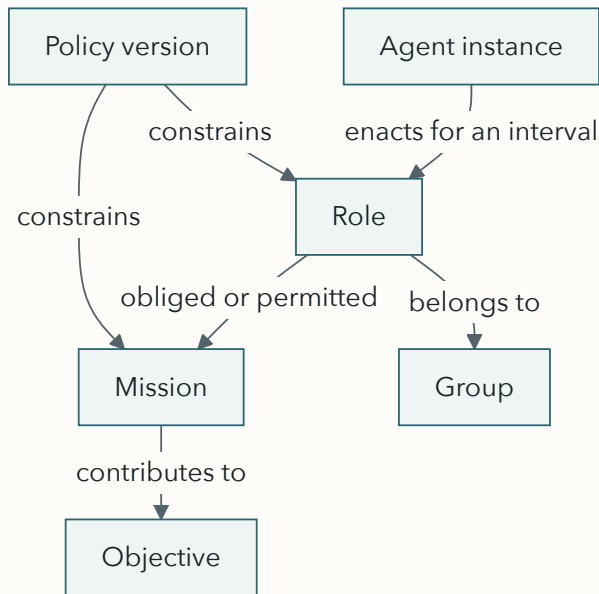
1. **Structural:** roles, groups, links, inheritance, cardinality, compatibility.
2. **Functional:** goals decomposed into schemes and missions.
3. **Deontic:** obligations and permissions binding roles to missions.

Use first-class roles when work is **long-lived**, agents are **replaceable**, duties must survive restarts, incompatible duties require separation, or auditors must ask who was accountable. GitLab's postmortem explicitly identifies missing ownership for regular recovery testing and proposes a data-durability owner.²² This supplies a real obligation to model; merely naming the role does not prove that its tests will run or that its holder has the necessary authority.

② **How to read Figure 1.** Follow the solid chain from an ephemeral agent to a durable objective. The role is the hinge: agents may change, while mission and obligation remain. Policy constrains both who may act and what the mission may do. The conclusion is that a prompt persona is not a role unless the surrounding system enforces this structure.

22. GitLab, “Postmortem of database outage of January 31,” 10 February 2017, especially “Broken recovery procedures,” “Recovering GitLab.com,” and “Root cause analysis.” This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.



FIGURE 1

5.2 When this family is a strong choice

Implementation needs **stable identifiers**, **enactment intervals**, explicit cardinalities, compatibility constraints, and **versioned role definitions**. Avoid embedding these only in natural-language prompts; **validate them before dispatch**.

Variants. Small systems can use a typed role table rather than a full organizational language. Dynamic teams can instantiate temporary roles with expiry. Skills should remain separate from roles: capability answers *can this agent do it?*; role answers *is this agent institutionally positioned to do it?*

Failure modes. Role explosion creates **synthetic bureaucracy**. Persona labels can disguise identical agents as independent experts. A role without **enforced authority or obligation** is decorative. Over-rigid compatibility rules make exceptional staffing impossible.

Reject this technique for a one-off, low-risk workflow with no persistent responsibility. Name the responsible function and use ordinary access control.

5.3 Worked case: ownership of recovery tests

Insight. GitLab identified both ownership and recovery testing as missing safeguards, showing that assigning responsibility and implementing a check solve different parts of the problem.

GitLab's improvement list distinguishes assigning ownership for data durability from automating recovery tests. These are not alternative descriptions of the same fix: one identifies responsibility and the other performs a mechanism.²³ The organization needs both a functioning test and someone accountable for its continued applicability, failures, and review.

Practice note

Replacement does not cancel a duty.

The obligation to test recovery should persist when an on-call assignment or employee changes. Applied to agent-assisted operations, retain the duty, record occupancy intervals, identify unfinished tests and live attempts, and transfer the authority needed to continue or cancel them. These are proposed handoff requirements, not a reconstruction of GitLab's staffing changes.

The role/agent distinction comes from organization-oriented MAS [Ferber and Gutknecht 1998; Hübner et al. 2002a, 2002b]. This handoff is an implementation heuristic. Test replacement during a pending recovery test, not just while idle. A durable role name alone does not prevent duplicate execution.

5.4 Role links, landmarks, and participation contracts

Key concept

Role, occupant, and duty.

A role defines an institutional position; enactment records who occupies it; a deontic relation binds a role to a mission [Ferber and Gutknecht 1998; Hübner et al. 2002b]. Changing an occupant need not change the duty. Model those changes separately so replacing a worker cannot silently erase responsibility.

Concept reference – specification is not occupancy. Use these records to separate organizational design from the agents currently carrying it out [Hübner et al. 2002a, 2002b; Dignum 2004].

Concept	What it specifies	Concrete question
Group	Context within which roles and relationships apply	Does this permission hold in this project or another one?
Enactment	Which actor fills which role during an interval	Who was responsible at the time of the action?
Cardinality	Required or permitted number of occupants	Can two actors occupy this position simultaneously?

23. GitLab, "Postmortem of database outage of January 31," 10 February 2017, especially "Broken recovery procedures," "Recovering GitLab.com," and "Root cause analysis." This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.

Concept	What it specifies	Concrete question
Compatibility	Which roles can be jointly occupied	Can the proposer also be the final reviewer?
Authority link	A direction of organizational control	Who may issue or monitor this work?
Deontic link	Duty or permission connecting a role with a mission	What is the role actually required or permitted to achieve?
Landmark	Required state without fixing every intermediate action	Must evidence be accepted, or must one exact procedure be used?

The [MOISE XML example box](#) shows several of these distinctions as separate declarations.

AGR

AGR separates an agent from the roles it occupies in organizational groups [Ferber and Gutknecht 1998]. A role's scope matters: permission in one group should not silently become permission in every other group an actor joins. Represent group identity and enactment interval in authorization checks rather than treating role names as global credentials.

MOISE+

MOISE+ distinguishes structural relationships such as **authority**, **communication**, and **acquaintance**, alongside the functional and deontic specifications [Hübner et al. 2002a, 2002b]. Communication identifies an allowed or expected interaction channel; acquaintance represents knowledge of another role; authority represents a direction of organizational control. A right to issue or monitor work is not itself the obligation to perform that work. Obligations are specified separately by the **deontic relation to missions**.

🔗 **Design implication – Keep authority and duty separately editable.** A reviewer may be allowed to request a correction without being obliged to do so in every case. A specialist may have a duty to report delay without having authority over the recipient. If every authority edge automatically creates a duty, the model cannot express either situation faithfully.

OperA

OperA adds an organizational model, participation arrangements, and concrete interaction agreements [Dignum 2004]. **Landmarks** describe required states rather than prescribing every intermediate action. In the recovery example, "recovery evidence accepted" can be a landmark reached through different valid investigations. This is useful when implementations vary but institutional obligations must remain stable. Temporary occupancy and negotiated participation still need explicit limits; landmark flexibility does not authorize arbitrary means.

💡 Key concept

Landmarks.

Specify a state that must be reached without prescribing every intermediate action [Dignum 2004]. This permits different participants and plans to satisfy the same organizational requirement. The flexibility concerns the route; it does not waive permissions, evidence requirements, or constraints on how that state may be reached.

Fox and colleagues' enterprise ontology distinguishes organizational relationships including empowerment and authority [Fox et al. 1996]. Terminology varies across formalisms. In this manuscript, technical access, institutional power, and permission remain separate questions (Section 7.2). When importing a model, map its terms explicitly instead of assuming that its “authority” means the same thing as an access token or another framework's “permission.”

Concrete example 3. MOISE: roles and obligations are different declarations

Specimen • XML selections, whitespace normalized. The MOISE repository's house-construction test specification, `src/test/jcm/house-os.xml`, at commit `2de19517d62798521458d2aff-bc556ee911a7f3b`. The selections below come from its structural, functional, and normative sections.

```
<role id="bricklayer" min="1" max="2"/>

<mission id="build_walls" min="1" max="1">
  <goal id="walls_built" />
</mission>

<norm id="n4" type="obligation"
  role="bricklayer" mission="build_walls" />
```

Read the artifact. The role permits **one or two occupants**. The mission names a **goal bundle**. The norm binds a role to that mission as an **obligation**. The same file separately declares an authority link from `house_owner` to `building_company`. None of these declarations can safely stand in for the others. The XML root declares `os-version="0.8"`; it is a versioned MOISE implementation artifact, not a verbatim serialization mandated by the 2002 paper.

Borrow the idea. Inspect the organization before inspecting an individual agent's prompt. Ask which section changes when staffing changes, which changes when the plan changes, and which changes when a duty changes. Syntax inspection and XML parsing do not demonstrate that agents complete the house-building task.

Source and verification. Repository source inspected; excerpts checked against the pinned file. This selection omits the enclosing XML document. Follow the repository's license for reuse beyond this short explanatory quotation.²⁴ **Return:** [roles and missions](#).

5.5 OMNI: dimensions are not refinement levels

OMNI integrates organizational, normative, and ontological concerns across abstract, concrete, and implementation levels [Dignum et al. 2005]. The dimensions say **what concern** is being modeled; the levels say **how concretely** it is specified. Mixing them hides unresolved design work.

24. [MOISE house specification](#)



Key concept

Dimensions versus refinement levels.

OMNI separates organizational, normative, and ontological concerns from abstract, concrete, and implementation-level descriptions [Dignum et al. 2005]. A detailed schema can still leave a policy question unanswered. Check both what concern a statement addresses and how far it has been made operational.

Concern	Abstract	Concrete	Implementation
Organizational	Accountability and coordination pattern	Named roles, groups, participation arrangements	Enactment records and interaction operations
Normative	General obligations and protected values	Applicable permissions, prohibitions, and remedies	Guards, monitoring, violation records
Ontological	Shared concepts and relationships	Agreed domain vocabulary and identifiers	Schemas, event types, data mappings

The cells above are illustrative uses of the framework, not a verbatim OMNI schema. For example, “protect customer privacy” is not yet a complete rule about which fields a service agent may read. A JSON schema for a customer record is not itself a decision about whether that access is permitted.



Caution – A vocabulary, a policy, and a runtime mapping solve different problems. Author the relationship among them. Keep the meaning of “accepted,” “authorized,” and “effective” stable across departments, and version the operations that make those institutional facts true. Neither a prompt glossary nor a database field named `approved` establishes the missing organizational semantics.

InstAL

Selection principle: an analysis methodology and a runtime specification can be complementary. A diagram that clarifies responsibilities is not itself an enforcement mechanism. The MOISE and InstAL source boxes show what some of these distinctions look like after they acquire executable syntax.

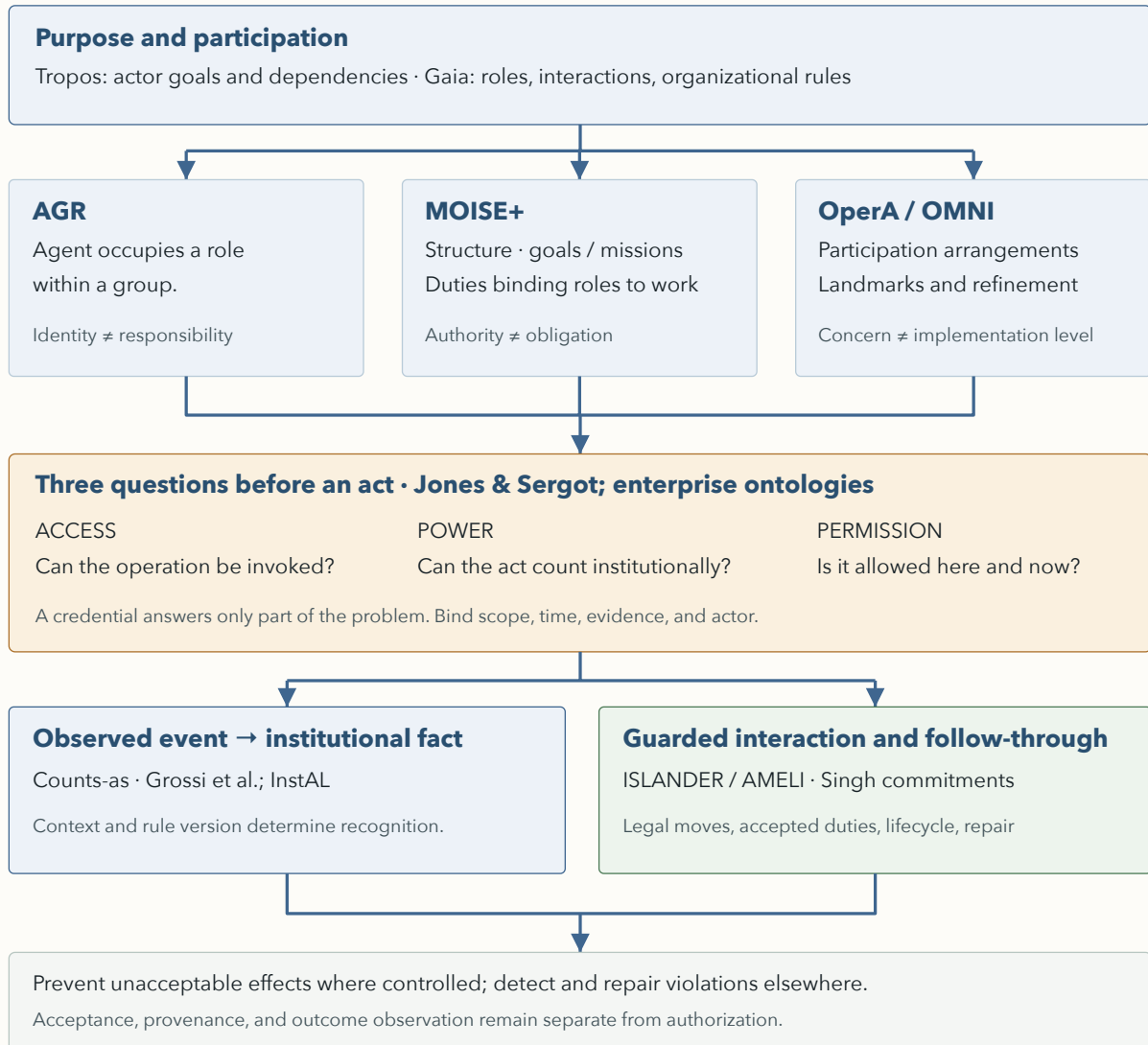
What to inspect next. Compare this synthesis with the role, mission, and norm fields in [the MOISE specimen](#). The original formalisms use different representations, including goal-dependency diagrams, role schemas, participation models, and cross-level specifications. There is no single configuration format shared by all of them.



How to read Synthesis plate III. Move from goals to organizational positions, missions, and participation arrangements. The central band separates **access, power, and permission**. Below it, event interpretation and guarded interaction lead to recognized effects and follow-through. The frameworks supply different views, not interchangeable implementations. The synthesis draws on AGR, MOISE+, OperA, OMNI, institutional-power theory, and electronic institutions, discussed in Chapters 5 and 8. The [MOISE](#) and [InstAL](#) specimens show two concrete representations of these distinctions.

INSTITUTIONAL DESIGN • FROM PURPOSE TO RECOGNIZED EFFECT

Complementary views, not equivalent languages or a ready-made integration standard.



Arrows: a design-to-execution reading path. No arrow confers authority by itself.

Synthesis plate III. From organizational purpose to a recognized institutional effect.

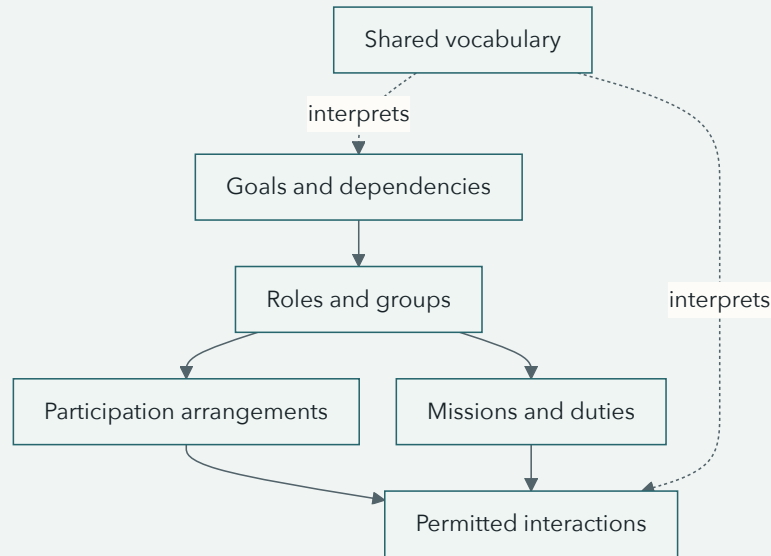
Compare related approaches – organizational specification. All of these make organizational structure explicit, but they do not operate at the same level [Ferber and Gutknecht 1998; Hübner et al. 2002a, 2002b; Dignum 2004; Dignum et al. 2005; Zambonelli et al. 2003; Bresciani et al. 2004].

Approach	Common concern	Distinctive contribution	Artifact to inspect	Choose or combine
AGR	Actors in an organization	Minimal separation of agents, groups, and roles	Group/role model and occupancy	A compact structural starting point; add goals and norms when needed
MOISE+	Roles pursuing collective goals	Linked structural, functional, and deontic specifications	Role definitions, goal schemes, missions, norms	Useful when duties must bind roles to goal bundles
OperA	Participation under organizational requirements	Organizational requirements, contracts, and interaction arrangements	Role contracts and landmarks	Useful for heterogeneous participants with different internal plans
OMNI	Coherent institutional design	Organizational, normative, and ontological dimensions across refinement levels	Cross-level specification and mappings	A design discipline for keeping concerns and implementations aligned
Gaia	Engineering a multi-agent system	Role responsibilities, interactions, and organizational rules	Analysis/design models	Use before implementation to clarify responsibilities and invariants
Tropos	Engineering from stakeholder needs	Goals and dependencies carried through development	Actor/goal dependency models	Start here when the reason for collaboration is still unresolved

Reading map 4. Organizational models: what a specification must separate

Reading map • editorial synthesis. Follow the questions from goals through roles to permitted interactions. The diagram compares concerns addressed by different organizational models; their source descriptions are listed below.

Read the map. Tropos makes actor goals and dependencies central. Gaia connects role responsibilities and interactions. AGR supplies a compact agent–group–role vocabulary. MOISE links roles, missions, and norms. OperA distinguishes organizational requirements from participation and interaction arrangements. OMNI checks organizational, normative, and ontological concerns across levels of refinement. The arrows here are **design questions**, not a proposed universal translation between those languages.

FIGURE 2

Sources. Tropos, ²⁵; Gaia, ²⁶; AGR, ²⁷; OperA, ²⁸; OMNI, ²⁹. These are model-description leads; detailed facsimiles and historical runtime behavior were not independently reproduced for this publication. **Return:** [organizational specification](#).

5.6 Further afield

Top recommended readings

Direct organizational models, ranked for the chapter's central separation of occupants, missions, and duties.

1. **Hübner, Sichman, and Boissier (2002b)**, *A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems*.

Read this fuller MOISE+ account alongside the shorter AAMAS contribution (2002a). Linking structural positions, functional goals and missions, and deontic relations is the decisive contribution for this chapter. Trace one obligation across those descriptions. The repository XML specimen is a later, versioned implementation artifact, not syntax guaranteed by the original paper.

2. **Ferber and Gutknecht (1998)**, *A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems*.

AGR gives the earlier, smaller foundation: an agent is not the role it occupies, and a role belongs in a group. Its economy makes it a useful comparison when deciding whether a full specification language is needed. Read for membership, enactment, and organizational relations; minimal vocabulary alone does not ensure that responsibility survives replacement.

25. <https://doi.org/10.1023/B:AGNT.0000018806.20944.ef>

26. <https://doi.org/10.1145/958961.958963>

27. <https://doi.org/10.1109/ICMAS.1998.699041>

28. <https://dspace.library.uu.nl/handle/1874/890>

29. <https://dspace.library.uu.nl/handle/1874/11498>

3. **Dignum (2004), *A Model for Organizational Interaction: Based on Agents, Founded in Logic*.**

OperA is the next step when participants may follow different internal plans. Its separation of requirements, participation, and interaction explains why landmarks can specify what must become true without prescribing every action. Read for the contract between flexibility and obligation. A choice of route does not authorize different acceptance criteria or otherwise prohibited means.

Broader explorations

A role is also something people identify with. Ashforth and Mael bring social identity theory into organizational analysis: identification concerns a sense of oneness with a group, with consequences for role conflict and relations between groups [Ashforth and Mael 1989]. This psychological perspective asks different questions from a formal occupancy record.

In human-agent organizations, explore whether naming an agent "auditor," "colleague," or "assistant" changes how people challenge it, attribute responsibility, or trust its output. Hold its actual capabilities constant. The hypothesis concerns human responses to institutional framing, not an assumption that software experiences belonging or professional identity. A provocative extension is to ask whether highly personified roles obscure the very replacement and accountability boundaries they were meant to clarify.

Roles have a presented face. Erving Goffman's *The Presentation of Self in Everyday Life* examines how people manage impressions in interaction [Goffman 1959]. It offers cultural context for role labels and interfaces: the visible presentation of competence is not identical to the work that supports it. Compare an agent introduced as a colleague, an expert, or a tool while holding its actual behavior constant. What changes in the user's questions and willingness to challenge it? Goffman's account concerns human social performance; applying it to interface design is a hypothesis about people's interpretation, not proof that a model has a backstage self.

Professional boundaries are negotiated. Abbott's *The System of Professions* asks how groups establish and contest jurisdiction over work [Abbott 1988]. Use it here to examine responsibility at a boundary: who identifies the problem, who interprets the evidence, and who is entitled to prescribe an action? Formal role compatibility rules may help operational clarity but cannot settle all professional or legal claims. An informative study could follow a task that crosses several occupations and compare the documented responsibility with the actual negotiation. The relevant lesson is that a clean role schema does not erase the surrounding institutional history.

Learning a role is participation in a practice. Lave and Wenger's *Situated Learning* develops legitimate peripheral participation as a way to understand learning in social practice [Lave and Wenger 1991]. This contrasts with treating expertise as a document that can simply be loaded into a worker. Ask how novices obtain access to meaningful tasks, feedback, and the judgments that distinguish competent performance. For human-agent organizations, a useful question is how automation changes opportunities to learn and retain expertise. The comparison concerns the organization of human learning; a model receiving more examples is not automatically participating in the social process the authors describe.

The work of connecting work. Anselm Strauss's "Work and the Division of Labor" gives attention to the articulation needed to make separate activities fit together [Strauss 1985]. This is useful when a role model looks complete but handoffs still fail. Someone must negotiate timing, repair misunderstandings, handle exceptions, and reassemble activities when plans change. Record that coordination labor rather than treating it as accidental overhead. An agent system may automate part of it while shifting the remainder onto operators. The research question is whether the new arrangement reduces the total burden or merely makes an essential part of the work less visible.

6 Coordination and organizational topology

© **Core point — Topology follows dependency.** Pipelines move artifacts; hierarchies route assignments and summaries; blackboards coordinate through shared state; contracts allocate work; deliberation exchanges reasons. Choose the dependency mechanism before choosing the number of agents.

6.1 Start with dependencies, not agent personalities

Chapter 5 identified who occupies a role and what that role owes. Coordination now asks how their work fits together. Organizational research, concurrent programming, workflow systems, and agent communication approach this problem from different directions. They become easier to combine when we separate the questions each answers, rather than choosing one community's vocabulary for all of them.

Design question	Examples of answers	What must still be supplied
How is work arranged?	Pipeline, supervisor, peer team, blackboard, Contract Net	Concrete interaction and execution contracts
How do participants exchange or discover work?	Tuple matching, addressed messages, channels, connectors	Meaning, ownership, and recovery rules
How is behavior described or analyzed?	CSP, CCS and the π -calculus; workflow state models	A matching implementation and relevant environmental assumptions
How do independently built systems communicate?	KQML/FIPA communicative acts; MCP tools; A2A tasks	Domain interpretation, institutional authority, and result acceptance
What survives a pause, retry, or replacement?	Durable workflow state, transactions, leases, operation identities	Reconciliation of uncertain effects and independent outcome checks

A2A

A pipeline can use channels between stages or a durable work store; a supervisor can send A2A requests or publish work into a tuple space. The π -calculus can model a changing network of connections without being the network's transport. Some choices substitute for one another; others can be combined, with constraints. They are not competing names for one layer, nor independent switches that can always be mixed freely. Visual study A relates **coordination patterns** to **the contracts needed to realize them**. The rest of this chapter develops those contracts [Hoare 1978; Gelernter 1985; Milner et al. 1992; Arbab 2004].

Coordination solves **dependencies among work**: precedence, shared resources, mutual exclusion, information flow, and joint completion. Choosing “five agents with colorful roles” before mapping dependencies reverses the design process.

Coordination pattern	Mechanism	Strong when	Main failure	Prefer simpler approach when
Sequential pipeline	Fixed artifact handoffs	Stages and acceptance tests are known	Early errors propagate	One deterministic function can do all stages
Hierarchy/supervisor	Manager assigns and integrates	Exceptions are sparse; ownership is clear	Bottleneck and distorted summaries	Workers are independent and outputs merge mechanically
Blackboard	Specialists post to shared state; scheduler selects next work	Problems are opportunistic and heterogeneous	Shared-state/control bottleneck	Order is stable
Contract Net allocation	Announce, bid, award, report	Costs and capabilities vary; allocation is the problem	Communication and strategic overhead	Static assignment is adequate
Peer team/debate	Agents exchange proposals and critique	Evidence is distributed; disagreement is informative	Endless talk, conformity, correlated error	One accountable expert plus a check suffices
Orchestrator-worker	Planner decomposes parallel work and synthesizes	Subtasks are genuinely separable	Bad decomposition and merge errors	Parallelism does not reduce critical path

Contract Net
Hearsay-II
Joint intentions
STEAM teamwork

Contract Net pioneered distributed task allocation through announce, bid, award, and report [Smith 1980]. Blackboard systems such as Hearsay-II separate a shared problem state from opportunistically activated knowledge sources [Erman et al. 1980]. Joint-intention and STEAM traditions model teamwork as joint commitment rather than mere message exchange [Cohen and Levesque 1991; Tambe 1997]. Contemporary systems draw on related patterns, but a resemblance does not establish that they implement the same commitment semantics. These organizational patterns are not all topologies in the strict sense. A topology describes connections; Contract Net also specifies an allocation interaction, and a blackboard includes problem state and a selection policy. They must be distinguished from their computational substrate: a blackboard may use a transactional store or a tuple space; a hierarchy may communicate through actors or channels. The topology alone does not specify atomic claim, message ordering, backpressure, or recovery after a participant fails.

6.2 Complementarity and composition

GitLab's recovery plan combines sequential setup with parallel copying and subsequent webhook recovery.³⁰ That is a documented mixed dependency structure. Reading it as a dependency graph is useful; calling it an implemented blackboard or Contract Net would add mechanisms the source does not establish.

Practice note

Blackboard, supervisor, and orchestrator differ.

A supervisor owns assignment and integration. An orchestrator commonly constructs a task decomposition for a particular run and merges results. A blackboard lets specialists react to a shared partial solution; a control policy still decides which eligible contribution runs next.

For an incident with an unknown cause, the first observation may invalidate the original investigation plan. A board can expose it to every eligible specialist without routing each exchange through a manager. But write conflicts, stale observations, and scheduling now matter more. For a known restore procedure, a pipeline may be simpler. Choose for changing dependencies, not conversational appearance [Erman et al. 1980; Malone and Crowston 1994].

30. GitLab, "Postmortem of database outage of January 31," 10 February 2017, especially "Broken recovery procedures," "Recovering GitLab.com," and "Root cause analysis." This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.

 Key concept
Blackboard coordination.

Specialists contribute to an explicit shared problem representation; a control policy selects useful next contributions [Erman et al. 1980]. The board is neither another manager nor an unstructured chat history. Its advantage is opportunistic coordination; scheduling and shared-state consistency remain real work.

Two related views, not two independent menus. The first drawing asks who selects contributions and how they become a joint result. The second opens up interactions within such a pattern: how participants find each other, exchange data, resume work, and protect shared state. It is not a second set of six organizational alternatives, and its panels do not correspond by position to those in the first drawing. A durable workflow, for example, helps implement the very order shown by a pipeline; it is not an unrelated extra dimension.

The contract view keeps those identities where it shows participants. In M1, S1 publishes, S2 consumes, and S3 observes; these are operations performed by the specialists, not three newly named actors. In M2 and M3, the same S1 and S2 are realized as actors or communicating processes. Other specialists are omitted to focus on one interaction, not removed from the organization. Mailboxes, channels, records, and tool servers are different kinds of objects.

Follow one handoff through both views. In an editorial design sketch, S1 produces a finding, S2 checks it, and S3 integrates the checked result. That is P1's precedence pattern. S1 could publish an explicitly identified work record for S2 to consume (M1), or send a message to S2's actor (M2), without changing that precedence. The implementations differ: taking a tuple and receiving a message do not have identical ownership or recovery semantics. M1 shows the first handoff: S3's read is observation, not permission to integrate an unchecked finding. Integration remains conditional on S2's checked output.

 **How to read Visual study A's pattern view.** Follow blue information arrows and amber assignment or decision arrows. S1, S2, and S3 are identities, not different job names in every panel. The supervisor, announcer, selection policy, and decomposition/merge functions describe coordination responsibilities; they need not be additional LLM agents. In P3, distinguish the **shared problem state** from the **policy choosing the next contribution**. Shared state is not itself a scheduler, and the drawing makes no crash-durability guarantee. These are source-grounded schematic patterns, not performance results [Smith 1980; Erman et al. 1980; Horling and Lesser 2004].

Blue: information / artifacts

S1-S3: specialist identities. Hubs: coordinating functions, not necessarily extra agents.

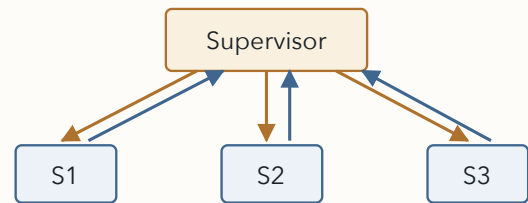
P1 Pipeline



Each result becomes the next input.
Fixed order; early mistakes travel.

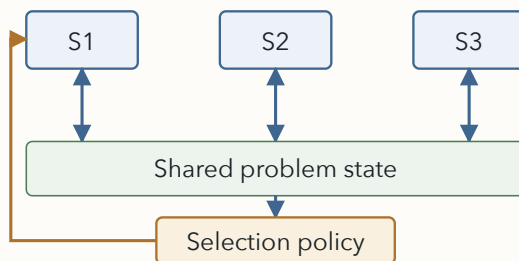
Amber: assignment / decision

P2 Supervision



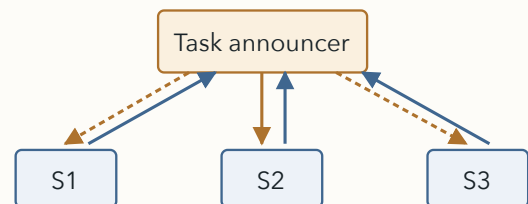
Assignments down; summaries up.

P3 Blackboard



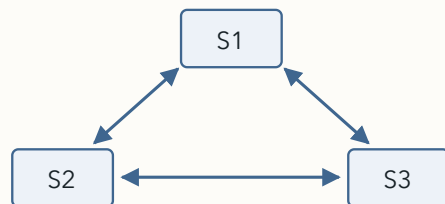
State informs policy; policy selects S1.

P4 Contract Net allocation



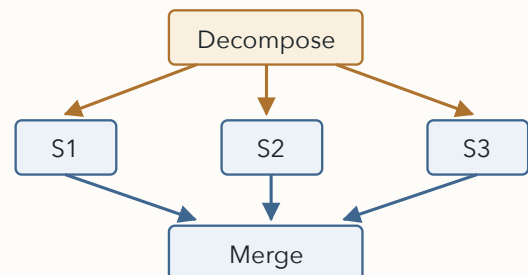
Broadcast; receive bids; award S2.
Dashed: announcement. Solid: award.

P5 Peer deliberation



Reasons circulate among peers.
The decision owner is still required.

P6 Orchestrator / workers



Parallel tasks, then explicit synthesis.

Visual study A, pattern view. P1-P6 compare coordination patterns while retaining specialist identities S1-S3. Coordinating functions may be supplied by people, programs, or agents.

🕒 **How to read Visual study A's contract view.** M1 discovers a tuple by content; M2 routes a message to an addressed actor; M3 makes channel and connector constraints explicit. These interaction models may offer alternatives for a particular handoff. M4 records execution order and progress; M5 defines an interface across independently implemented systems; M6 governs shared-state changes. They answer different questions but are **not independent layers**: recovery depends on delivery, operation identity, and state semantics. CSP, the pi-calculus, and Reo are distinct formalisms or coordination models, not three names for the channel pictured. Section 6.6 develops their differences.

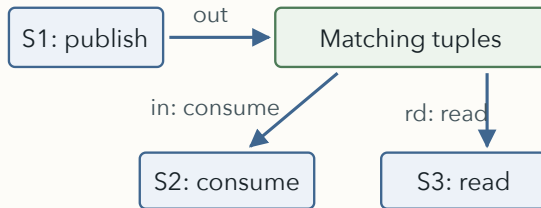
Examine a handoff without changing specialist identities

S1-S3 are participants; the other objects expose implementation contracts.

Blue: data / messages

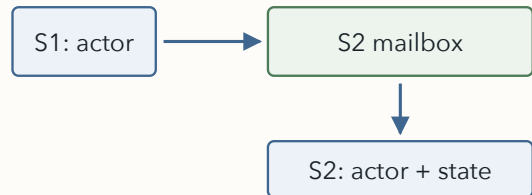
Amber: execution / admission control

M1 Interaction: tuple space



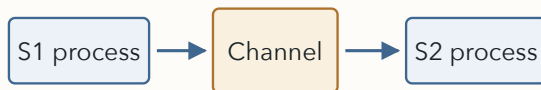
Discover work by content, not address.
Taking a tuple does not finish its task.

M2 Interaction: actors



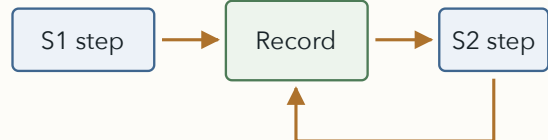
Receiver identity routes the message.
Delivery and recovery need a contract.

M3 Interaction: channels



CSP: synchronized input / output.
π-calculus: pass channel names.
Reo: connectors compose constraints.
Distinct models, not one channel API.

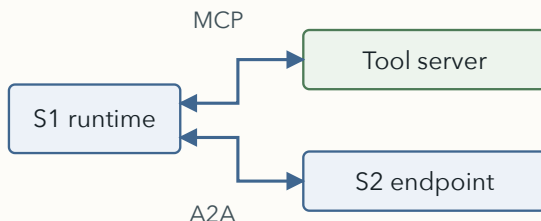
M4 Execution: durable workflow



Resume from recorded execution state.

A workflow owns order and progress.
Remote effects still need reconciliation.

M5 Interfaces: tool or agent



A tool server is not another specialist.
KQML / ACL: communicative meaning.

M1-M3: interaction models. M4-M6: further contracts. No one-to-one match to P1-P6.

M6 State: ownership and merging



Exclusive ownership needs agreement
or correctly allocated rights.

Mergeable evidence can use CRDTs;
convergence is not authority or truth.

Visual study A, contract view. M1-M3 contrast interaction models; M4-M6 expose execution, interface, and shared-state contracts. Specialist identities remain stable, but these contracts overlap and constrain one another.

When recovery after interruption matters, M4 records the workflow's progress; it does not prove that a lost reply means a failed external action. If S2 is a separately exposed agent, M5's A2A endpoint supplies an interface. MCP instead connects a runtime to a tool server: relabeling that server S2 would falsely turn an endpoint into a specialist. Concurrent attempts may also require M6's claim or version checks. None of these additions establishes the finding's truth or S2's institutional authority.

Contract Net

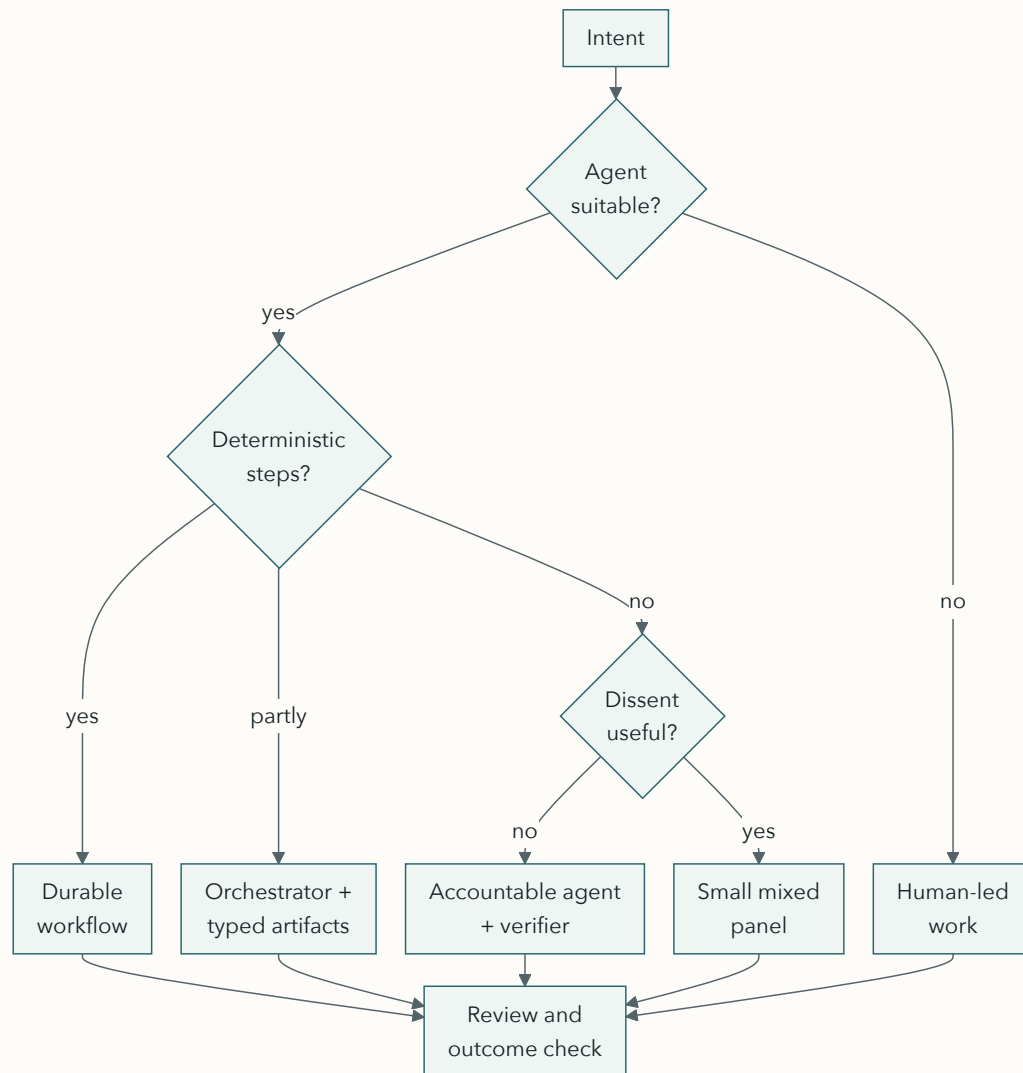
Changing this sketch to P3 is more than changing the transport. Contributions now depend on shared problem state and a selection policy, rather than only the fixed S1-to-S2-to-S3 order. A tuple space could store some of that state, but it would not supply the policy. Similarly, an A2A connection can carry messages used by P4 without implementing Contract Net's announcement, bidding, award, and commitment rules automatically.

Pattern view	Possible contract-view realization	What the combination must still specify
P1 Pipeline	M1, M2, or M3 for a handoff; M4 for durable progress	Stage preconditions, acceptance, and safe retry
P2 Supervision	M2 addressed workers or M5 remote-agent interfaces	Assignment rights, exception routing, and accountable integration
P3 Blackboard	M1 as one possible state substrate; M6 for concurrent changes	Shared problem representation and contribution-selection policy
P4 Contract Net allocation	M2 messages and/or M5 interoperable endpoints	The allocation protocol, award identity, and outstanding commitments
P5 Peer deliberation	M2 or M5 for exchanges; recorded reasons where needed	Argument semantics, termination, and a decision owner
P6 Orchestrator-worker	M4 for dependencies and joins; M5 for tools or remote workers	Decomposition quality, merge criteria, and uncertain-effect recovery

Convergent replicated data

These are illustrative combinations, **not required stacks or equivalence claims**. A durable workflow can support supervision or a blackboard scheduler, not only a pipeline. Actor messages can be carried across protocol boundaries. CRDT convergence is suitable only for compatible merge semantics; it does not make every shared-state invariant coordination-free. Start from the work dependency, choose compatible contracts, and then test their composition.

🕒 **How to read Figure 3.** Begin with demonstrated suitability, then decomposition, not desired agent count. If capability or risk prevents appropriate agent use, keep the work human-led rather than compensating with more agents. "Partly" means some components decompose deterministically while others require judgment or negotiation; keep the former in a workflow and use agents for the latter. "No" means judgment, opportunism, or value conflict runs through the problem rather than forming an isolatable step. Where agent assistance is justified, distinct evidence or competing values may warrant a panel. Every branch converges on an outcome check. The conclusion is that multi-agent design is an escalation from simpler machinery, not a default.

FIGURE 3

Compare related approaches – who chooses the next contribution? This comparison complements Visual study A. It concerns control and information, not an empirical speed ranking [Erman et al. 1980; Smith 1980; Tambe 1997].

Pattern	What they share	Who selects work?	Where intermediate information lives	Main tradeoff
Pipeline	Multiple transformations toward an outcome	Predetermined sequence or branching rule	Handoff artifacts	Predictability versus inability to exploit unforeseen contributions
Supervisor	Work is distributed across specialists	Accountable coordinator	Coordinator state and worker outputs	Clear ownership versus bottleneck and information loss
Blackboard	Specialists contribute partial solutions	Scheduling/control policy reacting to shared state	Explicit common problem representation	Opportunism versus shared-state conflicts and scheduling complexity
Contract Net	Tasks are allocated among candidates	Announcer awards after proposals	Announcement, bids, award, result	Capability/cost discovery versus negotiation overhead
Peer deliberation	Participants exchange relevant information	Interaction protocol, often with a final decision owner	Proposals, critiques, shared record	Distributed evidence versus convergence and termination problems
Orchestrator-worker	Decomposition and specialist execution	Planner for the current task	Subtask outputs and merge state	Parallelism versus decomposition and integration errors

 Key concept

Blackboard is not tuple space.

A blackboard architecture organizes problem-solving around shared partial solutions and a selection policy for contributions. Linda supplies associative publication, observation, and consumption operations [Gelernter 1985; Carriero et al. 1994]. A tuple space can support part of a blackboard, but it does not automatically supply its scheduler, task semantics, or independent acceptance criteria.

Combine carefully: a hierarchy may own accountability while a blackboard supports diagnosis and a pipeline performs the approved repair. The organization must still state which component owns the final acceptance decision.

Concrete example 5. AutoGen: the speaker-selection policy is visible

Specimen • method-body excerpt. AutoGen AgentChat's

RoundRobinGroupChatManager.select_speaker, commit 027ecf0a379bc-c1d09956d46d12d44a3ad9cee14. The long assignment is wrapped for print.

```
current_speaker_index = self._next_speaker_index
self._next_speaker_index = (
    current_speaker_index + 1
) % len(self._participant_names)
current_speaker = self._participant_names[current_speaker_index]
return current_speaker
```

Read the artifact. A stored index chooses the next participant; modular arithmetic rotates through the list. **The scheduler is deterministic even if participants use language models.** The accompanying example constructs `RoundRobinGroupChat([agent1, agent2], termination_condition=termination)`. Turn selection and termination are separate mechanisms.

Borrow the idea. Before calling a team “collaborative,” inspect who speaks, what each participant sees, and what stops the conversation. A `TERMINATE` marker is a protocol condition, not independent acceptance of a business outcome.

Source and verification. Source excerpt inspected; the extracted scheduler was tested over seven calls with three participants, without a model. No multi-agent conversation was run for this source example. AutoGen repository license: MIT.³¹

Return: [coordination topologies](#).

6.3 Failure arithmetic

If independent steps each succeed with probability p , an uncorrected chain of n required steps succeeds with probability p^n . At $p = 0.95$ and $n = 50$, this is about 7.7%. Real failures are not independent, but the calculation shows why more handoffs do not automatically improve reliability.

Durable execution, retries, checkpoints, and idempotency reduce **infrastructure failure**. They do not correct a **shared misunderstanding of the objective**. For that, use explicit acceptance criteria, independent evidence, and milestone verification.

Nor do these mechanisms remove every infrastructure ambiguity. A participant can claim a task and stop before recording its result; an external effect can succeed while its acknowledgement is lost. Independent step-success arithmetic does not describe these histories. The transition and recovery contracts in Sections 6.5–6.6 and 12.11 must be checked before “retry” is treated as safe.

31. [Round-robin implementation](#)

6.4 Beyond the common patterns

Horling and Lesser also survey coalitions, federations, holarchies, matrix structures, and compound organizations [Horling and Lesser 2004]. A coalition forms around a shared opportunity; a federation connects units through representatives; a holarchy nests units that act as both wholes and parts. A matrix gives a participant more than one organizational axis, such as professional discipline and project membership.

These patterns help when one reporting tree cannot represent the dependencies. They also require answers about conflicting assignments, shared costs, and responsibility after dissolution. GitLab's outage ended, but the postmortem's data-durability ownership and recovery-test commitments concerned continuing operations.³² Temporary incident coordination and durable maintenance obligations must not disappear together. Membership and topology remain design choices, not evidence of effective cooperation.

6.5 Linda and generative communication

Linda separates computation from coordination through a logically shared, associatively addressed **tuple space**: a collection of records whose fields can be matched by content [Gelernter 1985; Gelernter and Carriero 1992]. Participants publish tuples and retrieve them by matching fields rather than by naming a receiver. Spatial decoupling removes the need to know the partner's identity; temporal decoupling allows publication before a consumer is ready. Neither establishes persistence after the space's machine fails.

Operation	What changes	What the caller must not infer
out	Adds an evaluated tuple	Repeated publication is deduplicated
in	Waits for and consumes one matching tuple	The claimed task has finished
rd	Waits for and observes a match without consuming it	The observed resource is reserved
eval	Concurrent evaluation eventually produces a tuple	Every Linda-like library provides this operation
Probe variants	Test current matching availability without waiting for a future tuple	No future work, in-flight claim, or delayed result exists

In the model described by Carriero et al. [1994], several matching tuples need not be selected in FIFO order. Duplicate tuples can coexist. Selection fairness, priority, physical placement, and recovery depend on the implementation and protocol. A globally named space is not necessarily one server, and associative matching is not automatically an efficient query for every pattern.

32. GitLab, "Postmortem of database outage of January 31," 10 February 2017, especially "Broken recovery procedures," "Recovering GitLab.com," and "Root cause analysis." This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.

 Key concept**Atomic claim is not reliable work.**

One consuming operation can exclude a second taker of the same tuple. A stopped consumer can still leave the work unfinished, and two duplicate publications can admit two consumers. Keep publication identity, durable attempt ownership, result submission, and outcome acceptance distinct. Linda's small operation set is useful precisely because its scope can be stated clearly.

The Rinda source box below demonstrates these operations with a real library. Its local test confirms claim, matching, and duplicates; it does not certify a distributed job service. This is a more precise comparison than equating tuple space with either a chat transcript or an ordinary message queue.

 Concrete example 6. Linda and Rinda: matching work without naming its worker

Insight. The local Rinda check shows that consuming a tuple prevents a second take of that record, while duplicate publication can still admit two consumers.

Source specimen. Carriero, Gelernter, Mattson, and Sherman (1994), p. 636, show a Linda expression that publishes a computed table entry:

```
out('table entry', i, j, f(i,j))
```

Read it. Publish a tuple after evaluating its fields. A consumer retrieves by a matching template, not by knowing the producer's address. `in` consumes a match; `rd` observes without consuming it. Publication, claim, and computation are distinct operations.³³

Implementation specimen. Seki's Rinda walkthrough gives this exchange:

```
$ts.write(["take-test", 1])
$ts.take(["take-test", nil])
```

The answer is `["take-test", 1].nil` is a wildcard here. The original walkthrough uses dRuby; the retained local harness uses an in-process tuple space. It also runs the source's factorial request/result pattern and obtains 120. The local checks exercise the documentation's operations without requiring a business scenario.

Verified behavior. On Ruby 2.6.10, two competing threads taking one tuple produce one success and one zero-wait expiration. Two identical writes permit two successful takes. Repeated reads retain the tuple, and a take does not create a completion record. The check covers neither durable recovery nor a network partition; the full Coordination report records those boundaries.³⁴

Source and verification. The Linda expression is illustrative historical notation, inspected rather than run with a Linda compiler. The Ruby results above come from the separate local Rinda harness.

33. Carriero et al., "The Linda Alternative to Message-Passing Systems," *Parallel Computing* 20 (1994), 633-655, especially pp. 635-637. <https://heather.miller.am/teaching/cs7680/pdfs/Linda-Alternative-to-Message-Passing.pdf>.

34. Masatoshi Seki, *The dRuby Book*, Section 6.2, "How Rinda Works." <https://www.druby.org/sidruby/6-2-how-rinda-works.html>. Local verification: the maintained `concurrent-coordination/verify-rinda.rb` harness; no remote service or business action run.

6.6 Channels, actors, spaces, and coordinated invariants

Compare related approaches – computational coordination. These mechanisms can implement organizational patterns, but they make different contracts visible.

Family	Addressing and state	Synchronization focus	Organizational risk if omitted
Linda-like space	Shared tuples selected by content	Read versus exclusive consumption	Claimed work disappears without recovery evidence
CSP-style channels	Explicit communication connections	Rendezvous in the original CSP model; buffering varies in implementations	Deadlock or unbounded pressure hidden behind a handoff
π -calculus	Communication names are themselves transmitted; restricted names delimit scope	Synchronization can change subsequent connectivity	A static diagram hides which participant can reach which channel after a handoff
Actors	Addressed participants with local state	Asynchronous messages and local handling	Restart mistaken for recovery of external effects
Reo connectors	Coordination outside components	Composed interaction constraints	Prompt conventions mistaken for enforced protocol
Transactional work store	Durable task and attempt records	Atomic claim, version checks, lifecycle updates	Separate reads/writes allow duplicate admission
Replicated coordination service	Small shared configuration or ownership state	Ordered updates, sessions, versioned ownership	Stale actors continue acting after takeover
CRDT / monotonic accumulation	Mergeable facts or lattice state	Convergence under defined merge and delivery conditions	Replica agreement mistaken for a valid budget or policy decision

Hoare's Communicating Sequential Processes (CSP) [1978] makes matching input/output a synchronization point. Actor systems emphasize addressed participants and private state. Reo makes connector composition a first-class coordination mechanism [Arbab 2004]. Those are useful alternatives to embedding every interaction rule in the worker's computation. They do not eliminate the need to state the protocol's lifecycle and failure model.

Communicating
sequential
processes
Exogenous
coordination

 Biography
C. A. R. "Tony" Hoare (1934–2026).

Tony Hoare speaking in
Lausanne, 20 June 2011.

Born in Colombo and educated at Oxford, Hoare studied classics and philosophy before moving into statistics and computing. Industrial work at Elliott Brothers preceded academic appointments in Belfast and Oxford, and later research at Microsoft. His contributions include Quicksort, axiomatic reasoning about programs, monitors, and Communicating Sequential Processes.

The 1980 Turing Award recognized his contributions to programming-language definition and design. For this book, his central legacy is making the meaning of an interaction explicit enough to reason about. Elegant notation matters when it brings a difficult behavior under intellectual control, not simply when it shortens its description.³⁵

[Portrait credit.](#)

 Concrete example 7. CSP: a copy process is a pair of synchronized handoffs

Source specimen • normalized transcription. Hoare's 1978 paper, Section 3.1, "COPY," printed page 670. The repetition mark and arrow use ASCII equivalents; the process repeatedly passes a character from west to east.

```
X :: *[c:character; west?c -> east!c]
```

Read the expression. *X* names the copying process. The bracketed command repeats; *c* is its character variable. *west?c* receives a character from the process named *west*, and *east!c* sends it to the process named *east*. The arrow separates the input guard from its following command. Each communication requires the corresponding partner to be ready; there is no implicit unbounded message queue.

The process itself nevertheless provides **one-character buffering**: after receiving a character, *X* holds it while waiting for *east*. Meanwhile *west* can compute its next character. The distinction is between a value held in an intermediate process and automatic buffering inside the communication operation. Hoare explicitly explains this behavior and the termination behavior of the input guard.

A discriminating check. Pause *east* after one input to *X*: *X* cannot receive another character until it completes its output. A trace that accepts unbounded further characters would describe a different contract. This is a reasoned consequence of the cited semantics, not a newly executed historical runtime test. The expression does not define durable storage, external-effect recovery, or institutional permission.

Source and verification. The paper's full text and COPY discussion were inspected. This is historical notation rather than current compiler input; no historical CSP implementation was run.³⁶

35. Cliff Jones, "C. Antony R. Hoare," ACM A. M. Turing Award biography, inspected 16 September 2026. https://amturing.acm.org/award_winners/hoare_4622167.cfm. Dates and principal career/contribution claims checked against this account.

36. Hoare, "Communicating Sequential Processes"

The π -calculus adds changing connectivity. Milner, Parrow, and Walker's calculus treats communication links as names that can themselves be sent in messages [Milner et al. 1992]. After receiving a name, a process can use that link in later communication. This makes a service handoff, a private reply channel, or a changing network expressible as part of the process behavior, rather than an unexplained rewiring of a static diagram. Its lineage includes Milner's Calculus of Communicating Systems (CCS); its purpose is a precise model of interaction, not a wire format or a workflow product.

Biography

Robin Milner (1934–2010).

Milner was a British computer scientist whose career included industrial programming, Stanford, Edinburgh, and Cambridge. His work linked practical tools with precise mathematical accounts of their behavior. The 1991 Turing Award recognized LCF for machine-assisted proof, ML with polymorphic type inference and safe exception handling, and CCS as a theory of concurrency.

With Joachim Parrow and David Walker he developed the π -calculus, which models communication networks whose connections can change. His later bigraphs distinguished locality from connectivity. Across these projects, interaction was a central scientific object, not a detail added after sequential computation.

[ACM's biography and portrait gallery](#) provide further cultural and historical context.³⁷

Do not turn a formal boundary into an unearned security claim. Restricting a name expresses scope within the calculus; a real system still needs identities, access controls, transport protections, and an account of how names can escape. Similarly, a CSP rendezvous says when communication can occur, not whether the message is authorized. The concrete expressions in this section compare what the formalisms make explicit. Chapter 12 then connects that reasoning to protocol and runtime boundaries.

Concrete example 8. The π -calculus: passing a name changes the communication network

Mathematical illustration • editorial example. The π -calculus represents communication links as names that can themselves be communicated [Milner, Parrow, and Walker 1992]. In synchronous monadic notation:

$$\bar{a}(b).P \mid a(x).Q \rightarrow P \mid Q\{b/x\}.$$

Read the rule. The left process sends the name b on a , then continues as P . The right process receives a name on a , binds it to x , and continues as Q . Parallel composition is $\mid$. Communication replaces free uses of x in Q with b , avoiding capture by other binders. P and Q denote process continuations, not wire-format fields.

Consider a private channel passed to a waiting recipient:

$$(\nu b)(\bar{a}(b).\bar{b}(c).0 \mid a(x).x(y).0) \rightarrow (\nu b)(\bar{b}(c).0 \mid b(y).0).$$

37. Michael Fourman, “Robin Milner,” ACM A. M. Turing Award biography, and University of Cambridge's obituary, inspected 16 September 2026. https://amturing.acm.org/award_winners/milner_1569367.cfm; <https://www.cl.cam.ac.uk/obituaries/milner/>.

(νb) creates a scoped fresh name; $\emptyset$ has no further behavior. After the first communication, the recipient can receive c on the newly learned channel b . A second communication leaves $(\nu b)(\emptyset \mid \emptyset)$, equivalent to $\emptyset$ under the usual structural laws. The mobility is in **connectivity**, not a claim that a process physically moved between computers.

Borrow the question. When a service passes a reply channel or another service's address, who can communicate afterward? Name passing makes the change visible. Restriction is a formal scope construct, not proof that a real network has authentication, revocation, or confidentiality. Those require a separate implementation and threat model.

Source and verification. The institutional report abstract establishes name-passing and changing linkage as the calculus's central mechanism. These expressions are original explanatory reductions, not quotations or executed specimens. The published article's full-text route was access-blocked during this inspection; no claim of reproducing its proofs is made.³⁸

Choice rule. Use associative spaces for content-based work discovery; channels for explicit flow and backpressure; actors for state-owning services; durable work records for auditable claim and completion. Use coordination services for small authoritative metadata, not every large artifact. Compose only the mechanisms required by the invariant. Published Linda performance results are workload- and implementation-specific, not proof of a universal advantage over message passing [Carriero et al. 1994].

Concrete example 9. Reo: coordination belongs to the connector

Mathematical reading map • a synchronous-channel contract. Reo composes connectors from channels with defined behavior. The coordination is expressed outside the participating components rather than repeated inside each worker. The following contract describes a synchronized transfer:

$\text{Sync}(a, b): \quad \text{write}(a, v) \text{ and } \text{take}(b, v) \quad \text{complete together.}$

The source end a and sink end b participate in one synchronized transfer of the same value v . If the sink is not ready, this contract does not allow the write to complete by silently queuing the value. A buffered channel would have a different contract: acceptance can precede consumption, but its state and capacity then matter.

Composition is the central idea. Suppose an application needs a handoff to wait for both a receiving service and another required participant. A connector can impose that interaction pattern without asking each component to reconstruct the entire coordination procedure. Conversely, adding a buffer changes when participants may proceed. Drawing the same arrows with different channel semantics does not produce the same system.

38. Milner, Parrow, and Walker, report and publication identity



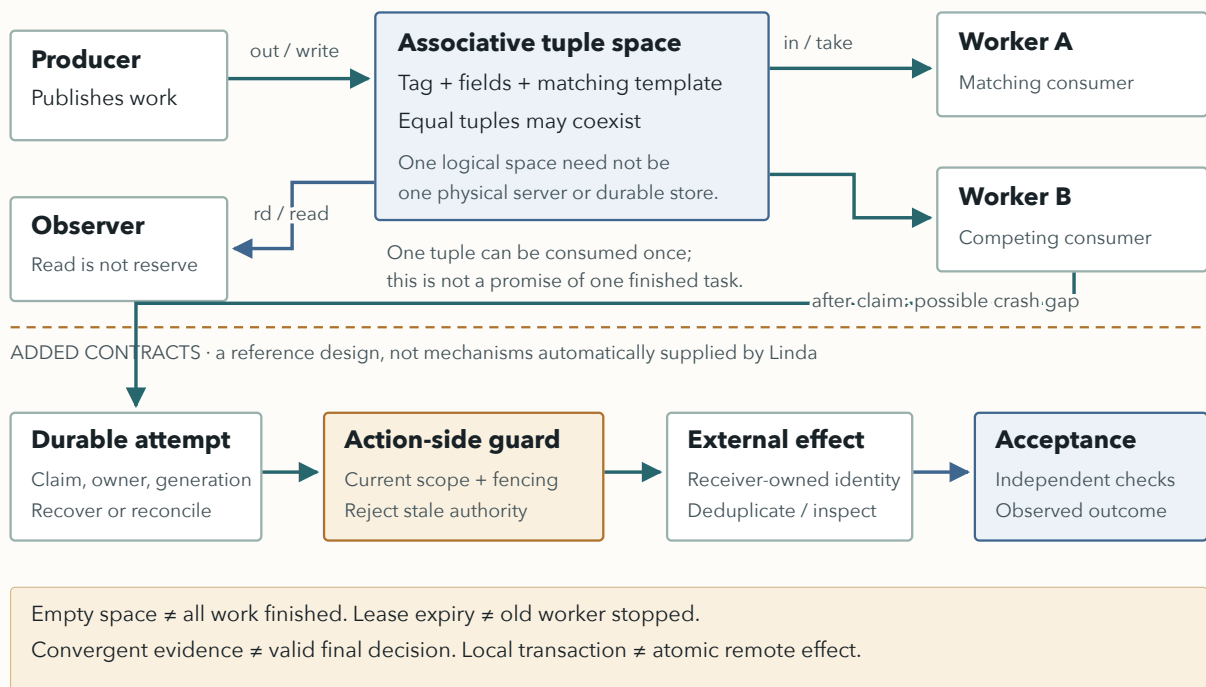
What to inspect. Identify channel ends, channel types, node-composition rules, and any state held by the connector. Check which sets of operations can finish together and which must wait. A Reo connector is neither an organizational manager nor a complete durable-workflow engine. Its coordination constraints need to be connected to the application's duties, failure assumptions, and acceptance checks.

Source and verification. Arbab's primary publication identity and abstract were checked against the CWI institutional record. This is an editorial mathematical explanation of channel-based coordination, not a source-file transcription or an executed Reo tool. The publisher's full-text endpoint was rate-limited; detailed channel implementations must be checked in the chosen Reo toolchain.³⁹

Ⓢ **How to read Visual study B.** Above the dashed boundary, a producer publishes, competing workers consume, and an observer reads without reserving. Below it, durable attempts, action-side guards, external effects, and acceptance require additional contracts. The lower path is an editorial reference design, not functionality automatically implemented by Linda. An empty work space can coexist with active or lost work; a completed take is not completed business work.

COORDINATION IS NOT COMPLETION

Linda-like space operations above; additional reliability and institutional contracts below.



Visual study B. Tuple-space operations and the additional boundaries required for reliable organizational work.

6.7 Business processes: coordinating an outcome across roles

A department groups people or capabilities; a business process follows work across those boundaries toward an outcome. The paint factory in Chapter 4 illustrated coupled decisions; a process view adds the progression of particular work through those responsibilities. Its model need not prescribe every

39. Arbab, "Reo: A Channel-based Coordination Model for Component Composition"

choice: BPMN distinguishes executable and non-executable processes, and case-based and declarative approaches leave decisions to runtime [OMG 2014; OMG 2016].

Key concept

Business process.

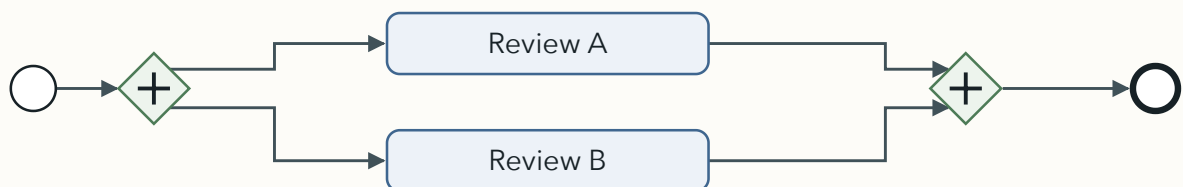
A coordinated set of activities, events, decisions and interactions directed toward an organizational outcome. Its model describes possible behavior; a process instance is one particular unfolding case. A workflow makes aspects of the routing and execution rules explicit enough to support or automate them. Neither the model nor its successful execution proves that the intended business outcome occurred [van der Aalst et al. 2003; OMG 2014].

The process is not just the arrows. The workflow-patterns literature distinguishes control flow, data, resources and activity operations. Choosing a route does not identify its qualified worker, supply its inputs or resolve its exceptions [van der Aalst et al. 2003]. These are the roles and interaction contracts already examined, viewed across a whole case.

BPMN makes routing choices visible. Circles mark events, rounded rectangles activities, and diamonds gateways. A parallel gateway activates both branches; an exclusive gateway chooses one. Their corresponding joins have different semantics. Waiting for both branches after selecting only one can deadlock.

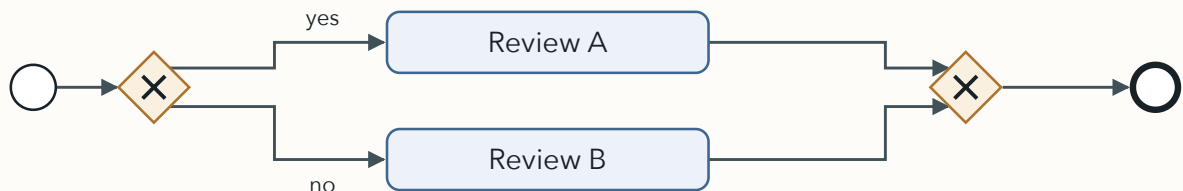
How to read Visual study C. In the upper model, both reviews must finish before the parallel join proceeds. In the lower model, a condition selects one review and the exclusive merge passes that branch onward without waiting for the other. Start events have thin circles; end events have thick circles. These are editorial notation studies based on BPMN 2.0.2 and workflow Patterns 2-5, not measured business cases [OMG 2014; van der Aalst et al. 2003].

Parallel: both branches



Both reviews finish. Their order may vary; the join requires both arrivals.

Exclusive: one branch



One condition selects one review. The merge does not wait for the other.

Visual study C. BPMN parallel and exclusive routing. Each diamond is labeled with its standard gateway marker; solid arrows are sequence flows.

A BPMN pool represents a participant; sequence flow stays inside it and message flow connects participants. Lanes organize activities but do not themselves create autonomous agents. Public process views can hide internal steps [OMG 2014, Sections 7.2, 9 and 10].

Key concept

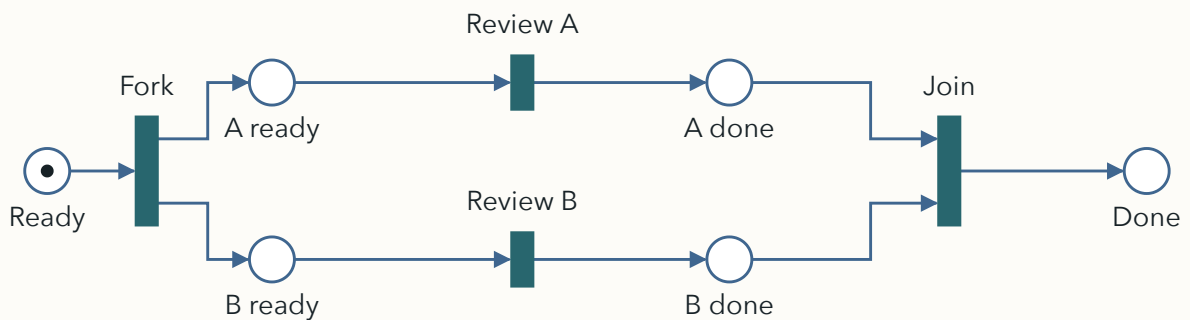
Orchestration and choreography.

Orchestration describes the control of work from one process's perspective. Choreography describes the exchanges expected among participants without supplying all their private implementations. A choreography can constrain autonomous parties, but a diagram of their exchanges does not guarantee compatible implementations, delivery, authorization or eventual completion [OMG 2014; Chopra et al. 2020].

This is a direct connection to MAS interaction protocols. A Contract Net conversation can allocate an activity while a private workflow carries it out. A2A can carry the exchanges without supplying that domain protocol. Several service-task boxes alone do not establish a MAS.

Workflow nets expose state and enabling. A Petri-net place holds tokens; an enabled transition consumes its input tokens and produces output tokens. The same parallel dependency can therefore be inspected as a marking rather than only as a route. YAWL is a related, patterns-motivated workflow language [van der Aalst 1998; van der Aalst and ter Hofstede 2005].

🔗 **How to read Visual study D.** The fork consumes the initial token and enables both reviews. They may finish in either order; the join requires both completion tokens. This unit-weight net has six reachable markings and two complete firing sequences, checked locally from the depicted arcs. It models control flow only, not reviewers' authority or the truth of their findings.



One initial token; six reachable markings; two complete firing sequences.

Visual study D. A small workflow net for the two-review parallel pattern. The filled dot is the initial token; circles are places and bars are transitions.

 Key concept
Workflow soundness.

In its classic workflow-net setting, completion remains possible from every reachable state, reaching the final state leaves no residual work, and no modeled transition is permanently unusable. This is a property of the modeled behavior, not a guarantee of deadlines, factual correctness, legal compliance or business value. An indefinitely repeated permitted loop can still require a termination policy in the implementation [van der Aalst 1998; van der Aalst et al. 2009].

The distinction matters for LLM-generated process models: XML that parses can still describe a deadlock, and a structurally sound model can still omit a required approval. Structural checks complement, rather than replace, the authority and evidence checks in Chapters 8–9.

Flexibility predates LLMs. Declare specifies constraints on acceptable executions instead of prescribing every route [van der Aalst et al. 2009].

 Key concept
Declarative process.

A process specified through constraints on acceptable behavior rather than a single complete procedure. The next step can be chosen at runtime while the constraints remain in force. Underspecifying the constraints can admit unwanted behavior; specifying incompatible ones can leave no acceptable continuation. Flexibility therefore shifts the modeling burden rather than removing it [van der Aalst et al. 2009].

CMMN supports evolving case information and discretionary tasks; DMN separates decision logic from process routing [OMG 2016; OMG 2024]. These are different contracts, not successive degrees of agent intelligence. As with situated human work, choosing a permitted continuation is distinct from changing the rules [Suchman 2007].

6.8 Autonomous participants inside and across processes

Processes and agency can be combined at different levels. ADEPT supplies a concrete pre-LLM example [Jennings et al. 1996].

 Example
ADEPT: a network-service quotation with negotiated participants.

Insight. ADEPT combined explicit process execution with agents negotiating service provision, so a known business goal did not require every resource and provider to be fixed in advance.

BT's quotation process crossed customer services, design, legal review, surveying and external customer vetting. Agents represented departments or enterprises. They negotiated service-level agreements, invoked selected providers and continued their own process descriptions. Some agreements covered repeated requests; occasional services used one-off provision. Negotiation complemented workflow rather than replacing it. A failed negotiation could still fail the case; general dynamic revision was future work.

Source note. Jennings, Faratin, Johnson, Norman, O'Brien and Wiegand (1996), Sections 2–5, Figures 9–10. The implemented research scenario simplified a real process of 38 tasks and nine choice points; it was not a measured production-wide productivity trial. No ADEPT system was rerun for this account.⁴⁰

A process can contain agents; a MAS can contain processes. A human clerk, a deterministic service and an LLM agent can occupy an activity boundary under different assumptions. An independent supplier may keep a private workflow and accept only some contracts. A single principal may instead coordinate several LLM contexts. The relevant differences are ownership, information and discretion, not the number of boxes.

BSPL makes information dependencies explicit. The paper's compact pricing protocol connects this idea to the earlier interaction mechanisms [Chopra et al. 2020, Listing 12]:

```
Pricing {
  role Buyer, Seller
  parameter out ID key, out item, out price
  Buyer → Seller: Request[out ID, out item]
  Seller → Buyer: Offer[in ID, out price]
}
```

out introduces information; **in** requires information already known to the sender; **key** identifies the interaction. Request introduces **ID** and **item**, and Offer uses that **ID** to bind a price. These dependencies, not the printed line order alone, constrain sending. They do not compel a seller to respond or prove that an offer is acceptable. This is the paper's notation with typesetting normalized, checked against the source; it is not an executed commercial service.

Message correlation, delivery assumptions and commitments still matter. Linda can expose work and Contract Net can allocate it without either supplying the whole process. Chapter 12 places these contracts around the runtime.

40. N. R. Jennings, P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien and M. E. Wiegand, "Agent-Based Business Process Management," *International Journal of Cooperative Information Systems* (1996). <https://doi.org/10.1142/S0218843096000051>. Author manuscript, inspected 20 September 2026; Section 3 and its footnotes distinguish the underlying process, implemented scenario and proposed extensions. ADEPT here means *Advanced Decision Environment for Process Tasks*.

6.9 Processes in the LLM era: design, execution and observation

An LLM can **help model a process**, **perform an activity**, or **select and revise a plan**. These are different claims. A generated diagram is not an executed case; a successful activity is not authority to change the procedure. Anthropic's workflow/agent distinction concerns where model-directed choices occur, not the full range of process languages [Schluntz and Zhang 2024].

Example

Hilti's PRODIGY: helping people model their processes.

Insight. The Hilti study found that an LLM process-modeling assistant's usefulness depended on organizational documentation and knowledgeable human review, not just its ability to produce a diagram.

Ziche and Apruzzese (2024) interviewed ten professional process modelers and evaluated a GPT-3.5-Turbo prototype with nine of them. It retrieved local material and produced text for BPMN Sketch Miner. Eight evaluators expected easier work; six expected faster work. The authors retained human review and governance of documentation. This is not autonomous execution or a controlled measurement of time saved.

Source note. Ziche and Apruzzese, *LLM4PM* (2024), Sections 3–6. The evaluation was small and mainly headquarters-based; the prototype itself was not privacy-compliant for general production use. No private Hilti material was accessed for this account.⁴¹

Match the evidence to the claim. A 16-model study uses POWL to construct structurally sound models, assessed against simulated event logs with supplied activity labels; it does not prove business correctness [Kourani et al. 2024b]. FlowBench supplies workflow knowledge to an LLM, not proof that an engine prevents an impermissible transition [Xiao et al. 2024]. Tau-bench checks simulated tool interactions against target database states, not persuasive dialogue alone [Yao et al. 2024]. None establishes a universal MAS advantage.

The *Process Mining Manifesto* separates discovery, conformance and enhancement using event logs [van der Aalst et al. 2012]. Logs can expose skipped steps, repeated attempts or bottlenecks, but conformance to a model does not establish that the model was authorized or that the outcome was useful. Preserve case and activity identities, process versions and missing-evidence states. Chapter 9's acceptance/outcome distinction still applies.

Choice rule. Keep known transitions and consequential effects explicit; delegate interpretation, search or negotiation where justified. Give exceptions an owner, and let agents propose revisions without granting them authority to erase obligations. The fuller source comparison remains in the research review; the architectural consequence is this bounded composition, not another layer of managers.

41. Clara Ziche and Giovanni Apruzzese, "LLM4PM: A Case Study on Using Large Language Models for Process Modeling in Enterprise Organizations," BPM 2024 industry forum. Version inspected: arXiv:2407.17478v1. <https://arxiv.org/html/2407.17478v1>. Ten preliminary interviewees and nine evaluation respondents are different sample counts; perceived helpfulness is not a measured productivity effect.

6.10 Further afield

Top recommended readings

Direct coordination literature, ranked from selecting an arrangement to inspecting two concrete mechanisms.

1. Horling and Lesser (2004), *A Survey of Multi-Agent Organizational Paradigms*.

This survey supplies the chapter's comparative orientation: organizational paradigms address different dependencies and impose different costs. Read the assumptions and tradeoffs, not just the taxonomy of shapes. Historically it consolidates substantial pre-LLM MAS work; it neither ranks current products nor makes every later interaction contract an independent layer.

2. Smith (1980), *The Contract Net Protocol*.

Contract Net is foundational distributed-AI work on allocating tasks through announcement, proposals, and award. It reveals exactly what a mechanism adds beyond selecting the next speaker. Follow the messages and decisions that establish allocation; an award does not demonstrate completed work, settlement of a social commitment, or permission for every consequential action.

3. Gelernter (1985), *Generative Communication in Linda*.

Linda makes coordination inspectable through a small associative shared-space model. Read publication, observation, and consumption carefully, then compare their semantics with addressed messages and a blackboard's selection policy. The historical contribution is a distinct coordination abstraction, not a promise that atomic consumption supplies durable ownership or crash recovery.

Broader explorations

The process literature as a bridge. Read *Workflow Patterns* [van der Aalst et al. 2003] for precise differences between apparently similar arrows and joins, then *Declarative Workflows* [van der Aalst et al. 2009] for the flexibility question. Jennings et al. [1996] show how a process and negotiating agents were combined in a concrete pre-LLM implementation. Read these works alongside, not as substitutes for, current agent-framework taxonomies.

Coordination without conversation. In biological research, *stigmergy* describes coordination mediated by changes participants make to their shared environment. Theraulaz and Bonabeau's history distinguishes mechanisms in which the amount or the kind of environmental change influences subsequent action [Theraulaz and Bonabeau 1999]. This is an unusual companion to blackboards and tuple spaces: a visible work product can guide the next contribution without a manager issuing another message.

A direction to explore. Could a carefully designed artifact workspace reduce the communication needed by an agent team? Compare a message-heavy coordinator with workers responding to explicit, inspectable changes in shared artifacts. Measure stale responses, conflicting updates, and recovery as well as throughput. The biological mechanism does not supply authority, trustworthy signals, or accountability; those remain additional design requirements. Read the original history for the mechanism, not as evidence that biological collectives are ready-made organizational architectures.

Synchronization can emerge from coupling. Strogatz and Stewart's "Coupled Oscillators and Biological Synchronization" introduces systems whose rhythms become coordinated through interaction [Strogatz and Stewart 1993]. It offers an intuitive contrast to coordination by explicit messages or a manager. Which local coupling rules produce a useful collective pattern, and which produce unwanted lockstep behavior? In computational teams, correlated retries or synchronized polling can be harmful rather than efficient. Use the analogy to formulate questions about timing and coupling, not to claim that an agent population obeys the same physical equations as the biological examples.

Local rules and collective patterns. Camazine and colleagues' *Self-Organization in Biological Systems* examines mechanisms through which local interactions produce larger-scale structures and behavior [Camazine et al. 2001]. Read it for feedback, amplification, and the role of environmental conditions rather than for a slogan that no coordination design is necessary. A research task might vary the rules by which workers respond to shared artifacts and measure both useful organization and runaway reinforcement. Emergent order does not establish legitimate authority or good outcomes. The biological literature helps generate mechanisms to test; the application's requirements still decide which patterns are desirable.

Individually modest rules can create strong aggregate effects. Schelling's "Dynamic Models of Segregation" shows how local choices can produce collective patterns that are not obvious from a verbal description of individual preferences [Schelling 1971]. This is a methodological companion to agent-based modeling. Inspect the update rule, neighborhood, and initial conditions before interpreting the result as a claim about society. For artificial organizations, one could study whether local partner-selection rules isolate groups or reinforce a narrow source pool. Such a simulation would test the specified mechanism, not independently establish the causes of segregation in a real organization.

The distribution of thresholds matters. Granovetter's "Threshold Models of Collective Behavior" examines participation that depends on how many others have already acted [Granovetter 1978]. Two groups with similar average attitudes can behave differently because their thresholds are distributed differently. This provides a useful question for escalation and consensus protocols: whose action triggers whose next step? Compare mechanisms that wait for a count with ones that require particular independent evidence. Cascading agreement can be a property of the interaction rule, not corroboration of a claim. Read the model's assumptions carefully before interpreting an observed cascade as the same process.

7 Delegation, authority, and human attention

© **Core point — Delegate rights, not just tasks.** State what the worker may observe, analyze, select, and execute, with scope, expiry, and a no-response default. Escalation quality depends on consequences and evidence, not on a confidence phrase or an alert rule's apparent accuracy.

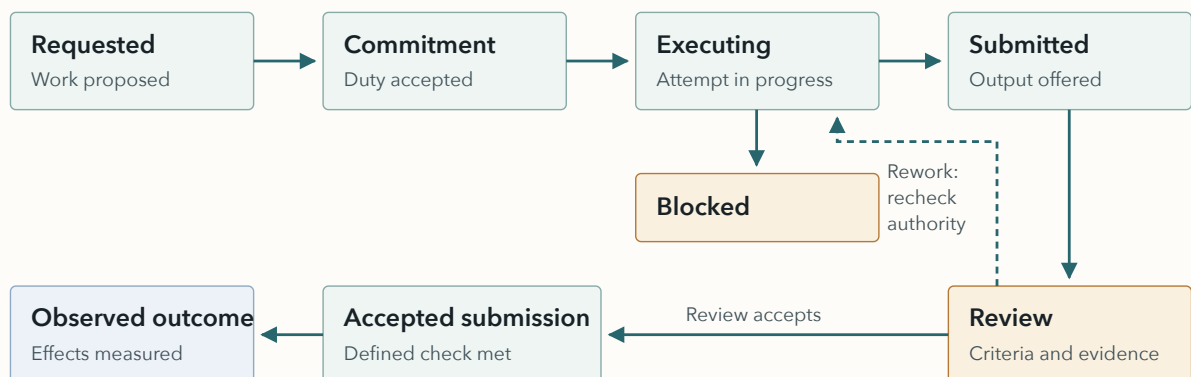
7.1 Delegation is a contract, not a message

Distributing work also distributes opportunities to act. This chapter separates the rights delegated to a worker from the attention a human must retain: first scope and stage of authority, then escalation, handoff, and review capacity.

A useful delegation names: objective, debtor role, creditor or accountable role, scope, resources, deadline, authority, evidence, review triggers, fallback, and revocation. Commitment protocols then track whether the commitment is created, activated, satisfied, violated, cancelled, released, assigned, or delegated [Singh 1998]. Jennings' conventions add rules for monitoring and reconsidering commitments [Jennings 1993].

Do not collapse these states:

Two acceptances, followed by an observation



Simplified lifecycle, not an automatic grant of authority. Observation can reveal success, failure, or uncertainty.

Delegation lifecycle. Accepting a duty is different from accepting its submission. Execution may be blocked; review may require rework and a fresh authority check. Observing the outcome records what happened, not necessarily success.

A worker can declare its output **submitted**. Acceptance requires the designated review mechanism; an observed business outcome remains a separate record.

7.2 Authority is stage-specific

Key concept

Stage-specific automation.

Acquiring information, analyzing it, selecting a decision, and implementing an action can have different allocations of human and machine control [Parasuraman et al. 2000]. Permission to analyze is not permission to execute. A single autonomy score hides this distinction precisely where consequences become real.

Parasuraman, Sheridan, and Wickens divide automation into information acquisition, information analysis, decision selection, and action implementation [Parasuraman et al. 2000]. **[Established framework; optimal level is context-dependent.]** Treat each stage separately.

Reading GitLab's reported recovery through this framework produces the following analysis. It is a classification of activities, not a claim about its formal authorization policy:

Stage	Activity documented in the recovery account
Acquire information	Locate available backups and snapshots
Analyze	Compare age, missing data, and transfer constraints
Select a decision	Choose the more recent staging snapshot as the recovery source
Implement action	Copy data, reconstruct webhooks, adjust sequences, and re-enable service

A single **"autonomy level"** cannot express this. Nor can an **OAuth token**. Distinguish three questions: can the actor reach the payment API (**technical access**), can its act create a recognized transfer (**institutional power**), and is that act allowed in this case (**permission**)? The latter two need not coincide [Jones and Sergot 1996]. This monograph uses **authority** for the governed allocation of decision and action rights, not as a synonym for credentials.

Key concept

Access, power, permission.

Technical access answers whether an interface can be reached. Institutional power answers whether the act can create a recognized effect. Permission answers whether the act is allowed in this case [Jones and Sergot 1996]. A technically possible or institutionally effective act may still violate a rule.

[Heuristic] Permission lease fields: principal, delegate, action class, resource and destination, amount/risk ceiling, start and expiry, concurrency or rate limit, required evidence, rollback or compensation, and revocation path.

 Key concept**Lease expiry needs action-side enforcement.**

An expired lease does not stop a paused or disconnected participant from resuming. Chubby's sequencers carry lock identity, mode, and generation to the receiving service, which must validate them [Burrows 2006, Section 2.4]. For agentic work, require the effect-owning boundary to reject obsolete authority. A generation number in a log, or a check followed by an unguarded remote call, is not fencing.

Keep three leases conceptually separate: a tuple's residence lifetime, a worker's claim lifetime, and an institutional delegation's validity. JavaSpaces exposes the first explicitly; none automatically establishes the other two. Section 12.11 examines their interaction with transactions and external effects.

7.3 Escalation as expected utility

 Key concept**Expected-loss communication.**

The reason to notify someone is the loss that informing them can prevent, compared with the cost of doing so [Tambe 1997]. Low confidence alone does not establish that a human should be interrupted. Consequence, recipient knowledge, and the opportunity to act must enter the judgment.

STEAM teamwork

STEAM's decision-theoretic communication policy compares the expected cost of silence with communication cost [Tambe 1997]. In simplified form, communicate when

$$\tau C_{mt} > C_c,$$

where τ is the probability the recipient does not already know, C_{mt} is the cost of miscoordination if they do not know, and C_c is the cost of communicating. This simplified decision rule expresses a comparison of expected losses; it is a teaching formulation, not a verbatim equation from Tambe. The rule is a design discipline, not a promise that these values are easy to estimate.

Useful extensions add deadline decay, response probability, reversibility, and the value of better human judgment. A fixed “confidence below 70%” rule is weaker because low confidence can be harmless and high-confidence errors can be catastrophic.

7.4 Signal detection and alert fatigue

 Key concept**Positive predictive value.**

The fraction of alerts that are real depends on prevalence as well as sensitivity and specificity. For rare events, false positives can dominate a seemingly accurate detector's queue [Bliss et al. 1995; Meyer 2001]. Estimate the actual review burden, not just the detector's headline accuracy.

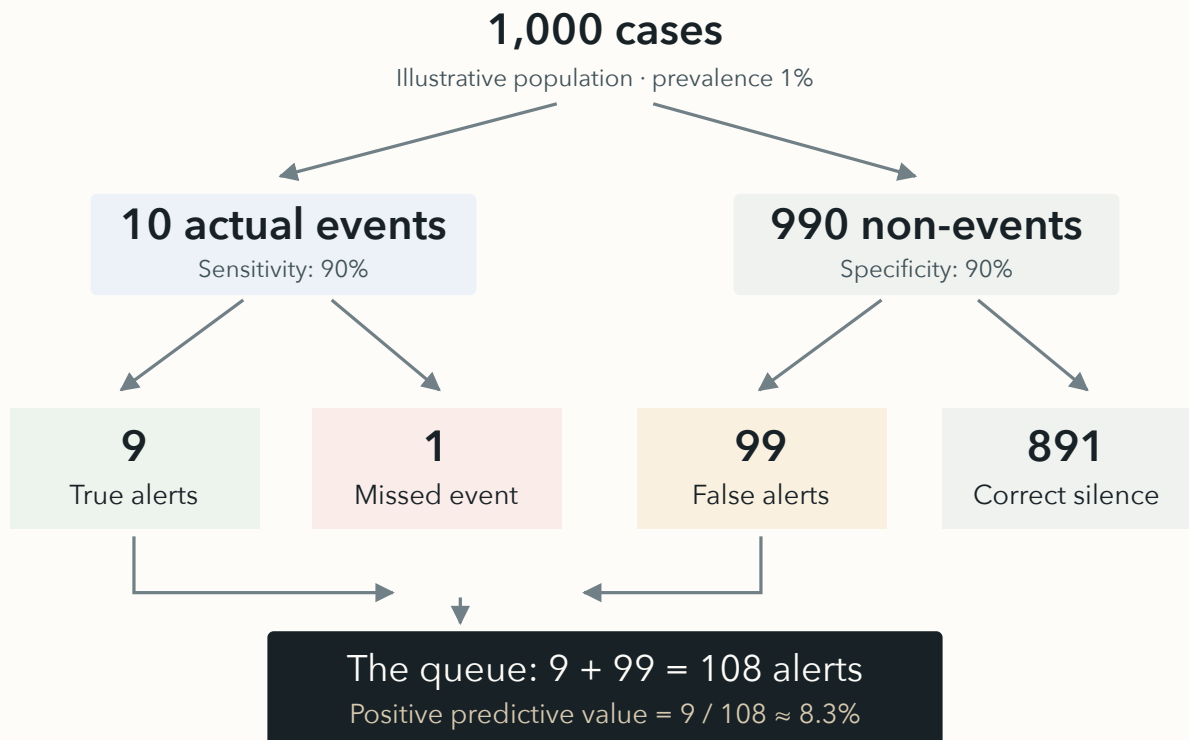
One useful model of escalation is rare-event detection. If event prevalence is π , sensitivity is s , and specificity is c , the positive predictive value (fraction of alerts that are real) is

$$PPV = \frac{s\pi}{s\pi + (1-c)(1-\pi)}.$$

At $\pi = 0.01$, $s = 0.90$, and $c = 0.90$, $PPV \approx 8.3\%$. **Nine alerts out of ten are false** despite a detector described as “90% accurate.” [Established arithmetic and human-factors concern.] Low PPV erodes compliance with alerts; misses erode reliance on silence [Bliss et al. 1995; Meyer 2001].

Measure prevalence, sensitivity, specificity, PPV, misses found by sentinel sampling, time to response, and downstream harm per trigger. **“No alerts”** is not **proof of health**.

🕒 **How to read Visual study C.** Split actual events from non-events, then apply sensitivity and specificity separately. The alert queue merges the two positive branches: only 9 of its 108 entries are real, about 8.3%. These are illustrative counts from Section 7.4’s assumptions, not a measured fraud dataset. Misses and false alarms have distinct costs that need local measurement [Bliss et al. 1995; Meyer 2001].



Visual study C. In 1,000 hypothetical cases, 9 true alerts compete with 99 false alerts.

 More detail

Accuracy is not review capacity.

At the same 1% prevalence and 90% sensitivity, increasing specificity to 99% raises PPV to about 47.6%. This arithmetic holds sensitivity fixed; moving a real detector's threshold commonly changes both measures. Validate the actual operating point.

If 108 alerts each require five minutes, the original hypothetical queue needs nine hours of review per 1,000 cases. Budget arrival rate, handling time, deadline clustering, and reserve for urgent exceptions. Sample the negative branch too: fewer alerts might mean better precision, suppressed detection, or a failed monitor. This is a workload example, not an empirical limit on how many agents a person can supervise.

Separate the denominators. Sensitivity is measured among actual events; specificity among non-events; PPV among raised alerts. In the illustrative population there are ten events and 990 non-events. At 99% specificity, the expected false-alert count is 9.9, while 90% sensitivity still gives nine detected events. Thus $9/(9 + 9.9) \approx 47.6\%$. Fractional counts here are expectations across comparable populations, not a claim that a particular batch contains a fraction of an alert.

Convert detection into workload. Five minutes per alert gives an expected 94.5 minutes of review in that improved example. Compare this with the nine hours required by the original 108-alert queue. Neither figure includes investigation after triage, interruptions to other work, or recovery from mistaken interventions. A useful capacity estimate needs those costs as well as the number of messages displayed.

Check timing, not just totals. Eighteen alerts spread across a day and eighteen arriving just before one irreversible deadline create different supervision problems. Record arrivals, handling times, deadlines, and which actions remain possible at the time of review. If review demand exceeds available attention, restrict work admission or authority rather than assuming a shorter dashboard solves the overload.

Measure what the queue cannot show. Reviewing raised alerts can estimate their usefulness, but it cannot reveal every missed event. Independently sample suppressed or apparently normal work and check monitor health. Preserve the distinction between a detector that found nothing and one that did not run. Threshold selection, staffing, and the consequences of missed interventions form a joint design problem; this arithmetic is only one component of it.

7.5 Interruption channels

In a 36-participant experiment, McFarlane compared immediate, negotiated, mediated, and scheduled interruption; negotiated interruption best supported overall performance in the tested setting [McFarlane 2002]. Transfer cautiously to agent work.

[Heuristic] Use four channels:

- immediate for imminent, high-consequence, time-sensitive events;
- negotiated, with a deadline and safe default, for most decisions;
- next breakpoint or digest for reversible matters;
- audit-only for routine verified work.

Every escalation should answer: What changed? Why me? By when? What happens if I do nothing? What authority is requested? What evidence and dissent exist? What can be reversed?

Example

Toyota: a signal linked to a response.

Insight. Toyota's account of *jidoka* makes detection useful by connecting the abnormality to stopping the work and summoning a responsible responder.

Toyota's account of *jidoka* describes equipment detecting an abnormality and stopping, or an operator pulling a stop cord. The *andon* display identifies the abnormality and calls attention to the affected work. Detection, stopping, and notifying the responsible person are connected parts of the arrangement, rather than three unrelated dashboard features.⁴²

For an agent workflow, the transferable question is what an exception actually does: pause the risky operation, preserve its state, and reach someone able to resolve it. The pause condition and restart authority still need definition. Toyota's description is an institutional account of its own practice, not a controlled test of alert effectiveness or evidence that stopping is always the safest response in another domain.

7.6 Transfer of control, practice, and situation awareness

Sheridan and Verplank's work on human-computer control is an important precursor to stage-specific automation [Sheridan and Verplank 1978]. A handoff is not just a control-flow pause. The recipient must understand the **current state**, the **decision required**, and the **time available**. An automation-initiated handoff can arrive when the human has little context; a human-initiated intervention can arrive after excessive reliance. Neither failure is inevitable, but both belong in representative tests.

Bainbridge's automation ironies explain why rare exceptions can be demanding precisely when routine operation looks most successful [Bainbridge 1983]. The operator may lose practice while still being expected to recover from unfamiliar failures. Recovery exercises, rehearsal, and selective manual practice are possible mitigations. Do not impose surprise consequential decisions merely to manufacture "engagement"; choose training and review tasks according to risk.

Situation Awareness-based Agent Transparency (SAT) distinguishes information about the agent's current action and plan, its reasoning or constraints, and its uncertainty and predicted outcomes [Chen et al. 2014]. This is a useful design frame for a handoff packet. A narrated explanation, however, is still a claim: it must not masquerade as mechanically recorded provenance.

Design implication – A handoff must make the decision recoverable. State what is happening, which constraints apply, what remains uncertain, what authority is requested, the response deadline, and the no-response default. Link the concise packet to the underlying evidence. Evaluate whether a person can actually resume control, not merely whether a "resume" button works [Parasuraman et al. 2000; Chen et al. 2014].

42. Toyota Motor Corporation, *Toyota Production System*, "Jidoka" and the accompanying descriptions of andon displays and operator stop cords, inspected 16 September 2026. <https://global.toyota/en/company/vision-and-philosophy/production-system/>.

Example

Supervision is organized work.

Insight. Apollo's staffed consoles illustrate that supervision requires people with assigned observations and responsibilities, not just more displays.



Historical photograph H4. Engineers monitor an Apollo 11 vehicle test in Launch Control Center Firing Room 1, 1969. Rows of staffed consoles make the distribution of observation and responsibility visible. NASA, KSC-69P-3253.

Supervision depends on distributing observations and making responsibility actionable. The staffed consoles make that work visible: which observations reach which responsible person, and how can that person act? The same questions apply to an agent interface. More displays help only when people can interpret the state and intervene appropriately.

This photograph illustrates an actual control setting rather than the cited human-factors experiments; it does not by itself establish effective oversight. [Image source and reuse terms.](#)

Concrete example 10. LangGraph: pause is a control primitive, not approval policy

Specimen • documentation function, comments omitted. LangGraph's “Pause using interrupt” example, inspected 15 September 2026.

```
from langgraph.types import interrupt

def approval_node(state: State):
    approved = interrupt("Do you approve this action?")
    return {"approved": approved}
```

Read the artifact. The function exposes a question and receives a resume value. Its surrounding graph needs a state declaration, a checkpoint, and a stable thread ID. **On resume, the node starts again from its beginning.** That is why side effects before the interrupt need special care. An indefinite pause does not implement a deadline or a safe no-response default.

Borrow the idea. Separate the review payload, authorized responder, resumed state, and consequential action. Re-check authority where the action occurs. For a concrete source trace, the documentation links a public run as well as full approve/reject examples.

Source and verification. Documentation inspected; fragment parsed as Python with its missing `State` dependency understood, not executed as a complete graph.^{43, 44} **Return:** [human control](#).

43. [Interrupt documentation](#)

44. [published trace](#)

7.7 Compliance, reliance, and appropriate trust

Key concept

Appropriate reliance.

Responding to an alarm and trusting silence are different behaviors [Meyer 2004]. The aim is reliance suited to the system's capabilities and context, not maximal trust [Lee and See 2004]. Evaluate actions on false alarms and harmful misses during silence separately.

In alarm-response research, **compliance** concerns responding when automation signals a problem; **reliance** concerns depending on its silence [Meyer 2004]. Excessive compliance can produce action on false alarms; excessive reliance can leave missed events unattended. Refusing a correct alarm is another compliance problem. These behaviors cannot be inferred from a single trust score.

Sensitivity, specificity, event prevalence, consequences, and the operator's alternatives all matter. Do not assign compliance exclusively to specificity or reliance exclusively to sensitivity as if human response were determined by one parameter. Lee and See's central design goal is **appropriate reliance**: trust should track the automation's capabilities in its actual context, not be maximized as a product metric [Lee and See 2004].

Track at least action on false alarms, failure to act on true alarms, harmful misses during silence, response latency, and downstream consequence. Compare these with the workload and expertise of the human reviewer.

7.8 Fan-out as a measured workload relation

Key concept

Neglect tolerance and interaction time.

Supervisory capacity depends on how long work can proceed without attention and how much attention each intervention requires [Crandall et al. 2005]. Simultaneous exceptions and context switching can invalidate a simple average. More parallel workers do not create more parallel human attention.

Crandall, Goodrich, Olsen, and Nielsen study human-robot interaction under multitasking and introduce a useful fan-out relation [Crandall et al. 2005]:

$$FO = \frac{NT}{IT} + 1.$$

Here *NT* is neglect tolerance and *IT* is required interaction time. Under the model's scheduling assumptions, a worker that can be left alone for 15 minutes after a five-minute interaction suggests a fan-out of four. This is an illustrative calculation, not an established capacity for supervising four LLM agents or a general law of management.

Organizational tasks can synchronize their exceptions, compete for the same expert, change difficulty, and incur context-switch costs. Neglect tolerance can vary across task phases. Averages conceal deadline clusters; parallel dispatch does not create parallel human attention.



⚠ Caution – Admit work against review capacity, not runtime capacity. Measure interruption arrival rates, handling time, missed deadlines, review debt, and recovery performance for the real task mix. Test bursts and an unavailable operator. Do not import the human-robot fan-out equation's numerical result into an artificial organization without checking its assumptions and gathering local evidence.

7.9 Extending expected-loss escalation without false precision

Concept reference — attention is measurable in several different ways. These quantities answer different questions; none should substitute for the others [Meyer 2004; Crandall et al. 2005].

Quantity	Meaning	Operational implication
Prevalence	Fraction of cases in which the target event actually occurs	Rare events can produce mostly false alarms even with apparently good detection
Sensitivity	Fraction of real events detected	Describes misses, not the fraction of alerts worth reviewing
Specificity	Fraction of non-events correctly left unflagged	Small false-positive rates matter greatly at high volume
PPV	Fraction of raised alerts that are real	Helps estimate the useful fraction of review demand
Handling time	Human effort required per intervention	Turns alert volume into workload
Neglect tolerance	How long work can proceed without needed intervention under the model	Depends on task phase and environment, not just the worker
Review debt	Work awaiting required verification	Completed execution can coexist with growing unaccepted work
Response deadline	Latest point at which intervention can matter	An accurate alert can still arrive too late

Section 7.3's expected-loss comparison becomes more useful when it distinguishes whether a message arrives, whether someone responds in time, and whether that response can change the outcome. Delivery probability is not the probability that the concern is real. These are proposed extensions to the teaching model, not additional equations attributed to STEAM. A risk-class gate such as “always require authorization before deleting the only copy” may be preferable when consequences are unacceptable or the required probabilities cannot be estimated. Expected utility complements explicit constraints; it need not replace every rule.

Compare related approaches – oversight mechanisms. These methods share a concern with effective human control, but answer different questions [Parasuraman et al. 2000; Tambe 1997; McFarlane 2002; Crandall et al. 2005].

Approach	Question answered	Required inputs	What it does not decide
Stage-specific automation	Which stage may be automated?	Task decomposition, consequence, human capability	When a particular exception deserves attention
Expected-loss communication	Is informing a recipient worth its cost?	Event beliefs, preventable loss, communication cost	Who legally or institutionally may authorize the act
Signal detection / PPV	How useful and costly are alerts?	Base rates and labeled outcomes	Which interruption channel the person should receive
Interruption strategy	When and how should attention be requested?	Urgency, workflow, deadline, user availability	Whether the underlying evidence is sound
SAT transparency	What must be visible for understanding?	State, plans, constraints, uncertainty	Whether displayed explanations are causally faithful
Fan-out analysis	Can the person keep up with admitted work?	Interaction time, neglect tolerance, scheduling assumptions	A universal limit across tasks and settings

The methods are **complementary controls**, not competing scores to average. An accurately detected event can require no human decision; an authorized decision can arrive too late; and a comprehensible queue can still exceed capacity.

Reading map 11. Oversight methods: inspect the observable, not just the formula

Reading map • operational comparison grounded in the cited studies. An oversight method becomes concrete when one can name the intervention, observation, and outcome it requires.

Method or framework	Concrete artifact	Essential fields or structure
Parasuraman et al.	Stage-by-stage allocation of human and machine control	Acquisition, analysis, selection, implementation; authority at each stage
STEAM / expected-loss communication	Communication decision with estimates and consequences	What the recipient may not know; avoidable miscoordination loss; communication cost
Signal detection	Labeled confusion matrix	True positives, false positives, false negatives, true negatives, denominator
McFarlane interruption study	Interruption protocol and measured task performance	Request timing, user deferral/acceptance, competing task, outcome measures
SAT	State/plan, reasoning/constraint, and uncertainty display	What the operator can observe at each transparency level
Crandall et al.	Multitasking interaction record	Neglect tolerance, interaction time, scheduling conditions, performance
Brier scoring	Resolved forecast dataset	Event, horizon, predicted probability, outcome, score

Borrow the idea. Choose the observation that matches the question: alert counts for detection, response time for timeliness, and outcome checks for effective intervention. The alert-count plate illustrates arithmetic; [concrete example 10](#) shows a pause mechanism. Reproducing a human-factors result additionally requires the original tasks, procedure, population, and measurements.

Source and verification. This synthesized inspection guide draws on Parasuraman et al., ⁴⁵; Tambe, ⁴⁶; McFarlane, ⁴⁷; Crandall et al., ⁴⁸. SAT and Brier are cited in References. Historical apparatus and software were not run; these leads indicate exactly what a reader should inspect next. **Return:** [human oversight](#).

7.10 Further afield

Top recommended readings

Direct oversight foundations, ranked for deciding what to delegate and when to request human attention.

- 1. Parasuraman, Sheridan, and Wickens (2000), *A Model for Types and Levels of Human Interaction with Automation*.**

This framework is the clearest antidote to treating autonomy as one dial. It distinguishes information acquisition, analysis, decision selection, and action implementation in the supervisory-control tradition. Read its allocation dimensions before writing permissions or review gates. It is not a validated optimal setting for a particular LLM workflow; that still requires local evidence.

45. <https://doi.org/10.1109/3468.844354>

46. <https://www.jair.org/index.php/jair/article/view/10193>

47. https://www.interruptions.net/literature/McFarlane-HCI02_2.pdf

48. <https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=1362&context=facpub>

2. Bainbridge (1983), *Ironies of Automation*.

This influential human-factors argument explains why automating routine work can leave people responsible for difficult intervention while undermining their opportunity to stay practiced and informed. Read it against handoff and recovery, not as a general rejection of automation. It motivates testing whether the operator can actually resume understanding and control when the system needs help.

3. Tambe (1997), *Towards Flexible Teamwork*.

STEAM connects team commitments, changing situations, and communication decisions. Its decision-theoretic treatment is important because notification should depend on preventable loss and what teammates know, not merely low confidence. Read the communication reasoning with its assumptions intact; the simplified escalation rule here is not a universal interruption policy.

Broader explorations

Remembering to act, not merely remembering information. Prospective-memory research studies carrying out an intention when the appropriate occasion arises. Einstein and McDaniel's experiments asked participants to act when a target event appeared and found benefits from external aids in their conditions [Einstein and McDaniel 1990]. Retrieving a fact and noticing that now is the time to use it are different problems.

This suggests studying an oversight interface as an external memory aid: does it connect a pending duty to the event that makes action possible, or merely store another item in a queue? Test whether contextual reminders reduce missed deadlines without increasing interruption costs. The proposed agent-interface study does not inherit the original experiment's population or results, and the human findings do not show that an LLM has the same memory mechanism.

Make action and feedback intelligible. Don Norman's *The Design of Everyday Things* examines how people understand possible actions, constraints, mappings, and feedback [Norman 2013]. Read it beside an agent approval interface: can an operator tell what will happen, to which object, and whether it happened? An interface that makes an action easy to invoke may still make its consequences hard to understand. Compare the user's interpretation before action with the system's actual state afterward. This is a human-factors design question, not a guarantee that a familiar button or clear explanation creates informed consent. Consequential authority still requires an explicit institutional rule.

Attention has multiple competing demands. Wickens' "Multiple Resources and Performance Prediction" examines resource competition in human performance [Wickens 2002]. It suggests looking beyond a single count of tasks or agents. Two demands may interfere because of their sensory channel, processing stage, or response requirements; merely placing them on different screens does not establish independence. For an executive interface, compare the work of reading, remembering, deciding, and acting under realistic interruption patterns. Use measured task performance rather than importing a universal capacity number. The theory supplies dimensions for a study; the actual workload and operator population determine what the findings mean.

Seeing a display is not understanding the situation. Endsley's model of situation awareness distinguishes perceiving relevant elements, comprehending their meaning, and projecting their future status [Endsley 1995]. It is a useful companion to transparency: a detailed trace may expose observations without helping a person understand an approaching deadline or failure. Ask what the operator must know to intervene effectively, and test that understanding under changing conditions. The model is not a checklist whose mere presence proves awareness. Its value is separating different ways in which an apparently informative interface can leave the person unable to act appropriately.

Instruction also consumes working capacity. Sweller's work on cognitive load during problem solving examines the relationship between processing demand and learning [Sweller 1988]. This matters when oversight requires people to learn an unfamiliar process while handling exceptions in it. More explanatory text can help, but it can also compete with the work of diagnosing the immediate situation. Compare a worked example, an integrated explanation, and a raw trace under the same task conditions. Do not transfer a laboratory learning result into a universal interface recipe; use it to design a test of whether the presentation builds understanding or simply increases reading demand.

8 Norms, policy, and institutional facts

© **Core point — Permission, power, and evidence are different gates.** A rule may forbid an action, define an official act, or prescribe repair after a violation. Decide which mechanism is needed and which boundary enforces it.

8.1 Three distinctions that prevent policy confusion

Delegation operates within rules. This chapter explains how rules govern behavior, create institutional meaning, and require repair after a violation. Keep those purposes separate even when one service implements several of them.

A **regulative norm** obliges, permits, or forbids behavior. A **constitutive rule** determines when an event counts as an institutional fact [Searle 1995; Jones and Sergot 1996]. The InstAL greeting specimen below shows this distinction in actual source rules: an observed wave generates an institutional greeting.

Regimentation makes a forbidden institutional effect unavailable through controlled infrastructure. **Enforcement** leaves an action technically possible, detects a violation, and applies sanction or repair. Regimentation does not make physical violation impossible; it can only refuse effects within its boundary.

Power or empowerment is the capacity to create an institutional effect. **Permission** is whether exercising that power is allowed. An act can be institutionally effective yet forbidden. Technical access, discussed in Section 7.2, is a third question: whether the actor can invoke the operation at all.

Technique	Choose when	Tradeoff	Combine with
Regimentation	Harm is irreversible, illegal, security-critical, or easy to define	Blocks legitimate exceptions; boundary may be incomplete	Human exception path, action preview
Enforcement	Context matters and repair is possible	Violations occur; detection and sanction can fail	Monitoring, contrary-to-duty repair
Constitutive rules	Business status must derive from runtime events	Mapping can be wrong or stale	Versioning, provenance, tests

Technique	Choose when	Tradeoff	Combine with
Defeasible norms	Policies conflict and priority is context-sensitive	Reasoning and audit complexity	Explicit precedence and human appeal
Commitment protocol	Directed obligations and delegation matter	Lifecycle overhead	Norms, evidence, deadlines

8.2 Policy lifecycle

A policy is **not a settings string**. Record proposal, authority to legislate, effective interval, scope, version, supersession, exceptions, monitoring, violation, reparation, and retirement. This allows an auditor to determine which rule applied when an action occurred.

Contrary-to-duty

A **contrary-to-duty** rule says what must happen after a primary duty is violated: if spend exceeds the lease, freeze further spend, preserve evidence, and notify the accountable role. It is not permission to violate.

Return to Case B. Air Canada's chatbot advice and policy page presented inconsistent information to the customer.⁴⁹ A policy lifecycle must cover the interfaces that communicate a rule as well as its authoritative record. Correcting future advice and repairing an existing customer's loss are separate obligations; updating a document alone does not accomplish both.

Concrete example 12. A policy decision: an explicit default and a narrow exception

Specimen • complete small rule module. Open Policy Agent's Rego documentation, “Default Keyword,” inspected 15 September 2026. OPA is an additional concrete example of the manuscript's policy-engine pattern, not an implementation of MOISE or InstAL.

```
package example

default allow := false

allow if {
    input.user == "bob"
    input.method == "GET"
}
```

Read the artifact. The default is **deny**. Both conditions must hold for `allow` to become true. The caller supplies structured input; a separate enforcement point must consult the result. A policy that returns false does not magically prevent an application from ignoring it.

49. *Moffatt v. Air Canada*, 2024 BCCRT 149, Civil Resolution Tribunal, 14 February 2024: paragraphs 14–23 describe the chatbot evidence and policy conflict; paragraphs 24–32 explain negligent misrepresentation; paragraphs 40–44 give the remedy. Primary decision inspected 15 September 2026: <https://decisions.civilresolutionbc.ca/crt/crtd/en/item/525448/index.do>. The case does not identify an LLM architecture.

Borrow the idea. Keep the decision relation small enough to test with positive and negative inputs. This example has no expiry, delegation chain, resource scope, or institutional-power model. Adding those is a domain design task, not a property implied by the name `allow`.

Source and verification. Exact documentation specimen, formatting normalized. The source page identifies the Apache 2.0 license. Runtime evaluation status is source inspection only: the OPA executable was not available and the policy was not evaluated. No organization-wide security claim is made.⁵⁰ **Return:** [policy boundaries](#).

8.3 Common failures and rejection criteria

- **Prompt-only policy:** the same model interprets and polices itself.
- **Token fallacy:** access scope is mistaken for case-specific authority.
- **Timeless policy:** edits overwrite the history needed for audit.
- **Sanction theater:** a violation is logged but no repair or learning follows.
- **Over-formalization:** low-risk conventions acquire a legalistic state machine no one can operate, such as requiring committee approval to rename a reversible internal draft.

Reject a full normative engine for simple, local, stable rules. A deterministic validation at the action boundary is clearer. Add richer norm machinery when rules conflict, change over time, or create directed obligations.

Example

A forbidden act can still have an effect.

Insight. Prohibiting an official act and making it institutionally invalid are different rule designs, with different consequences when the rule is breached.

An employee may have institutional power to issue an official notice but be forbidden to do so without review. The notice can then be effective yet violate policy. Alternatively, the institution can define an unreviewed notice as invalid. Those are different designs: permission in the first case, constitutive validity in the second [Jones and Sergot 1996].

Institutional languages such as InstAL represent events and consequences [Padget et al. 2016]. A small service may need only explicit state transitions. Record the event, rule version, recognized effect, violation, and remedy. Logging a violation does not by itself cancel an external payment or restore an affected person's rights.

50. [OPA policy language](#)

8.4 Counts-as mappings and the four policy questions

Key concept

Constitutive and regulative rules.

A counts-as mapping explains when an observed act acquires an institutional meaning; a regulative rule concerns what is permitted, prohibited, or required. Recognizing an act is not the same as approving it. The institutional-power and counts-as sources discussed here keep those questions distinct.

Concept reference – independent governance dimensions. A system may need several of these mechanisms for the same event [Jones and Sergot 1996; Grossi et al. 2006; Singh 1999].

Concept	Question answered	Example and boundary
Technical access	Can this actor reach or invoke the operation?	Possessing an API credential does not settle institutional permission
Institutional power	Can this actor's act create the recognized effect?	A notice may be effective yet violate policy
Permission	Is exercising that power allowed here?	Approval may depend on amount, mission, and time
Obligation	What must a party do?	A deadline-bound duty requires monitoring as well as access
Constitutive rule	What counts as an institutional fact?	A qualified review counts as acceptance
Regulative norm	What behavior is required, allowed, or forbidden?	A spending prohibition need not define the meaning of a payment
Regimentation	Which effects are refused by controlled infrastructure?	A guard blocks unauthorized execution within its boundary
Enforcement	What happens when a violation is detected?	Notification, sanction, repair, or a compensating act
Contrary-to-duty rule	What must happen after a duty is breached?	Preserve evidence and initiate repair; not permission to breach

Compare the [InstAL event rules](#) with a [small policy decision](#). They make different dimensions executable.

Counts-as relations connect descriptions at different levels. Some classify an event under an existing category; others help constitute institutional effects [Grossi et al. 2006; Grossi 2007]. A signed decision in the appropriate context may count as acceptance, but an arbitrary signature elsewhere need not. Do not assume all counts-as relations have identical logical properties or can be chained transitively without conditions.

For each consequential rule, answer four separate questions:

1. Is it regulative, constitutive, or a combination with explicitly separated parts?
2. Which effects are prevented at controlled boundaries, and which violations are detected and repaired afterward?
3. Which obligations or commitments exist, and how do their lifecycle states change?
4. What authority, priority, scope, and validity determine conflict resolution?

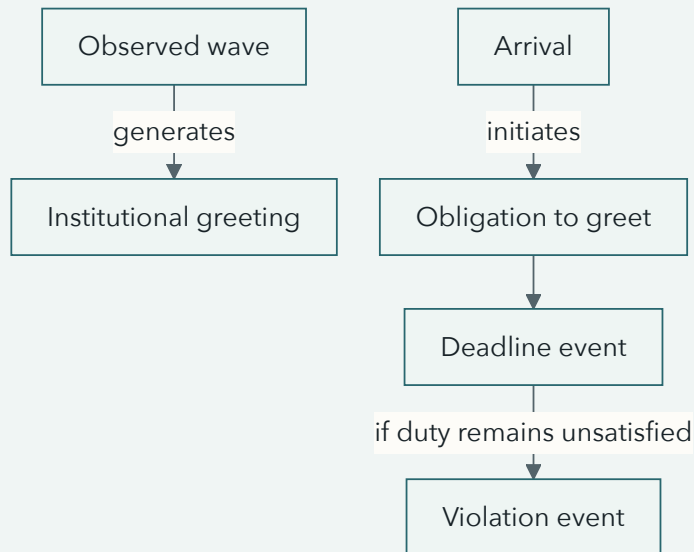
These are an engineering checklist, not four interchangeable tags. Not every rule creates a commitment or requires a new counts-as relation. In a small system, the answers may be explicit guards and records rather than a general normative reasoning engine.

Concrete example 13. InstAL: an observation becomes an institutional event

Specimen • selected rules. InstAL's `examples/greeting/greeting.ial`, commit `ffb6df61-dae1737e20e94858ad790b610d2ded82`. Line breaks are normalized; declarations and initialization rules outside the selection are omitted.

```
institution greeting;
exogenous event wave(Person);
inst event greetsroom(Person);
violation event rude(Person);
obligation fluent obl(greetsroom(Person), socialdeadline, rude(Person));
wave(A) generates greetsroom(A);
```

Read the artifact. `wave` is an externally observed event; `greetsroom` is the institutional event generated from it. The obligation names **performance, deadline, and violation**. Elsewhere in this example, entering the room initiates the obligation and permissions. Declaring the obligation's type is not the same as activating an obligation for a particular person.

FIGURE 4

Reading map. This diagram is an explanatory reduction of the located rules, not an execution trace. It separates recognition of an act from the temporal duty surrounding it. A real application must supply the event stream, domain instances, initial conditions, and the semantics of its chosen InstAL release.

Source and verification. Source inspected, not executed with the historical InstAL toolchain. This example illustrates institutional logic rather than an access-control filter.⁵¹ **Return:** [norms and institutional facts](#).

8.5 Norm conflict, institutional scenes, and governors

Key concept

Regimentation versus enforcement.

Preventing a prohibited transition and detecting or responding to a violation are different controls. Prevention cannot undo an external act that already occurred; a violation record does not itself provide prevention. Select controls according to which actions the institution can actually constrain and observe.

BOID studies relationships among beliefs, obligations, intentions, and desires, including how priorities affect agent behavior [Broersen et al. 2001]. It helps make conflicts visible; it is not a ready-made universal corporate policy engine. Familiar legal heuristics include specificity (*lex specialis*), recency among comparable authorities (*lex posterior*), and higher authority (*lex superior*). Their applicability is jurisdiction- and rule-dependent. “Newest wins” is not safe when a junior actor cannot supersede the governing rule.

51. [InstAL greeting](#)

Electronic-institution work specifies interaction **scenes**: participants, entry conditions, permitted moves, and transitions [Esteva et al. 2001, 2002]. AMELI uses mediating infrastructure, including governors associated with agents, to regulate institutional participation [Esteva et al. 2004]. This is a precedent for guarded interactions, not proof that any authenticated endpoint is an institutional governor. Authentication alone supplies neither scene state nor the authority to create an institutional effect.

The Air Canada dispute shows the consequence of presenting policy through inconsistent channels.⁵² An approval scene would need to define the eligible participants, applicable policy, and recognized disposition; a chatbot answer is not a substitute for specifying that institutional meaning. A pause/resume primitive can implement part of an interaction without defining its legal effect. This is a design implication, not a model of the tribunal's law.

Some interaction rules can be placed in shared infrastructure rather than implemented separately by every participant. Tuple centres and Reo explore that computational choice [Omicini and Denti 2001; Arbab 2004]. If such a mechanism enforces an institutional rule, changing its reactions or connections is also a policy change: version it, test affected interactions, and restrict who can install it. The medium itself does not determine who has legitimate rule-making authority.

⌕ **Design implication – Combine prevention with accountable repair.** Do not equate constitutive rules with regimentation or regulative norms with after-the-fact enforcement. A regulative spending prohibition may be blocked before execution; a constitutive validity rule can determine an act's meaning afterward. Choose the enforcement mechanism according to the controlled effect, consequence, context, and legitimate exception path.

52. *Moffatt v. Air Canada*, 2024 BCCRT 149, Civil Resolution Tribunal, 14 February 2024: paragraphs 14–23 describe the chatbot evidence and policy conflict; paragraphs 24–32 explain negligent misrepresentation; paragraphs 40–44 give the remedy. Primary decision inspected 15 September 2026: <https://decisions.civilresolutionbc.ca/crt/crtd/en/item/525448/index.do>. The case does not identify an LLM architecture.

8.6 Commitments require explicit lifecycle semantics

Social
commitments

Social commitments are directed: one party owes another a performance, sometimes conditional on an antecedent [Singh 1999; Yolum and Singh 2002]. Creation, satisfaction or discharge, cancellation, release, and violation are not synonyms. A cancelled undertaking may still incur a **duty to notify or compensate**; a **creditor's release** differs from the debtor unilaterally stopping work.

If a worker stops an accepted task, the institution still needs to determine whether the duty was released, remains outstanding, or has been violated. Inspect the terms, deadline, triggering conditions, and evidence. Compute the recorded status from those rules, while preserving the possibility that external facts are incomplete. Version the lifecycle and test cancellation during execution, release after submission, and expiry while the recipient is unavailable.

 Key concept

Directed commitments.

A commitment relates a debtor, a creditor, and content, potentially under a condition [Singh 1998]. A task message alone does not specify how that relationship is created, discharged, released, or violated. Preserve the parties and lifecycle independently of the conversation that led to the duty.

Reading map 14. Protocols and commitments: messages are not duties

Reading map • editorial synthesis. Contract Net allocates work; commitment models represent what parties owe.

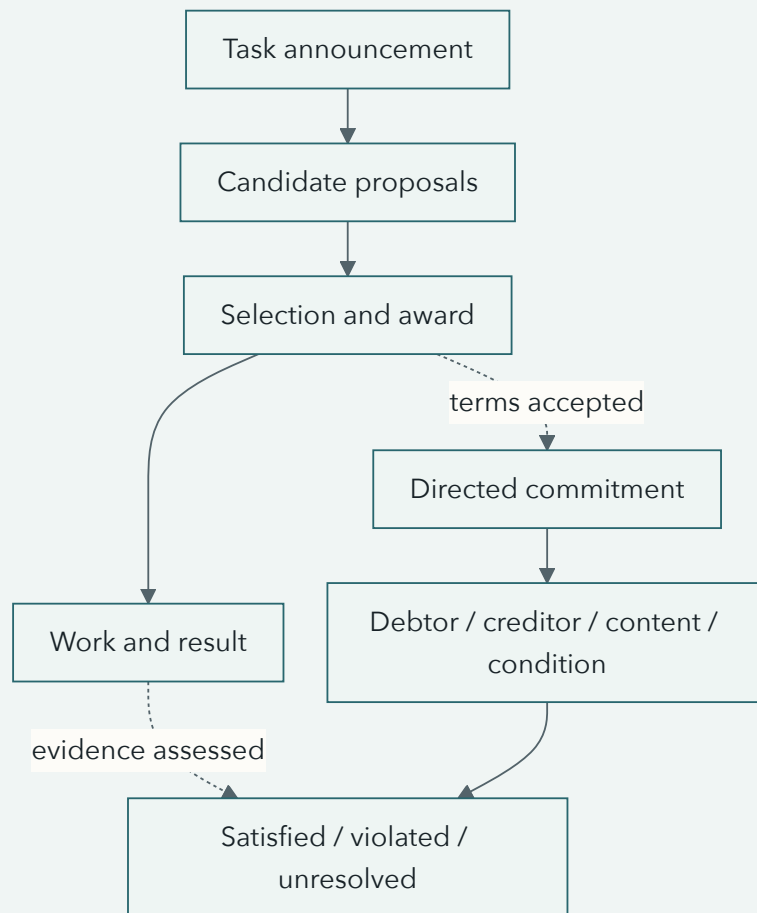
Read the map. Follow announcement, proposals, and award on the allocation path. The commitment path separately records the parties, conditions, and evidence that settles a duty. The two are **complementary, not equivalent**.

Source and verification. This is a conceptual comparison. Smith (1980), *The Contract Net Protocol*,⁵³; FIPA's communication specifications (legacy source unavailable at its former domain when checked); Yolum and Singh's *Commitment Machines*,⁵⁴. No historical deployment was executed. Compare the actual [MCP request](https://doi.org/10.1007/3-540-45448-9_17) for message structure, not commitment semantics.

Borrow the idea. Put the message record and the obligation record side by side. If the only object is a chat transcript, ask where acceptance, release, expiry, and violation are represented.

53. <https://doi.org/10.1109/TC.1980.1675516>

54. https://doi.org/10.1007/3-540-45448-9_17

FIGURE 5

Return: [coordination](#).

8.7 Further afield

Top recommended readings

Direct institutional literature, ranked for separating recognized effects, normative relationships, and executable interaction rules.

1. Jones and Sergot (1996), *A Formal Characterisation of Institutionalised Power*.

This is the crucial foundation for understanding why an effective institutional act need not be permitted. The formal distinction prevents access controls, permissions, and counts-as rules from collapsing into one ambiguous gate. Read the meaning of power and context before applying it to approvals or contracts; the account does not make every real-world effect technically preventable.

2. Singh (1999), *An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts*.

The commitment tradition shifts attention from messages or private intentions to directed social relationships. Read for the parties, content, conditions, and normative distinctions that a durable undertaking needs. Together with the chapter's lifecycle discussion, it explains why stopping a task and receiving a creditor's release are not interchangeable events.

3. Esteva and colleagues (2001), *On the Formal Specification of Electronic Institutions*.

This work brings institutional expectations into explicit interaction structures, including participant roles and regulated scenes. It is a useful bridge from the preceding concepts to inspectable specifications and later middleware. Follow which interactions the institution governs, then ask what remains outside it; a formal scene description does not itself prove compliant deployment.

Broader explorations

Rules can be effective without being regarded as legitimate. Tyler's review of legitimacy and legitimation examines why people perceive an authority or arrangement as appropriate and feel an obligation to accept its decisions [Tyler 2006]. This is a psychological and social question, distinct from whether a policy engine returns `allow`.

Explore what makes a person willing to contest, accept, or cooperate with an agent-mediated decision. Explanations, opportunities to correct the record, and routes of appeal become research variables rather than decorative features. An orderly interface might increase acceptance of a bad decision, so measure both perceived legitimacy and substantive fairness. This route concerns people affected by institutions; it does not confer legitimacy on an automated act or settle what makes that act legally or morally justified.

Rules about rules. H. L. A. Hart's *The Concept of Law* distinguishes primary rules of conduct from secondary rules concerning recognition, change, and adjudication [Hart 1961]. This offers a legal-philosophical comparison for a policy system that records prohibitions but not who can revise them or settle a dispute. Ask how an applicable rule is identified and how competing claims about it are resolved. The analogy must remain bounded: a software policy engine is not a legal system, and a database field named "valid" does not establish legal validity. The useful insight is that maintaining rules requires institutions for governing the rules themselves.

The quality of rule-making matters. Lon Fuller's *The Morality of Law* examines demands such as publicity, clarity, consistency, and the relation between announced rules and official action [Fuller 1969]. Read it to consider how a rule can fail people even when an enforcement mechanism operates exactly as implemented. Can affected actors find the rule, understand what it requires, and rely on its reasonably consistent application? These questions can inform tests of changing agent policies and exception procedures. They do not settle the justice of the rule's content, nor make technical conformance equivalent to a defensible legal or moral arrangement.

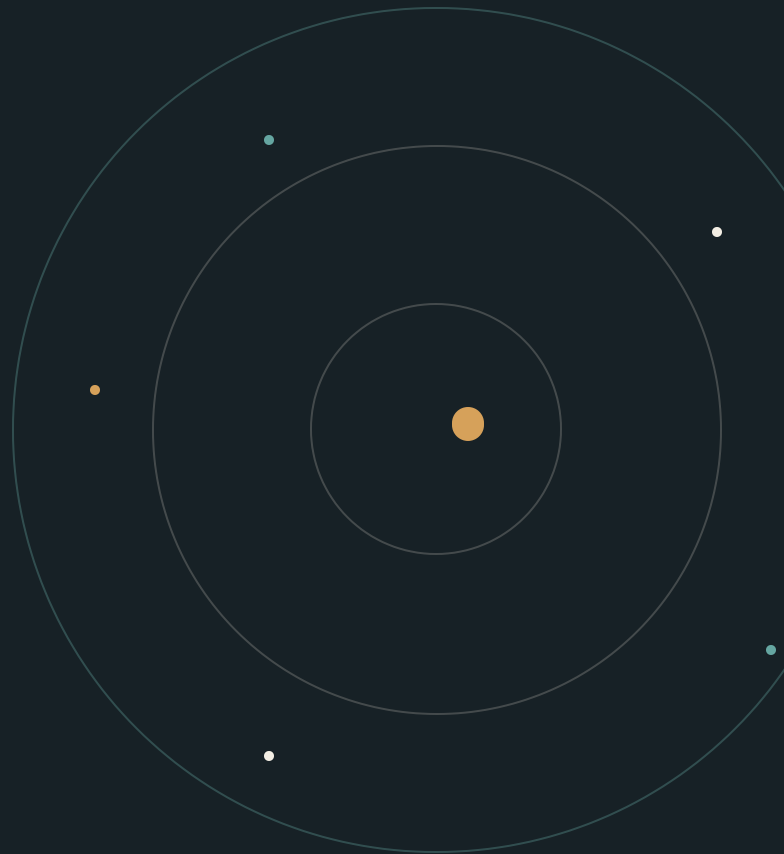
Words can be acts, under conditions. J. L. Austin's *How to Do Things with Words* develops speech-act analysis beyond a simple contrast between statements and performances [Austin 1975]. It is particularly useful when an interface appears to promise, approve, warn, or appoint. Ask what act was attempted, which conventions and circumstances matter, and whether its conditions were fulfilled. This deepens the distinction between text generation and an institutionally effective act. Do not infer that every fluent sentence changes the world in the same way: the relevant speaker, authority, context, and uptake must be established for the particular practice being analyzed.

A grammar for institutional statements. Crawford and Ostrom's "A Grammar of Institutions" distinguishes rules, norms, and shared strategies through components of institutional statements [Crawford and Ostrom 1995]. It offers an adjacent political-science route into questions often expressed computationally as permissions and obligations. Compare how a natural-language policy is parsed into actors, actions, conditions, and consequences, then inspect what the representation leaves ambiguous. A useful exercise is to apply two formalisms to the same short policy and identify disagreements. Neither grammar should be credited with eliminating interpretation or making an institution legitimate merely because the resulting representation is explicit.

PART III

Knowing and learning

Reliability, evidence, deliberation, memory, and the disciplined separation of claims from outcomes.



9 Reliability, evidence, and accountable closure

© **Core point — Verify results, not confident performance.** Preserve source and action lineage, check acceptance independently of the worker, and observe the outcome over an appropriate horizon. A good rationale replaces none of these.

9.1 Ship for measured capability, not fluent appearance

The previous chapters establish who may act and under which constraints. They do not establish that the work is sound. Here we move from measured capability to evidence, verification, and the later observation of outcomes.

Agent capability is task- and environment-dependent. The original OSWorld study evaluates 369 tasks across real computer applications using execution-based checks; it exposes difficulties in grounding actions and operational knowledge [Xie et al. 2024]. Vending-Bench tests sustained coherence while operating a simulated business. Some evaluated models usually make a simulated profit, yet all have runs that derail [Backlund and Petersson 2025]. These findings motivate trajectory-level evaluation, not a universal organizational failure rate.

Neither **simulated profit** nor **benchmark success** establishes real-world return on investment. Measure compute, integration, supervision, error, and recovery costs separately. Benchmark versions, model configurations, budgets, and evaluators change. The original study is a stable research reference, not a claim about today's leading performance.

METR estimates a roughly seven-month historical doubling in the human-equivalent task duration models complete with 50% reliability, while emphasizing uncertainty and benchmark limits [Kwa et al. 2025]. **[Established trend on the studied suite; disputed as a universal forecast.]** Design so authority can rise after representative evaluation, but do not grant tomorrow's projected capability today.

9.2 Evidence has several layers

W3C PROV distinguishes entities, activities, and agents and relates them through generation, derivation, attribution, and association [W3C 2013]. Toulmin's argument pattern distinguishes claim, data, warrant, backing, qualifier, and rebuttal [Toulmin 1958]. Subjective logic represents belief, disbelief, uncertainty, and base rate [Jøsang 2016]. They solve different problems:

- provenance answers **where did this come from?**;
- argument structure answers **why does this evidence support this claim?**;

OSWorld
Vending-Bench

METR

W3C PROV
Toulmin
argument model

- uncertainty representation states **how much support, opposition, or ignorance is being expressed?**; calibration tests whether numerical forecasts match observed frequencies.

Human-subject studies reinforce this separation. Explanations can increase agreement with both correct and incorrect LLM answers, while sources can improve resistance to wrong answers [Kim et al. 2025]. Chain-of-thought can omit causal influences or rationalize an answer [Turpin et al. 2023; Lanham et al. 2023]. Treat declared rationale as a **witness statement**, not an **audit trail**.

9.3 Silence states

Silence is an observation that needs interpretation. A quiet monitor may have found no problem, or it may not have checked. Likewise, an empty queue does not necessarily mean all work is complete. A worker may have consumed a task but not yet produced a result; a producer may be delayed; the work may be lost after a crash. Completion detection must account for active producers, in-flight work, and required results. “No matching tuple” is a local observation under an API contract, not a certificate of healthy or finished work.

At minimum distinguish:

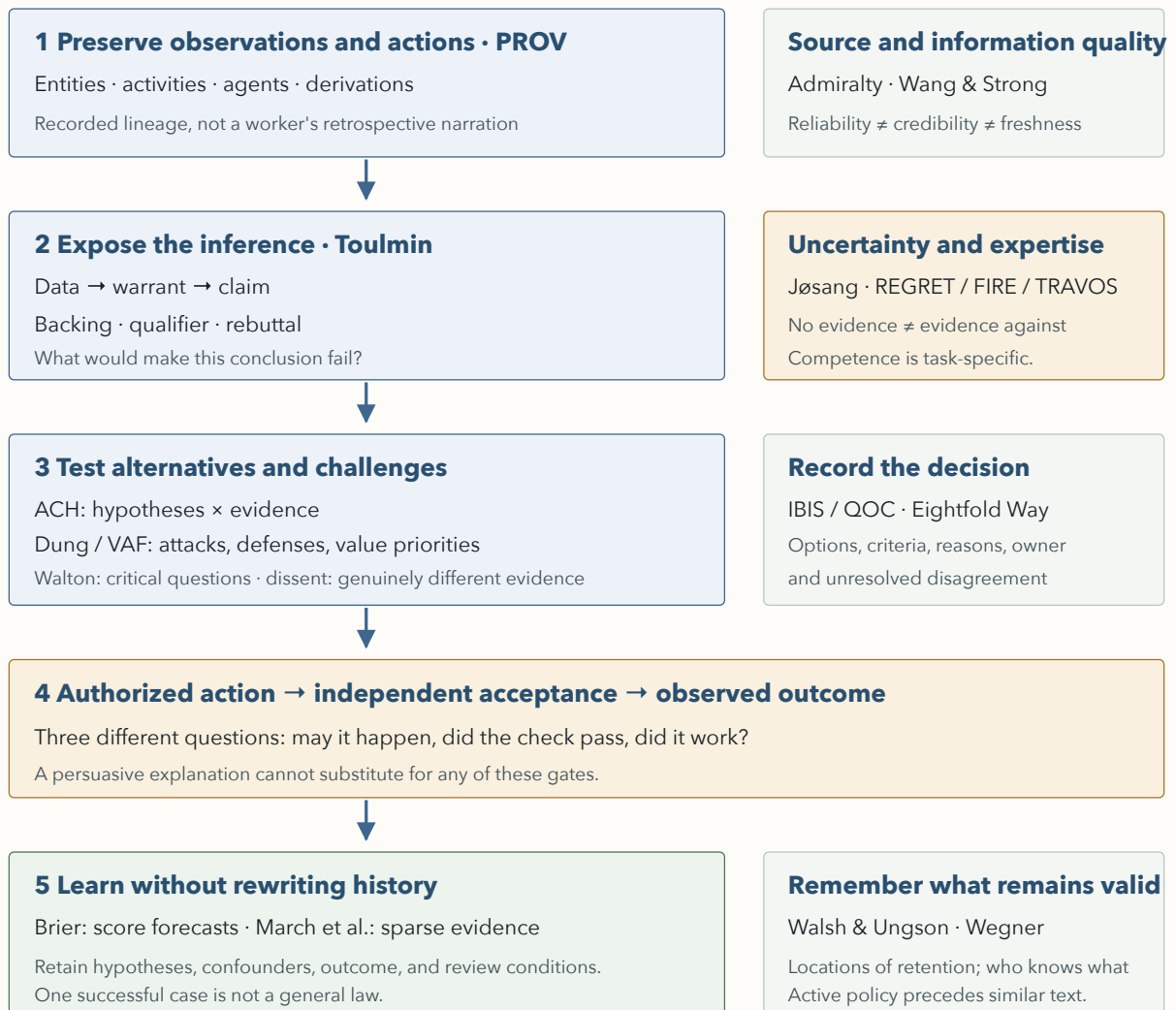
- checked, fresh, and within limits;
- checked but uncertain or degraded;
- not checked;
- check failed;
- evidence stale;
- not applicable.

Subjective logic's maximal uncertainty provides a formal way to represent “no evidence” rather than mislabeling it as neutral health. Simpler systems can use explicit enumerated states; the semantic distinction matters more than the mathematics.

🕒 **How to read Synthesis plate IV.** Follow the central path from preserved observations through explicit arguments and scrutiny to authorized action, acceptance, and observed outcomes. Read the side panels as **independent checks** on sources, uncertainty, decision records, and memory validity. The feedback statement closes the loop: later evidence can reopen a claim or policy without rewriting the earlier record. This is an editorial synthesis of PROV, Toulmin, ACH, argumentation, forecasting, and memory approaches, not a claim that any one of them implements the entire cycle [W3C 2013; Toulmin 1958; Heuer 1999; Dung 1995; Brier 1950; Walsh and Ungson 1991].

EVIDENCE → JUDGMENT → ACTION → LEARNING

A synthesis of distinct methods. Follow the blue path; use the side checks at each stage.



Feedback: later evidence can reopen a claim, decision, policy, or capability judgment.

The loop supports correction; it does not establish causality merely by preserving a sequence.

Synthesis plate IV. Evidence becomes judgment, action, and revisable institutional learning.

9.4 Completion is not outcome

 Key concept

Submission, acceptance, outcome.

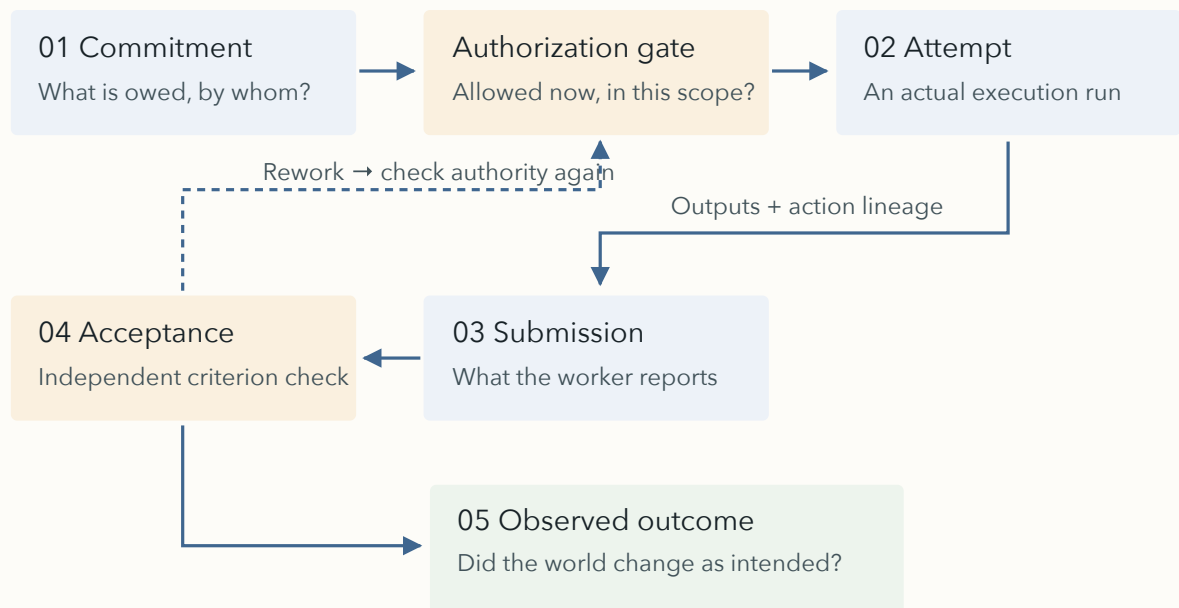
Submission records what the worker offers; acceptance records satisfaction of a review criterion; an outcome observation records what later happened. These are the monograph's distinct operational states. Conflating them lets a confident completion message masquerade as independent verification or business success.

Use separate records for **commitment**, **execution attempt**, **submission**, **acceptance**, and **observed outcome**. “Agent finished” means the run stopped. “Submitted” means an output was offered for review. “Accepted” means the designated review criterion was met. “Outcome observed” means relevant effects were measured over the specified horizon; they may confirm success, reveal failure, or remain inconclusive.

[Heuristic] Verification should scale with risk: deterministic checks for stable properties; independent model or human review for judgment; sampled audit for low-risk volume; forced review for novel, consequential, or irreversible work. Track review debt: completed work awaiting required verification.

Ⓢ **How to read Visual study D.** Blue carries work and evidence; amber marks authorization and acceptance gates; green marks later outcome observation. A retry is a new attempt under the same commitment, not a new business obligation. Rejected evidence returns for rework; a failed outcome can require a new decision even after valid acceptance. This is a design synthesis, not a universal standard lifecycle [Singh 1998; W3C 2013].

Work performed is not yet an outcome achieved



Accepted: the defined check passed. Observed: the business effect was measured.

A failed outcome can reopen the decision without rewriting the earlier evidence.

Visual study D. Work, authorization, acceptance, and outcome remain distinct.

9.5 Evidence packets: lineage, arguments, uncertainty, and quality

Key concept

Provenance is not proof.

PROV represents entities, activities, agents, and their relationships [W3C 2013]. It can establish the recorded derivation of a claim without establishing that the claim is true, its inference is warranted, or its producing act was permitted. Keep lineage and evidential evaluation connected but distinct.

Concept reference – the evidence chain. These are the objects a reviewer needs to reconstruct and assess a claim [Toulmin 1958; W3C 2013; Jøsang 2016].

Object	What it establishes or records	What it cannot establish alone
Source snapshot	The exact material inspected	That its claims are true or applicable
Provenance	Actors, activities, inputs, outputs, and their relationships	Permission, truth, or causal adequacy of an argument
Claim	A proposition put forward for acceptance	Its own evidential support
Warrant	The inference connecting data to claim	That the assumptions hold in this case
Qualifier	Strength and conditions of the conclusion	Empirical calibration without observations
Rebuttal	A condition or argument that defeats or limits the inference	Automatic resolution of the disagreement
Submission	What a worker offers for review	Independent acceptance
Acceptance	Satisfaction of the specified review criterion	The later business outcome
Outcome observation	What happened in the relevant environment	That the original act was authorized

The [PROV specimen](#) makes lineage explicit; the [argument graph](#) represents attacks. Neither is a substitute for the other.

A PROV record can express which entity **wasGeneratedBy** an activity, which inputs an activity **used**, who **wasAssociatedWith** it, what an entity **wasDerivedFrom**, to whom it **wasAttributedTo**, and which actor **actedOnBehalfOf** another [W3C 2013]. These relations support reconstruction; they do not independently prove that an input was **true** or an act **permitted**. Capture lineage at the execution boundary, then produce reader-appropriate views of the graph. A concise brief and a reconstructible audit record are complementary artifacts.

Toulmin's full argument model distinguishes **claim, data, warrant, backing, qualifier**, and **rebuttal** [Toulmin 1958]. The warrant connects evidence to a claim; backing supports the warrant; the qualifier limits the strength of the conclusion; the rebuttal identifies defeating conditions. In GitLab's case, the existence of a staging snapshot supported a recovery plan only subject to its age, completeness, and successful transfer. Staging's removal of webhooks defeated the inference that copying its main database alone would restore every necessary record.⁵⁵ This is a Toulmin-style reading of the postmortem, not a claim that its authors used that formalism.

Key concept

Warrants, qualifiers, and rebuttals.

Evidence does not connect itself to a conclusion. Toulmin's warrant states the inference, backing supports it, the qualifier bounds the claim, and the rebuttal marks defeating conditions [Toulmin 1958]. Preserving only the claim and source loses precisely what a reviewer needs to challenge the reasoning.

 **Design implication – Preserve the qualifier and rebuttal slots.** A compact decision packet should identify the claim, its evidence, the inference that connects them, its scope, and what would reverse it. Make the source lineage inspectable without forcing the reader to consume the entire graph. Do not claim that a particular packet format has a universal empirical benefit just because its slots are conceptually useful [Toulmin 1958; W3C 2013].

Concrete example 15. PROV: a trace has a grammar

Specimen • notation excerpts. W3C's *PROV-N: The Provenance Notation*, Recommendation of 30 April 2013, Examples 5 and 13. These three statements are selected to contrast detailed derivation, minimal derivation and generation.

```
wasDerivedFrom(e2, e1, a, g2, u1)
wasDerivedFrom(e2, e1)
wasGeneratedBy(e2, a1, -)
```

Read the artifact. In the first line, *e2* is derived from *e1*; *a*, *g2*, and *u1* identify the activity, generation, and usage that make the derivation more specific. The second line records less evidence about the same kind of relationship. In the third, *a1* generates *e2*; *-* marks an unspecified time. The compact syntax exposes missing detail instead of hiding it in a narrative.

Borrow the idea. Give evidence relationships identifiers and explicit missing values. Link summaries to preserved records rather than expecting prose to encode the whole audit trail. A syntactically correct trace can still describe false observations; PROV is not an authorization or truth-validation engine.

55. GitLab, "Postmortem of database outage of January 31," 10 February 2017, especially "Broken recovery procedures," "Recovering GitLab.com," and "Root cause analysis." This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.

Source and verification. The statements come from separate informative examples, rather than one complete document, and were checked against the normative publication. This is notation inspection, not a provenance-store execution. W3C retains rights in its specification; its document license is linked in the source.⁵⁶

Return to the argument: [evidence and provenance](#).

9.6 Subjective logic: ignorance is not disbelief

 Key concept

Ignorance is not disbelief.

In subjective logic, uncertain evidence and evidence against a proposition occupy different components [Jøsang 2001, 2016]. A supplier that has not been checked is not a supplier found invalid. The distinction matters even when the implementation uses explicit states rather than a numerical uncertainty calculus.

Subjective logic

A binomial subjective opinion can be represented as (b, d, u, a) , with all four components in $[0, 1]$. Belief, disbelief, and uncertainty satisfy $b + d + u = 1$; a is a base rate [Jøsang 2001, 2016]. Its projected probability is $b + au$. When $u = 1$, the opinion expresses no committed evidence for or against the proposition, and the projection defaults to a . When $d = 1$, the opinion commits fully against the proposition. Both have $b = 0$, but they mean very different things.

In the supplier example, calling both states “0% confidence” conceals the next action: obtain evidence for an unchecked license, or address the evidence of invalidity. Even without a numerical calculus, `not_checked` and `invalid` preserve that operational distinction.

 **Caution – A formal uncertainty representation does not calibrate itself.** Explain how observations produce an opinion, which base rate is used, and whether evidence sources share dependencies. Arithmetic over ungrounded confidence numbers creates precise-looking output without additional knowledge. Never display “no evidence” as verified health or as established disbelief.

Representing uncertainty is one task; checking whether judgments are well calibrated is another. When the judgment is a forecast of a defined event, later observations make that second task possible.

 More detail

Confidence needs a defined event.

A “70% likely” forecast is assessable only with an event, time horizon, and resolution rule. For a binary event, the Brier score is $(p - y)^2$: p is the forecast and y is 1 if the event occurs, 0 otherwise [Brier 1950]. Lower is better. A forecast of 0.7 scores 0.09 when the event occurs and 0.49 when it does not.

One score cannot establish calibration. Across comparable forecasts near 0.7, events should occur about 70% of the time, allowing for sampling uncertainty. Compare against a base-rate forecast: calibration alone does not establish useful discrimination. Selectively recording successes or combining unrelated task

56. <https://www.w3.org/TR/2013/REC-prov-n-20130430/>

s can make an aggregate misleading. Subjective logic represents uncertainty explicitly, but its numerical assignments still require an evidential interpretation [Jøsang 2016].

Specify resolution before forecasting. “The recovery will succeed” is too ambiguous to score. A testable proposition might name the required tables, integrity checks, and deadline. If service returns but records remain missing, the result depends on that prior definition, not on a retrospective decision to count whatever happened as success. Record forecast time so later information cannot leak into the prediction.

Score the whole set. Suppose ten forecasts are each 0.7 and seven events occur. Their mean Brier score is $(7 \times 0.09 + 3 \times 0.49)/10 = 0.21$. That frequency agrees with 0.7 in this small set, but ten observations provide weak evidence about future calibration. If the event occurs in 70% of cases regardless of the worker's information, a constant base-rate forecaster can match this performance without distinguishing easy cases from difficult ones.

Separate calibration from discrimination. Calibration asks whether stated probabilities match observed frequencies. Discrimination asks whether forecasts distinguish cases with different outcomes. Examine both across comparable task classes, with uncertainty intervals and an appropriate baseline. Do not interpret an improved aggregate score as evidence that every important subgroup improved, especially when their consequences or base rates differ.

Connect the score to a decision. A probability estimate matters only in relation to the action it informs and the losses at stake. A well-calibrated forecast does not grant authority to act, and uncertainty about the event definition cannot be repaired by displaying additional decimal places. Retain the proposition, source information, forecast, resolution, and decision together so a reviewer can tell which part of the process actually improved.

9.7 Trust is task-specific and evidence-dependent

 Key concept

Task-specific reputation.

A reliability judgment must state what activity and evidence it concerns [Sabater and Sierra 2001, 2005].

Accurate extraction does not establish sound legal judgment, and competence does not establish permission. A global trust score can conceal the very distinction needed to delegate safely.

REGRET distinguishes individual experience, social information, and an ontological dimension specifying what a reputation concerns [Sabater and Sierra 2001, 2005]. A worker's reliability on extraction is not its reliability on legal interpretation, pricing, or truthful reporting. Keep competence, compliance, and reporting quality separate when those distinctions change a decision.

Model family	Useful contribution	Required deployment evidence
REGRET	Direct and social information with task-specific reputation	Relevant interaction history and explicit reputation context
FIRE	Combines interaction, role-based, witness, and certified reputation information	Available signals, their origins, and applicable aggregation rules
TRAVOS	Reasons about reputation when information sources may be inaccurate	Outcomes against which source accuracy can be assessed
Beta Reputation System	Updates a reputation representation from positive and negative observations	A defensible outcome classification and suitable statistical assumptions

The models require different signals and assumptions [Huynh et al. 2006; Teacy et al. 2006; Jøsang and Ismail 2002]. Missing feedback, task drift, sparse observations, and incompatible interfaces can make a richer model less useful than a simpler measurement loop. Check those prerequisites before selecting it.



Design implication – Earn a trust-model claim with a working measurement loop. Name the outcome signal, task class, observation count, recency or validity rule, dependence assumptions, and decision that the score changes. Start with simpler auditable metrics if the data cannot support the richer model. Importing a library is not evidence that organizational trust is modeled.

9.8 Intelligence tradecraft and information quality

Intelligence analysis supplies useful precedents for separating observations, judgments, confidence, and source assessment. Admiralty-style grading distinguishes source reliability, conventionally A–F, from information credibility, conventionally 1–6. Unassessable categories are not ordinary points on a single numeric scale. Use the applicable institutional definitions rather than silently treating the letters and numbers as calibrated probabilities.

Intelligence Community Directive 203 (ICD 203) sets analytic standards including attention to sourcing, uncertainty, assumptions, alternatives, and distinctions between information and judgments [ODNI 2015]. Kent's discussion of estimative language shows why terms such as “probable” need shared interpretation [Kent 1964]. A local vocabulary can reduce ambiguity, but an analyst's verbal range is not a universal mapping from language to probability.

Heuer's **Analysis of Competing Hypotheses (ACH)** makes the comparison explicit: evaluate evidence against several plausible hypotheses, focus on diagnostic inconsistencies, and examine the sensitivity of the conclusion [Heuer 1999]. Do not reduce ACH to **counting contradictions** without considering evidence quality, dependence, missing observations, and the completeness of the hypothesis set. It is a structured analytic aid, not a mechanical proof of the selected hypothesis.



Key concept

Analysis of Competing Hypotheses.

Compare plausible alternatives against evidence, especially evidence that distinguishes them [Heuer 1999]. Evidence consistent with every hypothesis provides little discrimination. Diagnostic inconsistencies deserve attention, but counting them without assessing quality and dependence is not a substitute for analysis.

Information
quality

Wang and Strong distinguish intrinsic, contextual, representational, and accessibility dimensions of data quality [Wang and Strong 1996]. An accurate but stale record and a current but incomprehensible report require different repairs. Quality must be assessed relative to use, not compressed into an unexplained “high-quality” label.

Evidence question	Practical check	Failure it can expose
Can the source be relied on for this claim?	Provenance, competence, past accuracy, conflicts	A credible institution cited outside its expertise
Does this information support the inference?	Specificity, corroboration, alternatives, warrant	Reliable source, weak conclusion
Is it applicable now?	Effective time, freshness, relevant population	Accurate historical result used as a current forecast
Can the recipient understand and access it?	Terminology, representation, permissions, retained copy	Valid evidence that cannot be inspected when needed



Design implication – Give evidence several visible dimensions. Use a concise claim, source, applicability date, uncertainty, and strongest contrary evidence in the main brief. Add trust and quality dimensions when they affect action. ACH is useful for ambiguous high-stakes hypotheses, but it should not become a mandatory ceremony for questions settled by a deterministic check.

9.9 Benchmark families measure different kinds of work

Evaluation family	What the task tests	What it does not establish
OSWorld and versioned computer-use suites	Grounded interaction with applications and files	Safe long-run operation of an organization
TheAgentCompany	Professional tasks in a simulated company with tools and coworkers	The effect of adding any particular institutional layer
Vending-Bench	Coherence and economic decisions in a simulated continuing business	Real-world profit after deployment and supervision costs

Evaluation family	What the task tests	What it does not establish
SWE-bench and SWE-bench Verified	Resolving specified repository issues against an evaluator	All maintenance, product judgment, or freedom from contamination
τ-bench	Tool-using agents interacting with simulated users under domain policies	General competence outside the retail and airline task settings
METR task horizon	Human-equivalent duration of tasks achieved at a specified reliability	A universal horizon across jobs, users, or tools
Mixture-of-Agents on AlpacaEval	Evaluator-assessed response quality under a particular setup	Independence of model errors or organizational outcome accuracy

Concrete example 16. OSWorld: the task and its acceptance test are separate objects

Specimen • selected evaluator fields. The Chrome task 030eeff7-b492-4218-b312-701ec99ee0cc , OSWorld commit b138d348256078fa634fc3b73567a7337c793e6b . This selection omits postconfig and the surrounding task record; braces enclose the selected fields only.

```
{
  "func": "exact_match",
  "result": {"type": "enable_do_not_track"},
  "expected": {
    "type": "rule",
    "rules": {"expected": "true"}
  }
}
```

Read the artifact. The task asks the agent to enable Chrome's Do Not Track setting. The evaluator reads a named state property and compares it with the expected value. Setup and post-evaluation preparation are also part of the source task. **"I enabled it" is not the evaluator's success criterion.**

Borrow the idea. Pair a natural-language objective with independently inspectable state and an explicit comparator. The test demonstrates whether a setting changed, not whether websites honor the preference or whether overall privacy improved. Evaluate the property actually measured.

Source and verification. JSON and field selection checked; no browser benchmark run. The full task launches processes and needs OSWorld's environment.⁵⁷. **Return:** [evaluation and closure](#).

TheAgent
Company

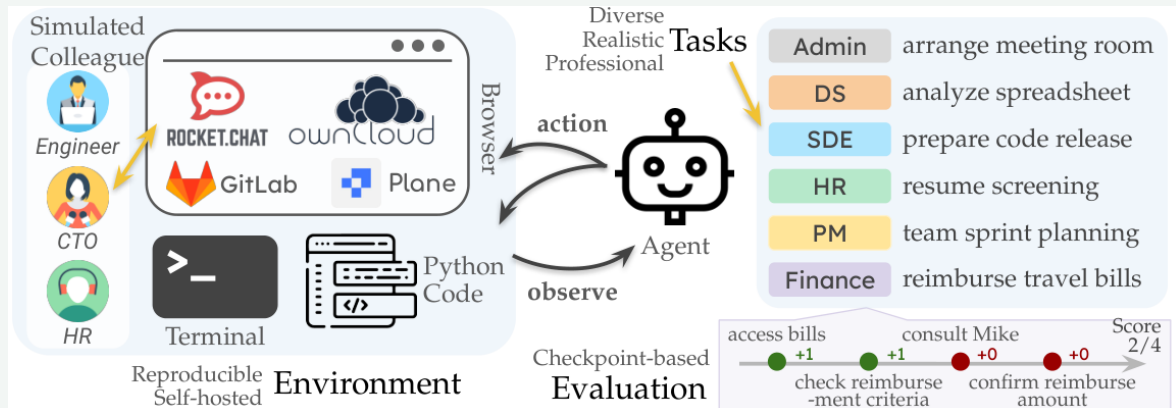
TheAgentCompany's September 2025 revision reports **30.3% full completion** for OpenHands 0.28.1 with Gemini 2.5 Pro, and 39.3% on its partial-credit score [Xu et al. 2025, v3, Table 1]. Its 175-task simulated workplace involves websites, files, programs, and communication. That is evidence of difficulty in the evaluated environment, not causal evidence that a proposed governance architecture will improve completion. The earlier model-specific results belong to their exact model, scaffold, version, and date; they must not be mixed into a current leaderboard.

57. [OSWorld task](#)

Example

What the benchmark worker actually faces.

Insight. TheAgentCompany evaluates progress toward workplace outcomes, making successful tool use only one part of completing the assigned task.



Research image R1. TheAgentCompany's published environment overview: an agent acts through a browser, terminal, and Python; simulated colleagues and business applications supply the work setting; checkpoints assess progress. Reproduced from Xu et al., version 3, Figure 1.

Follow the action and observation arrows, then the checkpoint strip. The worker must coordinate tool use, information from colleagues, and progress toward a defined result. Using an application and completing the required business task are different achievements, which is why the evaluator checks more than whether an action occurred.

The image is the researchers' overview of a runnable simulated workplace, not a photograph of an autonomous company or a successful-run screenshot. [Source, attribution, and license.](#)

Mixture-of-Agents reports 65.1% on AlpacaEval 2.0 against 57.5% for GPT-4 Omni in its published setup [J. Wang et al. 2024]. This supports the usefulness of that aggregation method on that evaluator. It does not isolate architectural diversity from additional inference effort, and it does not measure the task-error correlation needed to justify an independent-vote argument.

⚠ Caution – A dated number is not automatically a verified number. Store the model and scaffold, task version, evaluator, budget, date, denominator, and uncertainty with every benchmark comparison. Binary completion, partial credit, preference scores, and human-equivalent duration are different measures. Prefer a reproducible historical result to a recent but untraceable “state of the art” claim. Evaluate the actual organization before expanding its authority.

Concrete example 17. τ -bench: a user scenario and reference actions are not the same thing

Specimen • **three adjacent actions from the first retail test task.** τ -bench, commit 59a200c6d575d595120f1cb70fea53cef0632f6b, TASKS_TEST[0] .

```
Action(name="get_order_details", kwargs={"order_id": "#W2378156"}),
Action(name="get_product_details", kwargs={"product_id": "1656367028"}),
Action(name="get_product_details", kwargs={"product_id": "4896585277"}),
```

Read the artifact. The source task models a customer seeking two exchanges with preferences and fallback choices. Its record separates the user instruction, reference actions, and expected outputs. A neighboring task changes a fallback preference and therefore changes which items should be exchanged.

Borrow the idea. Evaluate **policy-constrained interaction**, not just one ideal tool call. The reference action sequence supports a benchmark's expected state; it is not necessarily the only legal conversational trajectory. The IDs are synthetic benchmark data, not records from this manuscript's author.

Source and verification. Source inspected; selected calls parsed inside a list, not executed against a retail environment.

⁵⁸. **Return:** [benchmark families](#).

9.10 Productivity, perception, and field evidence

Insight. In METR's early-2025 study, experienced developers believed AI tools helped even though measured task completion took longer, so perceived usefulness and productivity had to be evaluated separately.

METR's early-2025 randomized study involved 16 experienced open-source developers and 246 tasks in repositories they knew well. Allowing AI tools increased completion time by 19% in that setting, despite participants believing the tools helped [Becker et al. 2025]. The result does not show that AI slows all developers, other professions, or later tools. METR subsequently published an update and explicitly cautions that the historical result is not a statement of current capability [METR 2026]. The update identifies selection and time-use measurement problems that limit interpretation of its newer estimates; it is not simply a clean reversal of the original experiment.

This distinction matters for organizational evaluation. Benchmarks, controlled field studies, and self-reports answer different questions. A task can pass an automated evaluator while requiring substantial integration and review effort; a tool can feel helpful while increasing completion time; an autonomous agent can fail at a small bottleneck a human would quickly resolve. Measure quality, total elapsed and human time, accepted outcomes, and downstream rework together.

58. [Retail tasks](#)

9.11 Further afield

Top recommended readings

Direct evidence literature, ranked as lineage first, warranted inference second, and explicit uncertainty third. None substitutes for the others.

1. W3C (2013), *PROV-DM: The PROV Data Model*.

This Recommendation provides a standard vocabulary for entities, activities, agents, and their relationships. It matters because evidence should be reconstructable independently of a worker's retrospective explanation. Read the generation, derivation, and attribution relations alongside the book's specimen. A well-formed provenance record establishes recorded lineage, not truth, authorization, or adequacy of the inference.

2. Toulmin (1958), *The Uses of Argument*.

Toulmin's argument pattern makes explicit the warrant connecting data to a claim, along with backing, qualification, and possible rebuttal. This mid-century contribution remains important whenever a review packet preserves sources but omits the reasoning. Read it to discover what would challenge the conclusion, not to turn six labeled fields into a mechanical guarantee of sound judgment.

3. Jøsang (2016), *Subjective Logic: A Formalism for Reasoning Under Uncertainty*.

This later synthesis develops the uncertainty calculus introduced through the earlier papers discussed here. Its separation of belief, disbelief, uncertainty, and base rate makes ignorance visibly different from contrary evidence. Study the meanings and assumptions before the operators. A numerical opinion still needs appropriate evidence, dependence assumptions, and calibration where applicable.

Broader explorations

What does a score actually measure? Cronbach and Meehl's work on construct validity is a starting point for examining the relationship between an observed test score and the theoretical property it is meant to represent [Cronbach and Meehl 1955]. This perspective comes from psychological measurement, but it reaches directly into claims about agent "reliability" or "autonomy."

A completion score may be a useful measure without measuring organizational competence. Ask which observations would support the intended interpretation, which competing explanations remain, and where the test excludes relevant behavior. A research direction is to compare task scores with failures of authority, obligation continuity, and human recovery across the same systems. Those measures need their own validity argument; adding more metrics does not automatically create a valid construct.

Association, intervention, and counterfactuals answer different questions. Judea Pearl's "Causal Inference in Statistics: An Overview" introduces a formal treatment of causal questions and their assumptions [Pearl 2009]. This is useful when a provenance graph is mistaken for an explanation of what caused an outcome. Knowing that one event preceded another, or that a record was derived from an input, does not identify an intervention's effect. Ask what causal structure is assumed and which observations could challenge it. The connection is a warning against reading causality out of lineage alone, not a demand that every evidence record implement a causal-inference engine.

Design a comparison that can answer the question. Campbell and Stanley's work on experimental and quasi-experimental designs is worth returning to when evaluating an acceptance procedure [Campbell and Stanley 1966]. A before-and-after improvement may coincide with easier tasks, better workers, or a changed evaluator. Specify the comparison that would isolate the proposed contribution and list credible alternative explanations. Sometimes randomization is possible; sometimes only a narrower observational claim is defensible. The important move is to state that difference before interpreting a favorable result. A well-instrumented trace improves inspectability, but it does not on its own supply a valid counterfactual comparison.

A significant result is not a self-validating finding. Ioannidis' "Why Most Published Research Findings Are False" develops a model-based critique of how study power, prior probabilities, bias, and research practices affect the credibility of findings [Ioannidis 2005]. Read the assumptions, not just the provocative title. It offers questions for benchmark interpretation: how many variants were tried, how selectively were results reported, and how likely was the claimed effect before the study? The paper does not provide a universal false-result rate for agent research. It encourages explicit examination of the process that selected the apparently convincing evidence.

Whose testimony counts? Miranda Fricker's *Epistemic Injustice* examines wrongs done to people in their capacity as knowers, including credibility deficits connected to prejudice and gaps in shared interpretive resources [Fricker 2007]. This is an ethical complement to numerical source reliability. A system can preserve impeccable provenance while systematically overlooking the people most affected by its decisions. Ask whose observations enter the record, whose objections are intelligible within the chosen categories, and who can correct a misrepresentation. The application concerns institutional evidence practices; it must not reduce injustice to a confidence score or assume that all disagreements about credibility are equivalent.

10 Deliberation, dissent, and decisions

© **Core point — Preserve reasons that could change the decision.** Deliberation earns its cost when independent evidence or competing values matter. End with a decision record, an owner, and authorized follow-through. Agreement among similar agents is not independent corroboration.

10.1 The problem is not generating more opinions

Once evidence is inspectable, the remaining disagreement may concern an inference, a missing observation, or a priority. Deliberation earns its cost when it helps resolve that disagreement or makes the tradeoff explicit.

Argumentation frameworks represent arguments and attacks; Dung's abstract model studies which arguments can be accepted under different semantics [Dung 1995]. Issue-Based Information Systems (IBIS), Questions–Options–Criteria (QOC), and related design-rationale systems record the choices and reasons behind decisions [Kunz and Rittel 1970; MacLean et al. 1991]. Their historical problem was capture cost. Agents make capture cheaper, though whether people reuse the records remains

[OPEN].

Structured deliberation is useful when values conflict, evidence is ambiguous, or a decision is hard to reverse. It is wasteful when a deterministic check can settle the issue.

Compare related approaches – from evidence to decision. The shared concern is defensible judgment; the objects and outputs differ [W3C 2013; Toulmin 1958; Heuer 1999; Dung 1995; Bench-Capon 2003; MacLean et al. 1991].

Approach	Primary object	What it makes explicit	Typical output	Complement, not substitute
PROV	Entity, activity, actor	Lineage and responsibility relationships	Provenance graph	Supports audit; does not evaluate truth
Toulmin	A claim and its support	Warrant, backing, qualifier, rebuttal	Structured argument	Explains inference; does not compute acceptance by itself
ACH	Hypotheses and evidence	Diagnostic consistency and alternatives	Evidence-by-hypothesis matrix	Challenges confirmation; does not eliminate judgment

Approach	Primary object	What it makes explicit	Typical output	Complement, not substitute
Dung frameworks	Arguments and attacks	Formal conflict and defense	Extensions or acceptance decisions	Analyzes the supplied graph, not factual evidence
Value-based argumentation	Arguments associated with values	Dependence of defeat on value ordering	Audience-relative accepted positions	Exposes value choice; does not select societal priorities
IBIS / QOC	Issues, options, criteria, reasons	The structure of a design decision	Decision map and rationale	Preserves why; does not enforce an action
Brier scoring	Resolved forecasts	Prediction error over defined events	Scores and calibration analysis	Evaluates forecasts; not every decision is a forecast

Toulmin
argument model

Example combination: preserve sources with PROV, structure a recommendation with Toulmin, compare alternatives with QOC or ACH, and retain disagreement. Use formal acceptance semantics only when the graph and its cost are warranted.

10.2 Authentic versus assigned dissent

Authentic dissent

Nemeth, Brown, and Rogers found an advantage for authentic minority dissent in their setting [Nemeth et al. 2001]; Section 10.5 examines the comparison and its limits. For LLM teams, the relevant questions are whether critics contribute distinct evidence and whether the evaluation rewards agreement or self-preference (Section 10.7). **[DISPUTED]** Debate can improve selected judgments [Khan et al. 2024], but “more agents” is not a general accuracy theorem.

Architectural dissent uses meaningful independence: different model family where stakes justify it, separate evidence retrieval, different failure incentives, and preservation of the minority report. Disclose shared lineage. Three avatars running the same model and sources are not three independent votes.

 Key concept

Authentic dissent.

A person or agent assigned to disagree does not necessarily provide the same challenge as a genuinely held contrary position. The human evidence reviewed here distinguishes those conditions. For an artificial organization, preserve the contrary evidence and test the resulting decisions; a role called “critic” is not evidence of independence.



 Example

The IETF: answer the objection, not the headcount.

Insight. The IETF's account of rough consensus requires addressing a substantive objection even when its author is outnumbered, without granting that author an automatic veto.

Internet standards work needs a way to move forward without discarding a valid technical concern simply because few participants raise it. The IETF's RFC 7282 explains that rough consensus requires issues to be addressed, although not every requested change must be adopted. Its practical instruction is memorable: "Humming should be the start of a conversation, not the end." A room's hum helps locate disagreement; the reasons for it still need examination.⁵⁹

For an agent review, retain the objection, the evidence considered, and the reason for accepting or rejecting the proposed change. Ten approving agents do not answer one unresolved compatibility concern.

Conversely, considering an objection does not give its author an unlimited veto. That distinction makes dissent useful without requiring unanimity.

RFC 7282 is an Informational account of principles, not a new mandatory procedure or an evaluation showing that every IETF decision follows them.

10.3 A bounded decision protocol

Eightfold Way

McBurney, Hitchcock, and Parsons describe an eight-stage deliberation dialogue: open, inform, propose, consider, revise, recommend, confirm, close [McBurney et al. 2007]. A practical meeting object should define legal moves, termination, and outputs.

[Heuristic] Decision packet: question; decision owner; deadline and default; criteria; options; supporting and contrary evidence; strongest dissent; uncertainty; affected commitments; reversibility; selected action; review condition; provenance.

At **confirm**, persist the decision, owner, commitments, authority scope, and review trigger as one consistent institutional transition. Use an atomic local transaction where these records share a transactional boundary; otherwise make partial confirmation and recovery explicit. Do not dispatch consequential work from an incomplete confirmation (Section 12.11).

10.4 When to choose less

Use **one accountable decider** plus **one independent check** when stakes are moderate and time is scarce. Use a panel only when different evidence or values can change the decision. Reject debate if participants cannot produce new evidence, the decision criterion is already explicit, or delay costs dominate.

59. Pete Resnick, *On Consensus and Humming in the IETF*, RFC 7282, June 2014, especially Sections 3, 4, and 6. Section 4 supplies the quoted heading; Sections 3 and 6 distinguish addressing an objection from accommodating it or ignoring it. <https://www.rfc-editor.org/rfc/rfc7282.html>.

In Dung's abstract framework, arguments are nodes and attacks are directed relations. A conflict-free set has no internal attack. It defends an argument when it attacks each attacker of that argument; an admissible set is conflict-free and defends its own members [Dung 1995]. These properties concern the supplied graph, not the truth of its premises.



Practice note

Acceptability is not factual truth.

In the Air Canada decision, pointing to a correct policy page did not answer why a customer should distrust contradictory chatbot advice.⁶⁰ That challenges an inference about the sufficiency of another page, not the bare existence of that page. This is an analytical reading, not a reconstruction of a formal argument graph used by the tribunal. Another vote is not an answer to the challenge; formal machinery is worthwhile only when it repays its cost.

10.5 Structured conflict: what the human evidence does and does not say

Schwenk's meta-analysis supports a benefit of devil's advocacy over expert-only controls, not a universal advantage over every form of dialectical inquiry [Schwenk 1990]. Nemeth, Brown, and Rogers compare authentic minority dissent with three assigned devil's-advocacy conditions and find an advantage for authentic dissent on the studied outcomes [Nemeth et al. 2001]. Both findings have a population and a protocol. Neither proves that a prompted critic is always useless or that human minority psychology directly explains LLM behavior.

In an artificial organization, a critic can still be valuable when it performs a **concrete independent task**: check a license, reproduce a failure, seek a counterexample, or evaluate the leading option under a different criterion. Evaluate whether it adds evidence and catches errors rather than whether its persona sounds adversarial.



Caution – Do not confuse assigned disagreement with independent scrutiny. A devil's-advocate label does not establish different information or failure modes. Nor does the human dissent literature justify rejecting all artificial critics. Define the critic's evidence access, task, incentives, and acceptance test; preserve unresolved disagreement in the final packet.

60. *Moffatt v. Air Canada*, 2024 BCCRT 149, Civil Resolution Tribunal, 14 February 2024: paragraphs 14–23 describe the chatbot evidence and policy conflict; paragraphs 24–32 explain negligent misrepresentation; paragraphs 40–44 give the remedy. Primary decision inspected 15 September 2026: <https://decisions.civilresolutionbc.ca/crt/crtd/en/item/525448/index.do>. The case does not identify an LLM architecture.

10.6 Argument semantics and value disagreements

Key concept

Acceptance depends on semantics.

An abstract argument framework specifies arguments and attacks; the semantics determines acceptable sets [Dung 1995]. Skeptical and credulous inference ask about all or at least one relevant extension.

Neither formal acceptance nor a majority of agents establishes the factual truth of a premise.

Concept reference – disagreement has structure. These terms belong to different levels of analysis [Dung 1995; Bench-Capon 2003; Walton et al. 2008].

Concept	Distinction to retain	Practical consequence
Rebuttal	Challenges the conclusion	Preserve which claim is opposed
Premise attack	Challenges a supporting proposition	Check the source or observation rather than merely restating the conclusion
Undercut	Challenges the inference from premises	More instances of the same weak evidence may not repair it
Conflict-free set	Contains no internal attack in the supplied graph	Coherence does not imply factual truth
Admissible set	Conflict-free and defends its members	Specify the attack graph before computing it
Grounded extension	A uniquely determined cautious extension	Useful for conservative acceptance under that formalism
Preferred extensions	Maximal admissible sets, possibly several	Do not conceal alternative coherent positions
Skeptical / credulous acceptance	Membership in every / at least one relevant extension	State the inference mode as well as the semantics
Value ordering	Priority among values used to assess defeat	A formal result may depend on a human value judgment

Dung's semantics answer different questions about a graph [Dung 1995]. The grounded extension is a uniquely determined, cautious set under the framework's definitions. Preferred extensions are maximal admissible sets and may be multiple. Stable extensions attack every argument outside themselves and need not exist. A complete extension is an admissible set containing every argument it defends. Skeptical acceptance asks whether an argument appears in every extension of a chosen semantics; credulous acceptance asks whether it appears in at least one. "Preferred" is therefore not itself a synonym for "credulous."

Value-based argumentation associates arguments with values and evaluates defeat relative to an audience's value ordering [Bench-Capon 2003]. A lower-priority attacker may fail to defeat an argument advancing a preferred value; equal or incomparable values need the formalism's actual rule, not an invented strict ranking shortcut. The practical advantage is exposing how conclusions depend on value priorities rather than disguising a values dispute as a factual error.

Concrete example 18. Dung argumentation: an argument graph is a tiny input file

Specimen • complete graph data. TweetyProject's `ex3.tgf`, commit `fa3a23952480dbdfd9f63fc18a09dc7074ebb4a8`.

```
a
b
c
d
e
f
#
a b
b c
c d
d e
b f
e f
```

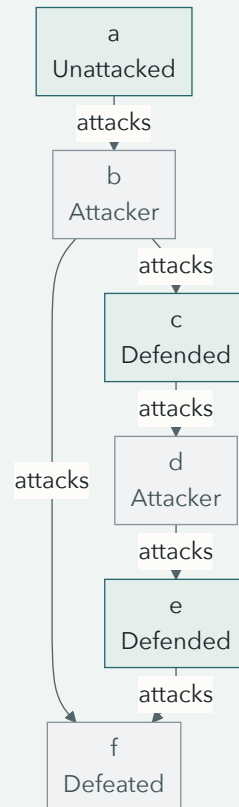
Read the artifact. Above `#` are arguments; below it are directed attacks. For example, `b f` means **b attacks f**, not that `b` causes `f`. The associated `AcceptabilityReasonerExample.java` reads this file and explicitly chooses an acceptability reasoner, semantics, and inference mode.

Reading map. Every arrow is an attack from the actual file. This graph has no attack on `a`; accepting `a` defeats `b`, enabling `c`, defeating `d`, and enabling `e`, which defeats `f`. Under grounded semantics its accepted set is `{a, c, e}`. An independent fixed-point calculation confirms that result; it does not claim the historical Java example was run. Its example runner also needs an external SAT solver configured by the user.

Borrow the idea. Store arguments and attacks independently of the reasoner. Green denotes membership in the grounded extension; the other nodes are outside it. Defense is achieved by attacking an attacker, not by adding a support arrow. These formal roles must not be confused with evidence, warr-

ants, or decision authority; a real application needs those objects as well. Changing semantics should not silently rewrite the graph. None of the letters carries factual evidence: formal acceptance is not factual truth.

FIGURE 6



Source and verification. Exact graph inspected and parsed; independent grounded-extension calculation checked. TweetyProject's source identifies LGPL-3.0.^{61, 62} **Return:** [argumentation and dissent](#).

GitLab's account makes data loss, restoration speed, and infrastructure cost visible without supplying a formal ordering over them.⁶³ A value-based analysis would have to elicit that ordering rather than infer it from whichever recovery path was feasible. The argument graph does not choose an organization's values, and an analyst must not invent its historical priorities.

61. [Graph file](#)

62. [reasoner example](#)

63. GitLab, "Postmortem of database outage of January 31," 10 February 2017, especially "Broken recovery procedures," "Recovering GitLab.com," and "Root cause analysis." This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.



Key concept

Value-based argumentation.

Defeat can depend on the audience's ordering of values [Bench-Capon 2003]. Two positions may differ because of priorities rather than evidence. Show that dependency explicitly; a formal calculation does not confer authority to choose the organization's values or override its non-negotiable constraints.

Argumentation schemes

Walton, Reed, and Macagno's argumentation schemes pair recurring inference patterns with **critical questions** [Walton et al. 2008]. For expert opinion, ask about relevant expertise, credibility, conflicts, agreement with other experts, and supporting evidence. These questions can guide review, but the reviewer must still understand the claim and domain. An automated checklist that answers every item affirmatively is not substantive scrutiny.



Design implication – Preserve the kind of challenge. Distinguish a rebuttal of the conclusion, an attack on a premise, and an undercutting of the inference. Record the strongest unresolved challenge and the evidence needed to resolve it. A readable claim graph or a carefully structured table can both work; use the least complex representation that preserves the distinctions.



Reading map 19. Decision methods: the shape of the artifact changes the question

Reading map • synthesized comparison. The methods below can organize the same decision, but their central objects differ.

Method	Artifact to construct or inspect	What the reader can see directly
Toulmin	Claim → data and warrant; backing, qualifier, rebuttal	Why the inference follows, and under what conditions it fails
ACH	Rows of evidence × columns of competing hypotheses	Which evidence discriminates among alternatives rather than merely agreeing with one
IBIS	Issue → positions → supporting/opposing arguments	What question is unresolved and which positions are under consideration

Method	Artifact to construct or inspect	What the reader can see directly
QOC	Question → options ↔ criteria	How an option fares against several explicit design criteria
Dung	Arguments + attacks + selected semantics	Which sets satisfy formal acceptability conditions
VAF	Argument graph + value associations + audience ordering	How a different priority ordering changes defeat
Walton schemes	Inference type + critical questions	What must be checked before accepting that kind of argument

Read the comparison. A matrix that contains only scores has lost the issue structure; a transcript with no explicit warrant has hidden the inference; a Dung graph with no factual support cannot establish the truth of its premises. These are different losses, not competing presentation styles. **Choose the artifact for the question you need it to answer.**

Source and verification. Heuer's ACH chapter, ⁶⁴; QOC, ⁶⁵; VAF, ⁶⁶; Walton et al., ⁶⁷. Toulmin and Kunz-Rittel are fully cited in References. The original matrices and graphical examples are research leads, not newly executed tools. [concrete example 18](#) gives an actual graph file for the argumentation family. This comparison does not measure which representation performs best. **Return:** [deliberation](#).

10.7 Sycophancy, evaluator preference, and correlated errors

 Key concept

Agreement is not independence.

Sycophancy, evaluator self-preference, and shared errors are different routes to misleading agreement [Sharma et al. 2023; Panickssery et al. 2024]. Different role names or model families do not prove independent judgment. Test error dependence and compare with strong single-worker and objective-check baselines.

Sharma and colleagues document sycophancy in evaluated assistants and evidence that human preference judgments can reward agreement with a user's beliefs [Sharma et al. 2023]. This identifies a risk in feedback and evaluation. It does not establish an unchangeable property of every model, or show that all same-family critics necessarily capitulate.

64. <https://www.cia.gov/resources/csi/static/Psychology-of-Intelligence-Analysis.pdf>
65. https://www.tandfonline.com/doi/abs/10.1207/s15327051hci0603_4_2
66. <https://doi.org/10.1093/logcom/13.3.429>
67. <https://doi.org/10.1017/CBO9780511802034>

ChatEval

Multi-agent debate, ChatEval-style evaluation, Society-of-Mind-inspired systems, and consensus-based chain-of-thought arrangements explore different interaction and aggregation protocols. Shared names do not imply identical mechanisms. Identify the concrete protocol, baseline, and compute budget before comparing claims [Du et al. 2024; Chan et al. 2024; Q. Wang et al. 2024].

✎ **Design implication – Measure architectural dissent rather than assuming it.** Compare strong single-worker baselines with same-model and heterogeneous teams on a common held-out task set. Control inference effort, prompts, retrieval, and shared sources. Estimate joint error patterns, not just average accuracy. Add diversity where it improves the decisions that matter; retain minority evidence without requiring every verbatim transcript in the main view.

10.8 Deliberation records and forecast feedback

Eightfold Way

The eight-stage deliberation model offers a protocol vocabulary, not a law that every decision must visibly traverse every stage [McBurney et al. 2007]. Missing information, unexamined alternatives, and ambiguous confirmation are useful failure diagnoses. Routine low-risk decisions can use a compressed protocol as long as their required authority and evidence remain explicit.

Probabilistic
forecasting
Brier score

Tetlock's work and forecasting-tournament research motivate evaluating forecasts over time rather than rewarding persuasive confidence [Tetlock 2005; Mellers et al. 2014]. Record the event and horizon, forecast revisions, resolution, and score. Report calibration and discrimination with sample-size caveats. Not every decision is a binary forecast; do not assign a Brier score to an outcome that was never operationally defined.

10.9 Further afield

Top recommended readings

Direct deliberation literature, ranked to distinguish formal acceptance, institutional dialogue, and empirical evidence about dissent.

1. **Dung (1995), *On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games*.**

This foundational abstract-argumentation account supplies the attack graph and acceptance semantics used throughout the chapter. Work through conflict freedom, defense, and contrasting extensions with the inspected graph. Its importance lies in making acceptance rules explicit; accepted membership in the supplied graph is not empirical confirmation of its premises.

2. **McBurney, Hitchcock, and Parsons (2007; online 2006), *The Eightfold Way of Deliberation Dialogue*.**

The dialogue tradition adds what a graph alone lacks: permitted moves and a progression toward collective action. Read the eight stages alongside the proposed meeting record, especially confirmation and closure. This formal account is a vocabulary for structuring deliberation, not evidence that every decision should incur an eight-stage ceremony.

3. Nemeth, Brown, and Rogers (2001), *Devil's Advocate versus Authentic Dissent: Stimulating Quantity and Quality*.

This experimental comparison matters because assigning a dissenting role and having a genuinely contrary position are different conditions. Read the studied tasks, outcomes, and comparison groups before borrowing the result. Its human evidence motivates scrutiny of artificial critics; it does not prove that prompted critics are always ineffective or that different models are independent.

Broader explorations

The group may already possess the answer and still miss it. In Stasser and Titus's political-caucus simulation, participants held partial information that could collectively favor a better candidate, yet discussion tended to preserve their initially biased picture [Stasser and Titus 1985]. Shared information and existing preferences could dominate the conversation instead of revealing the useful unshared facts.

This gives multi-agent researchers an unusually concrete test: distribute different decisive facts across agents, then compare free discussion with an explicit first round devoted to unshared evidence. Inspect which facts reach the decision record, not just the final answer. The human study motivates the design; it does not establish the same bias or remedy in LLM teams. Its broader challenge is whether the organization can use knowledge it already possesses.

Diversity needs a mechanism. Hong and Page model functionally diverse problem solvers and show circumstances in which diversity can outperform selection solely by individual ability [Hong and Page 2004]. Read the result with its assumptions: it is not a theorem that any heterogeneous team beats any expert. The useful question for agent deliberation is what distinct representations or search methods participants contribute. Compare teams under matched effort and outcome evaluation, and examine when their differences complement rather than merely disagree. Labels, demographic analogies, or different model vendors are not substitutes for establishing the actual problem-solving diversity that the comparison requires.

What would rational disagreement require? Aumann's "Agreeing to Disagree" establishes a result under demanding assumptions involving a common prior and common knowledge of posterior beliefs [Aumann 1976]. It is useful precisely because ordinary deliberation rarely supplies all those conditions. When two participants disagree, ask whether they share evidence, priors, event definitions, and knowledge of one another's conclusions. Do not invoke the theorem to force consensus or treat dissent as irrational. A structured review can expose which assumption differs, turning a vague dispute into a question about information or interpretation that a further observation might resolve.

Aggregation cannot satisfy every attractive condition. Arrow's *Social Choice and Individual Values* studies the difficulty of deriving a collective ordering from individual preferences while satisfying specified conditions [Arrow 1963]. This brings social-choice theory into the design of voting and ranking procedures. Ask which conditions the organization actually needs and which tradeoff its decision rule makes. A majority vote, weighted score, or judge model is not a neutral way to remove conflict. Arrow's result does not say that decisions are impossible; it makes the assumptions and limits of aggregation a subject of analysis rather than a hidden implementation detail.

Objectivity as a critical social practice. Helen Longino's *Science as Social Knowledge* examines how criticism and the social organization of inquiry contribute to objectivity [Longino 1990]. Read it beside the book's insistence on preserved dissent. Who can challenge a claim, through which forum, and what would count as a response? A review channel that records objections but never changes conclusions may offer ceremony rather than scrutiny. This is a philosophical account of scientific knowledge, not an empirical guarantee for multi-agent debate. It supplies questions about whether a deliberative arrangement permits consequential criticism instead of merely producing more text.

11 Organizational memory and learning

© **Core point — Retrieve what is valid before what is similar.** Policy, precedent, episodes, procedures, and transient context need different rules. Deliberate improvement needs evidence and review, not just a plausible retrospective story; explicit predictions make changes easier to evaluate.

11.1 Memory is more than retrieved conversation

💡 Key concept

Validity before similarity.

A relevant-looking memory can contain a superseded policy or an inapplicable precedent. This monograph's recordkeeping rule is to establish authority, scope, and effective time before using similarity to select among eligible records. A retrieval score is not a certificate that a rule governed the act in question.

Organizational memory includes both retained information and the arrangements that make it usable [Walsh and Ungson 1991; Wegner 1987]. Begin with the purpose and authority of each record; Section 11.3 then examines the wider places where knowledge persists, including people and routines.

Memory type	Concrete object suggested by the GitLab case	Retrieval rule
Working	Available recovery points and current restore progress	Recency and task relevance
Episodic	The published incident timeline and outcome	Similarity plus outcome
Role	Responsibility for data-durability testing	Role and active version
Procedural	Replication and restoration runbooks	Exact applicable procedure
Policy	Access restrictions on production and staging	Authority, scope, validity time
Decision/precedent	Why the more recent snapshot was selected	Issue, criteria, review condition

Memory type	Concrete object suggested by the GitLab case	Retrieval rule
Capability/transactive	Who understands the replication tool's waiting behavior	Demonstrated expertise and availability
External archive	Preserved postmortem and linked issue records	Provenance and retention policy

This classifies objects described in the public account; it is not GitLab's own memory schema.⁶⁸

For policy retrieval, **authority, applicability, and validity** must constrain recency and similarity: an **old but active rule** takes precedence over yesterday's obsolete discussion.

11.2 Learning without folklore

Organizations learn badly from small samples. Outcome variance, delayed effects, and selective memory create causal stories from anecdotes [March, Sproull, and Tamuz 1991]. Metrics invite displacement and gaming [Ridgway 1956; Goodhart 1984]. Agents do not need personal ambition to optimize a proxy incorrectly; objective pressure is enough.

[Heuristic] For each policy or strategy change record: hypothesis, baseline, intervention, predicted measurable outcome, confounders, review date, result, and whether to retain, revise, or reverse. Track forecast calibration rather than output volume.

Forgetting is also necessary. Preserve law, commitments, provenance, and precedent under retention rules; decay transient context; quarantine superseded instructions; and test retrieval after policy changes. Principled organizational forgetting for LLM systems remains [OPEN].

 Practice note

“Valid when?” differs from “known when?”

A policy becomes effective on Monday but reaches a local system on Wednesday. Reviewing a Tuesday action requires distinguishing the effective rule from what the operator's system knew then. Record effective intervals and recording times separately. This is a temporal-data design heuristic, not a legal excuse for acting on stale information.

Retrieve policy by subject, scope, authority, and effective time, then retrieve explanatory precedent. Retain superseded versions for audit without making them current instructions. Test late-arriving amendments and withdrawn proposals. Organizational-memory research motivates attention to retention and use [Walsh and Ungson 1991]; a similarity index does not resolve validity.

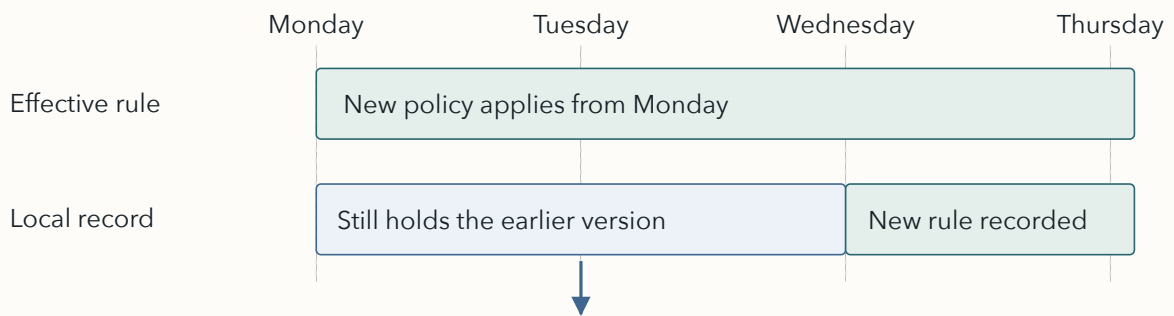
 How to read Visual study C.

Follow Tuesday vertically through the two timelines. The new rule applies, while the local system still holds the earlier record. On Wednesday the local record changes; that does not rewrite either the Tuesday rule or the Tuesday information state. The dates are an illustrative example, not a legal determination. Preserving both histories makes it possible to investigate what should have happened and why the actor acted as it did.

68. GitLab, “Postmortem of database outage of January 31,” 10 February 2017, especially “Broken recovery procedures,” “Recovering GitLab.com,” and “Root cause analysis.” This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.

Two clocks, two questions

Illustrative policy history, not a claim about a specific legal rule.



Tuesday review: which rule applied, and what information was available then?

Keep both histories. Late recording neither changes effective time nor excuses every action.

An audit needs the distinction; a similarity search cannot supply it.

Visual study C. Effective time and recording time answer different questions about the same policy history.

11.3 Retention locations are not administrative levels

Key concept

Organizational retention and transactive memory.

Knowledge persists in people, routines, structures, and other retention locations, not only documents [Walsh and Ungson 1991]. Transactive memory concerns knowing who knows what [Wegner 1987]. A searchable archive and an expertise directory support different retrieval questions; neither replaces the other.

Walsh and Ungson's retention locations are individuals, culture, transformations, structures, ecology, and external archives [Walsh and Ungson 1991]. They are not a hierarchy of "individual, role, department, company." Those organizational levels are useful implementation scopes, but represent a different analytical axis. A department can retain knowledge through procedures, shared norms, relationships, documents, and the people or agents who know how to use them.

Ecology concerns the physical setting in the original account, not simply a synonym for a database. Digital workspaces may play analogous contextual roles, but the analogy should be stated rather than attributed to the source. Transactive memory asks who knows what and who can retrieve the needed expertise [Wegner 1987]. A catalogue of skills is useful only if its entries are current and grounded in demonstrated capability.

 Example

The archive needs people and retrieval procedures.

Insight. TVA's central-files office illustrates how stored records become organizational memory through people and procedures that retrieve and interpret them.



Historical photograph H5. TVA central files, titled 1936 in the archive: staff retrieve folders, handle documents, and use a telephone among indexed cabinets. TVA Web Team, K-0897.

The cabinets retain documents; the staff and working arrangements make them usable. Organizational memory depends on connecting a question to the right record and interpreting what that record means. Indexed storage, retrieval procedures, and knowledgeable people contribute different parts of that work.

This historical office illustrates the distinction; the photograph does not establish implementation of Walsh and Ungson's later theory or the accuracy of its responses. [Image source and reuse terms.](#)

11.4 Memory systems, grounding, and the limits of retrieval

The retention functions above become useful only through concrete storage, retrieval, and revision mechanisms. Compare what each architecture preserves before considering how several workers may update the same institutional record.

MemGPT / Letta

MemGPT manages working context and external storage through an OS-inspired memory architecture; Letta develops that line into agent infrastructure [Packer et al. 2023]. Mem0 and Zep/Graphiti are examples of memory and knowledge-graph products. Their designs and releases differ. To determine which organizational records they can support, inspect their retention, consolidation, authority, and retrieval semantics the application needs.

GraphRAG
Context Rot

GraphRAG combines graph-derived structure with retrieval and summarization for query-focused answers [Edge et al. 2024]. Structured retrieval is not a substitute for provenance. Preserve the original source, transformation, policy version, and responsible actor separately from generated summaries. Chroma's Context Rot study gives a further reason to evaluate retrieval and performance as context length and composition change, rather than equating a larger context window with effective memory [Hong et al. 2025].

Compare related approaches – retention and retrieval. Similar storage technology can support very different semantics [Walsh and Ungson 1991; Wegner 1987; Packer et al. 2023; Edge et al. 2024; Shinn et al. 2023].

Approach	Common concern	Distinctive unit	Retrieval or update purpose	Main organizational caveat
Organizational retention model	Preserving useful experience	People, culture, routines, structures, setting, archives	Locate where knowledge persists	Not a database schema or six product modules
Transactive memory	Access to distributed knowledge	Who knows what	Find relevant expertise	Directory entries require evidence and maintenance
MemGPT	Limited working context	Memory tiers and explicit movement	Bring relevant material into an agent's context	Recollection is not authoritative policy
Generative Agents	Persistent agent experience	Events, thoughts, chats, salience, time	Support planning and believable behavior	Simulation believability is not institutional validity
Reflexion	Improvement across attempts	Retained linguistic feedback	Condition a later attempt	Incorrect feedback may reinforce error
GraphRAG	Synthesis across a corpus	Entities, relations, community summaries	Answer broad sensemaking queries	Generated graph structure is not complete provenance
Validity-aware policy retrieval	Correct rule application	Authoritative rule and effective interval	Retrieve the rule governing a particular act	Must distinguish effective time from recording time

One system can use several of these. **Keep the authority of a record separate from its relevance score**, regardless of whether it lives in a graph, document store, relational database, or an agent's context.

DOLCE / SUMO

Enterprise and foundational ontologies, including TOVE, DOLCE, and SUMO, provide precedents for making vocabulary explicit [Fox et al. 1996; Masolo et al. 2003; Niles and Pease 2001]. Reuse a small controlled vocabulary when that suffices. An upper ontology is not a prerequisite for every agent system, but terms with institutional consequences need stable meanings beyond a transient prompt.

✂ **Design implication – Build both the authoring and retrieval halves of institutional memory.** Record who may create, revise, supersede, and retire policies and precedents. Separate actor-local recollection from authoritative records. Query active policy by scope and validity before similarity, and test that a recent proposal cannot displace an older rule still in force.

💡 **Key concept**

Monotonic evidence, revisable decisions.

Adding an immutable observation need not invalidate the fact that an earlier observation was submitted. Declaring a policy current, a budget available, or a claim finally accepted can depend on facts not yet received. The CALM principle (consistency as logical monotonicity) connects monotonic program semantics with coordination-free distributed consistency under its assumptions [Hellerstein and Alvaro 2019]. It does not establish that every decision can avoid agreement or that communication becomes unnecessary.

Conflict-free replicated data types (CRDTs) define merge or update rules under which replicas converge [Shapiro et al. 2011]. Invariant confluence asks whether concurrent valid updates remain valid when merged [Bailis et al. 2014]. Two workers spending the same remaining capacity can violate a global limit even if their records converge. Preallocated rights can permit bounded local action, but allocation and transfer of rights must preserve the constraint. Separate a growing evidence record from the authority-bearing decision derived from it; similarity search and replica convergence answer neither question on their own.

📄 **Concrete example 20. Generative Agents: memory is a record, not an undifferentiated transcript**

Specimen • constructor assignments. `ConceptNode` in the original Generative Agents implementation, commit `fe05a71d3e4ed7d10bf68aa4eda6dd995ec070f4`.

```
self.created = created
self.expiration = expiration
self.last_accessed = self.created

self.subject = s
self.predicate = p
self.object = o
```

Read the artifact. A memory node records **time**, **access history**, and a **subject-predicate-object summary**. Other fields in this class include description, embedding key, poignancy, keywords, and links held in `filling`. The memory store keeps event, thought, and chat sequences separately.

Borrow the idea. Ask which retrieval question each field supports. An expiration field does not establish that a policy was valid when an action occurred. A memory architecture evaluated for believable simulated behavior does not automatically provide institutional recordkeeping.

Source and verification. Constructor inspected; no town simulation run. The original project is distributed under Apache 2.0; consult its license and asset terms before reuse.⁶⁹ **Return:** [memory and learning](#).

11.5 Rationale reuse: from capture to the next decision

IBIS / QOC

IBIS organizes issues and arguments; QOC structures questions, options, and criteria; DRL and systems such as SIBYL represent richer design-rationale relationships [Kunz and Rittel 1970; MacLean et al. 1991; Lee 1990, 1991]. Historical experience makes capture effort and integration into work practice central concerns [Moran and Carroll 1996]. The useful test is whether a retained rationale helps the next decision, not simply whether it can be stored.

Cheap AI-generated records solve only part of the problem. Trigger retrieval when a decision is reopened, a governing assumption changes, a role is replaced, or an observed outcome contradicts a forecast. Present the prior question, conditions, dissent, result, and applicability limit, not just a semantically similar paragraph.

 Key concept

Design rationale.

Preserve the question, options, criteria, and reasons so a later decision can reconsider the tradeoff rather than imitate its outcome [MacLean et al. 1991]. A past success is not sufficient evidence that the same choice applies under different constraints.

 **Caution – A lessons repository can grow while organizational learning declines.** Measure whether relevant rationale is found at the right moment and whether it changes a decision or avoids repeated work. Storage volume, meeting count, and number of generated “lessons” are not evidence of reuse. Where practical, capture rationale as part of the decision workflow rather than a second administrative task.

March, Sproull, and Tamuz describe learning from very few events, including richer examination of single events and consideration of hypothetical histories [March et al. 1991]. Use near misses, counterfactual explanations, comparable cases, and explicitly tentative lessons. Both human and artificial organizations can have sparse or selected records. Neither the number of workers nor the number of generated documents determines the effective sample size.

11.6 Further afield

Top recommended readings

Direct memory literature, ranked from what an organization retains to how people find expertise and reconsider decisions.

69. [Associative memory](#)

1. **Walsh and Ungson (1991), *Organizational Memory*.**

This foundational organizational account widens memory beyond stored documents to people, culture, transformations, structures, ecology, and external archives. Read the retention locations before designing a hierarchy of databases. Their conceptual contribution is not a set of administrative levels and does not imply that semantic search alone makes retained information institutionally usable.

2. **Wegner (1987), *Transactive Memory: A Contemporary Analysis of the Group Mind*.**

The transactive-memory perspective distinguishes knowing a fact from knowing who has relevant knowledge. It supplies a social foundation for thinking about expertise directories, routing, and dependence between participants. Read it alongside document retrieval: the two mechanisms answer different questions. Human group-memory theory does not by itself validate an agent directory's accuracy.

3. **MacLean, Young, Bellotti, and Moran (1991), *Questions, Options, and Criteria: Elements of Design Space Analysis*.**

QOC makes the structure of a design choice available for later examination. Its historical importance here is preserving alternatives and criteria rather than only the selected result. Reconstruct one decision and ask which changed condition would alter it. Recording a rationale is only the first step; useful retrieval and actual reuse require separate evidence.

Broader explorations

Retrieval can change what remains accessible. Anderson, Bjork, and Bjork found retrieval-induced forgetting in experiments where people practiced some members of learned categories: later recall of related unpracticed items could be impaired [Anderson et al. 1994]. This is a finding about human memory, not a claim that database retrieval erases stored records.

The provocative computational question concerns selection rather than erasure. If an agent repeatedly retrieves and summarizes the same precedents, do those precedents crowd out less familiar but relevant evidence in later decisions? Compare ordinary retrieval with deliberate sampling of overlooked cases and contradictory outcomes. A resulting feedback bias would need a computational explanation of its own; resemblance to human forgetting is a hypothesis-generating analogy, not an explanation by itself.

Not all knowledge is a retrievable sentence. Polanyi's *The Tacit Dimension* invites a closer look at what is lost when knowledge becomes a document or embedding [Polanyi 1966]. The experienced worker's ability to notice a relevant pattern may depend on practice, background distinctions, and situational cues. Ask which of those are represented in the retained artifact and which depend on the person using it. A useful study might compare successful retrieval with successful application. The aim is not to romanticize inexpressible expertise, but to avoid claiming that storage of descriptions establishes preservation of the capability those descriptions were meant to support.

Memory lives in participation as well as records. Lave and Wenger's *Situated Learning* examines learning through participation in social practice [Lave and Wenger 1991]. It adds a question that an archive alone cannot answer: how do people become able to use the organization's knowledge? When automation removes routine tasks, it may also change the experiences through which novices learn to recognize exceptions. Investigate mentoring, access to consequential work, and opportunities to compare judgment with outcomes. This concerns human learning in an organization; calling model updates “apprenticeship” does not establish the social relationships or forms of competence the authors discuss.

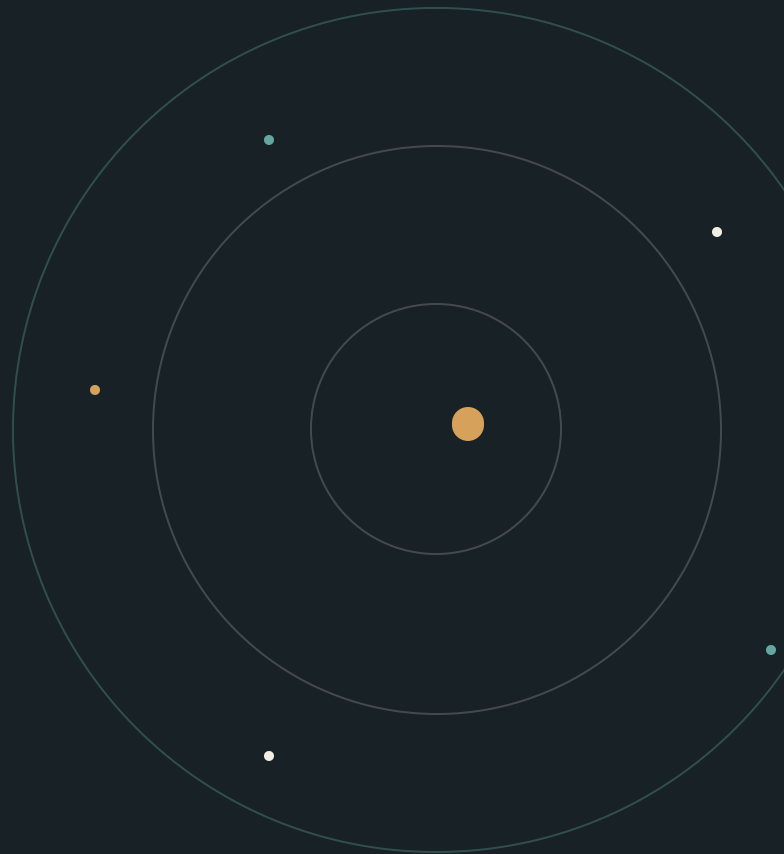
Experience becomes routines, not just memories. Levitt and March's review of organizational learning describes learning as routine-based, history-dependent, and target-oriented [Levitt and March 1988]. It connects retained experience to procedures that guide future action, including learning from other organizations. Read it to ask what changed after a lesson was recorded. Was a decision rule, search strategy, or recurring practice revised, and did that revision remain applicable? A repository full of retrospective narratives may preserve history without changing behavior. Conversely, a routine can encode past learning even when no participant can reconstruct its original rationale, creating a different problem for revision and audit.

Correct actions or revise governing assumptions? Argyris and Schön's *Organizational Learning* distinguishes correction within existing governing variables from inquiry that changes those variables [Argyris and Schön 1978]. This offers a stronger question than whether an agent remembers feedback. Does it merely improve performance against the current objective, or does the organization reconsider that objective and its rules when evidence warrants? Changes at the second level need legitimate decision procedures; a worker must not quietly rewrite policy because a task was difficult. The conceptual distinction helps separate authorized institutional learning from adaptation that merely bypasses an inconvenient constraint.

PART IV

Construction

A runtime architecture, a ten-step method, and concrete examples connecting coordination, authority, evidence, and outcomes.



12 Runtime architecture and protocols

© **Core point — Put institutional checks around the runtime.** Protocols transport requests and artifacts; they do not decide who may commit the organization. Bind execution infrastructure to policy, authority, provenance, and acceptance.

12.1 Reuse infrastructure and supply missing controls

Roles, policy, evidence, and memory describe the institutional requirements. This chapter maps them onto execution and communication infrastructure, then tests where existing components suffice and where an application must add its own guarded behavior.

Key concept

Interoperability is not authority.

A protocol can make a request intelligible without making it authorized. Tool names, message IDs, and task states support exchange; institutional scope, decision rights, and evidence requirements must be enforced at the relevant action boundary. Inspect those layers separately when evaluating a runtime.

Durable workflow engines provide mechanisms for persistence, retries, timers, and human gates. They can support idempotent processing, but preventing duplicate external effects also depends on the activity and downstream service. Agent runtimes provide model loops, tools, sessions, and traces. Identity systems provide credentials and lifecycle. Observability products provide traces and evaluations. Reimplementing these rarely creates organizational advantage.

The business-process model and its execution substrate are not the same thing (Sections 6.7–6.9). A BPMN model specifies process behavior; a durable runtime keeps execution state through interruption. A model supplied in a prompt is guidance unless an implementation checks the proposed transitions. Preserve the process version, case identity, selected branch, actor and evidence at consequential transitions; put policy and effect checks at the operation that changes the world, not only in the planner's conversation.

The Model Context Protocol (MCP) standardizes exchange between a host application, its protocol clients, and servers exposing tools and context.

Concrete example 21. MCP: an actual tool invocation is a structured message

Specimen • complete request. MCP tools specification, edition 18 June 2025, “Calling Tools.”

```
{
  "jsonrpc": "2.0",
  "id": 2,
  "method": "tools/call",
  "params": {
    "name": "get_weather",
    "arguments": {
      "location": "New York"
    }
  }
}
```

Read the artifact. The request ID correlates a response; the method selects a protocol operation; the name selects the tool; the arguments carry task data. The same specification separately shows discovery through `tools/list`, input schemas, structured output, and protocol-versus-tool errors.

Borrow the idea. Preserve the distinction between **transport correlation**, **tool selection**, and **institutional permission**. There is no spending lease or mission assignment hidden inside this weather request. An application's authorization layer must supply its own relevant rules.

Source and verification. Exact example, JSON parsed; not sent to a server. ⁷⁰ **Return:** [protocol boundaries](#).

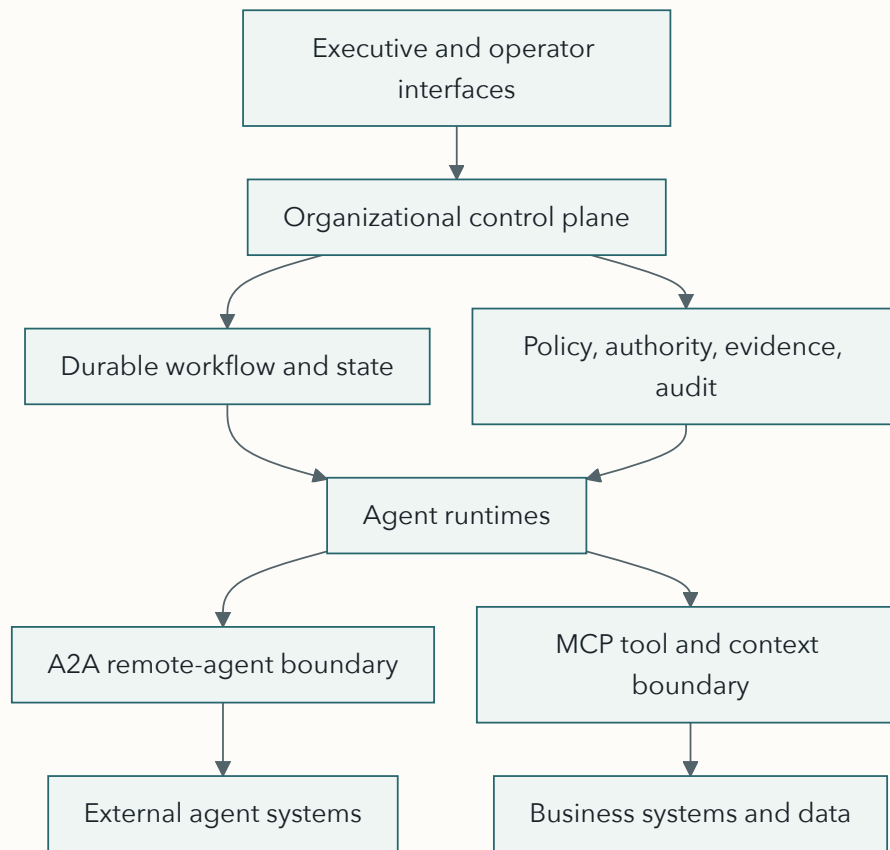
A2A

A2A 1.0 standardizes discovery, messages, tasks, artifacts, streaming, and asynchronous updates among potentially opaque agents. They are complementary. Importantly, A2A explicitly leaves the scope, validity, and revocation of in-task authorization to implementations [MCP 2025; A2A 2026].

 **How to read Figure 7.** The organizational control plane is above, not inside, the agent runtime. It owns objectives, roles, commitments, authority, and closure. Durable execution and evidence constrain runtime work. MCP and A2A connect outward at different boundaries. The conclusion is that protocols carry organizational data but do not define the organization.

70. [MCP tools, 2025-06-18](#)

FIGURE 7



Concrete example 22. A2A: a message can produce a task and an artifact

Specimen • JSON bodies from A2A 1.0.0, Section 6.1. The source sends the first body to POST /message:send , then returns a task and its artifact.⁷¹

```

{
  "message": {
    "role": "ROLE_USER",
    "parts": [{"text": "What is the weather today?"}],
    "messageId": "msg-uuid"
  }
}

```

71. Agent2Agent Project, *Agent2Agent Protocol Specification*, release 1.0.0, Sections 6.1, 8.5, and 9.4.3, inspected 15 September 2026. Apache-2.0 documentation. <https://a2a-protocol.org/v1.0.0/specification/>.

```

{
  "task": {
    "id": "task-uuid",
    "contextId": "context-uuid",
    "status": {"state": "TASK_STATE_COMPLETED"},
    "artifacts": [{
      "artifactId": "artifact-uuid",
      "name": "Weather Report",
      "parts": [{"text": "Today will be sunny with a high of 75°F"}]
    }]
  }
}

```

Read the artifact. The message, task, context, and output artifact have distinct identifiers.

ROLE_USER describes communication direction, not an institutional position. The temperature is sample content, not a verified forecast. TASK_STATE_COMPLETED is server-reported protocol state, not independent acceptance of a business result.

Version boundary. Do not mix this 1.0.0 representation with older examples using message/send, lowercase role values, or kind discriminators. JSON parsing and source matching do not test authentication or result accuracy.

Source and verification. Documentation messages with HTTP headers omitted; no weather-service request was executed for this book.

12.2 Minimal durable data model

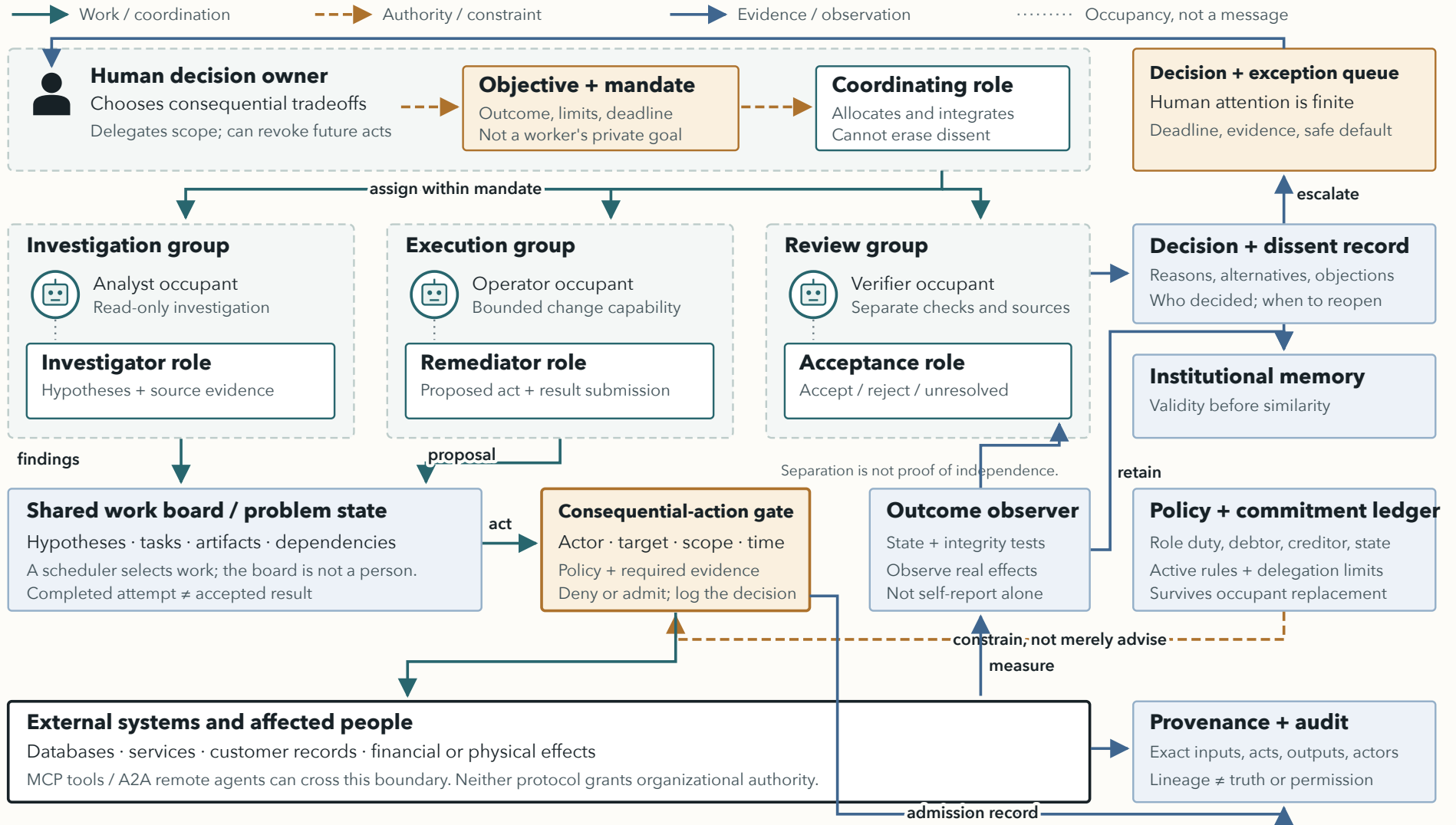
The following anatomy connects durable records to the participants and actions they govern.

Store company state **outside model context**. Give every mutation actor, timestamp, causal parent, policy decision, and idempotency key. Use append-only events where audit matters and materialized views where operation speed matters.

🕒 **How to read Synthesis plate V.** Follow the connections from mandate to action and back through evidence. At the top, a human decision owner delegates to a coordinating role. Across the middle, runtime participants occupy durable roles in investigation, execution, and review groups. Below them, findings and proposals enter shared work state; consequential actions pass through a gate into external systems. Observation returns through a separate verification path. The right-hand side retains rules, commitments, dissent, memory, and provenance outside individual participants' context. **Solid teal arrows carry work; dashed amber arrows carry authority or constraints; blue arrows carry evidence.** Dotted connectors mean role occupancy, not messages. Follow the gate-to-system-to-observer-to-review route to see why execution, acceptance, and observed outcome remain distinct.

ORGANIZATIONAL ANATOMY

Who occupies a position, who owes what, where work moves, and what can stop or verify an act.



Conceptual anatomy, not an observed org chart. Boxes are responsibilities or records, not necessarily separate services or employees.

Synthesis plate V. Organizational anatomy: participants occupy roles; work, control, and evidence connect different parts of the institution.

Start with explicit records for:

- objective and success measure;
- group, role, agent, and enactment interval;
- mission and role obligation;
- delegation and commitment state;
- policy version and applicability;
- authority lease and capability binding;
- work item, attempt, artifact, and acceptance;
- evidence entity/activity/actor lineage;
- escalation event, trigger outcome, and human response;
- decision, dissent, review condition, and observed outcome.

This is an **editorial conceptual synthesis**, not a reconstruction of GitLab's staffing or a claim that every organization needs these exact groups. The public cases motivate the boundaries: GitLab illustrates recovery ownership and observed data loss; Mars Climate Orbiter illustrates cross-team meaning; Knight Capital illustrates controls before external effects; Air Canada illustrates responsibility for public representations. The role, duty, communication, and evidence vocabulary comes from the sources developed in Chapters 5–11. Separation in the drawing does not prove independent errors or require one service per box. Return paths and routine exchanges are selectively omitted to keep the principal boundaries visible.

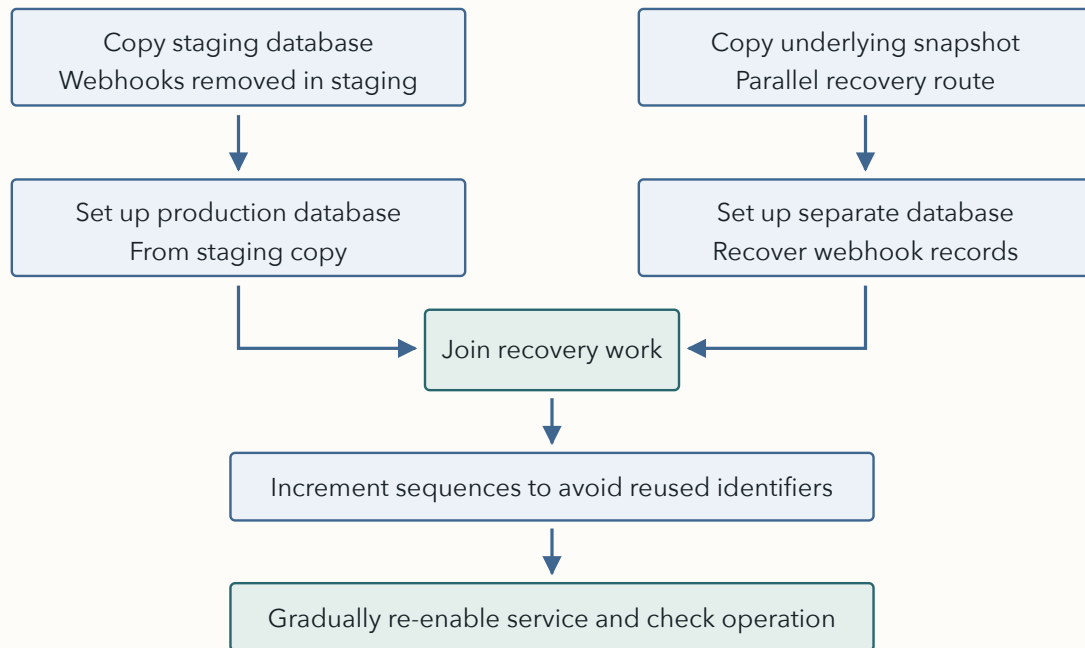
A dependency record has to distinguish its artifacts. GitLab's published recovery procedure provides a concrete contrast to the conceptual anatomy.⁷² The staging database copy supported production recovery but had webhooks removed; copying the underlying snapshot in parallel supplied a separate recovery route. The procedure set up databases from both copies, recovered the webhook records, incremented sequences to avoid reusing identifiers issued after the recovery point, and gradually re-enabled service with checks. A single “restore” status would conceal the separate artifacts, join, and verification obligations.

② **How to read Visual study D.** Follow both data paths before the join. The staging database copy supports production recovery but lacks webhook records; the underlying snapshot provides the separate path for recovering them. Sequence adjustment and service checks follow the restoration work. The diagram reduces the published procedure to its principal dependencies; it does not imply that every task had a separate worker or that restored availability recovered all changes after the snapshot.

72. GitLab, “Postmortem of database outage of January 31,” 10 February 2017, especially “Broken recovery procedures,” “Recovering GitLab.com,” and “Root cause analysis.” This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.

Two recovery paths, one incomplete recovery point

Arrows show dependencies and data movement in the published recovery account.



Service availability, restored webhooks, and changes lost after the snapshot are distinct results.

Source: GitLab's 31 January 2017 outage postmortem. This is not its original diagram.

Visual study D. GitLab's recovery combined parallel copying, separate restoration work, and a join before service checks.

12.3 Security and risk lifecycle

NIST AI RMF's Govern, Map, Measure, and Manage functions provide a useful continuous risk loop [NIST 2023, 2024]. The EU AI Act adds legal duties for covered actors and uses, including risk management, logging, documentation, human oversight, robustness, and post-market monitoring for high-risk systems [European Union 2024]. Neither supplies an organization model.

Threat-model prompt injection, confused-deputy behavior, credential leakage, cross-company data mixing, unauthorized delegation, false completion, alert flooding, provenance tampering, and roll-back gaps. Evaluate the whole system, not only the model.

 More detail

A local transaction cannot make a remote act atomic.

The payment provider receives an authorized request, but the connection drops before its response arrives. A timeout does not mean “not paid.” Use a stable idempotency key where supported, retain the intent and attempt, query the external outcome, and reconcile before sending another payment. When reversal is impossible, define compensation and escalation [Garcia-Molina and Salem 1987].

Atomically recording a local decision and delegation does not atomically commit an unrelated remote service. Revoking a lease prevents new authorized work but cannot retract every in-flight action. Check policy at admission and consequential write boundaries; preserve uncertain external outcomes. This is a systems design recommendation, not a promise of universal exactly-once execution.

Reconstruct the ambiguous history. A local record says an attempt began; the remote provider may have accepted and performed the operation; the caller then loses the response. Two worlds fit the same local timeout: no effect, or a completed effect with a lost acknowledgement. Retrying without additional information can distinguish neither world and may repeat the effect.

Use the receiver’s operation identity. Where the provider supports an idempotency key, keep the same key for retries of the same intended act and verify its scope, retention period, and treatment of changed payloads. A fresh key represents a different operation to many providers. A locally unique attempt identifier helps the audit but does not impose deduplication on a receiver that never sees or honors it.

Reconcile before declaring completion. Query the provider’s durable status where possible and compare the actual effect with the authorized intent. Record an unresolved outcome when the provider cannot answer. The local workflow may pause, escalate, or narrow subsequent work; it must not convert absence of a response into either verified success or verified failure.

Treat compensation as another consequential act. A refund, cancellation, or corrective publication is not time travel. It can fail, have costs, and require fresh authority. Saga-style reasoning tracks completed steps and compensating actions, but the application must specify which effects are reversible and which harms remain. Test the lost-response history directly; a normal request-and-response demonstration does not exercise this boundary.

12.4 From risk inventory to an executable boundary

Prompt injection crosses a trust boundary: task data attempts to become an instruction or induce an authorized intermediary to misuse its privileges. A supplier document may be evidence while having no power to change payment policy. Separate its claims from control instructions, scope tool access to the mission, and validate destinations and amounts at the action boundary. A warning in a prompt is **not an access-control mechanism**.

NIST’s generative-AI profile supplies a broader frame including information integrity, privacy, and human-AI configuration [NIST 2024]. Test an injected supplier request, altered evidence, a cross-company identifier, and stale authorization. Preserve denials and ambiguous results. Security claims should name tested controls and threats rather than promise immunity.

12.5 Contemporary architectures in their proper place

 Key concept

Acting, reflecting, and organizing are different mechanisms.

ReAct couples reasoning with actions and observations; Reflexion retains linguistic feedback across attempts [Yao et al. 2023; Shinn et al. 2023]. Neither mechanism alone supplies institutional authority or independent acceptance. Inspect the mechanism beneath a team's role names.

These are reference architectures from primary papers, not a ranking of current products or a claim about every present-day release. Each contributes a mechanism; none of the cited evaluations by itself establishes safe operation of an open-ended company.

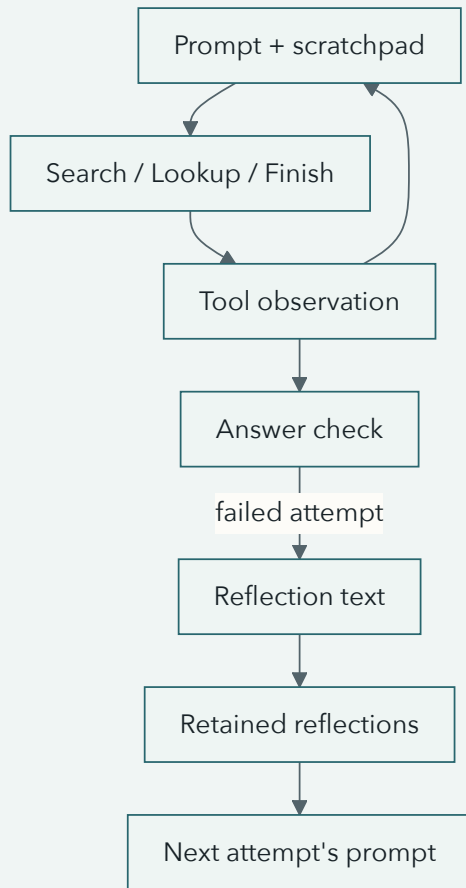
Architecture	Mechanism worth understanding	Organizational addition still needed
ReAct [Yao et al. 2023]	Interleaves reasoning and actions so observations can revise a plan	A trace is not permission; gate tools and verify outcomes
AutoGen [Wu et al. 2023]	Configurable conversations among agents, tools, and people	Define obligation owners, authority, and termination
MetaGPT [Hong et al. 2024]	Role-based collaboration and operating procedures for artifact production	Prompted roles still need durable authority, policy, and closure
Reflexion [Shinn et al. 2023]	Stores linguistic feedback for subsequent trials without updating model weights	Validate feedback and memory; distinguish self-critique from external evidence
Generative Agents [Park et al. 2023]	Observation, memory, planning, and reflection support believable simulated behavior	Believability is not correctness, legal authority, or production reliability

 **Concrete example 23. ReAct and Reflexion: observation and revision have different stores**

Specimen • reflection branch. The original Reflexion repository, `hotpotqa_runs/agents.py`, commit `218cf0ef1df84b05ce379dd4a8e47f17766733a0`.

```
self.reflections += [self.prompt_reflection()]
self.reflections_str = format_reflections(self.reflections)
```

Read the artifact. A generated reflection is appended to a list and rendered for the next prompt. The surrounding implementation keeps a separate scratchpad with **Thought**, **Action**, and **Observation** steps, and parses actions such as `Search[...]`, `Lookup[...]`, and `Finish[...]`. Reflection adds feedback across attempts; it does not update model weights.

FIGURE 8

Reading map. The diagram reduces the located implementation. It is not a trace of a new run. Notice the **answer key** available to the benchmark's check: ordinary organizations rarely possess that kind of oracle for every task. Self-critique and external feedback should not be conflated.

Source and verification. Source inspected and excerpt matched; legacy model and LangChain dependencies were not executed. Repository license: MIT.⁷³ For the original ReAct trajectories, see Yao et al., Figure 1 and the linked project examples:

⁷⁴. **Return:** [agent architectures](https://react-lm.github.io/).

ReAct

For research, a ReAct-style worker may suffice. For a production pipeline, role-based artifacts may reduce ambiguous handoffs. For simulation, believable memory-driven behavior may itself be the target. Do not transfer the evaluation criterion from one setting to another. A successful reflection loop is not yet organizational learning: retained lessons must improve future outcomes under an explicit evaluation protocol.

⁷³. [Reflexion agents](#)

⁷⁴. <https://react-lm.github.io/>

Concrete example 24. MetaGPT: a team is assembled before it is run

Specimen • selections from generate_repo. MetaGPT's `metagpt/software_company.py`, commit `11cdf466d042aece04fc6cfd13b28e1a70341b1f`. The role list is reformatted; comments, including a disabled `ProjectManager` entry, and intervening recovery logic are omitted.

```
company = Team(context=ctx)
company.hire([
    TeamLeader(), ProductManager(), Architect(),
    Engineer2(), DataAnalyst(),
])

company.invest(investment)
asyncio.run(company.run(n_round=n_round, idea=idea))
```

Read the artifact. Roles are instantiated into a team, a resource parameter is supplied, and the team runs for a configured number of rounds. The public file also has a deserialize/recovery branch. **A role class, an investment parameter, and a run limit are distinct controls;** none alone establishes legal authority or independent verification.

Borrow the idea. Inspect the artifacts produced between roles, not only the role names. The selected file shows where the team is assembled and invoked; the older `startup.py` entry point forwards to it.

Source and verification. Pinned implementation inspected, not executed. This workflow uses external model services and may create a project. MetaGPT repository license: MIT.⁷⁵ **Return:** [contemporary architectures](#).

12.6 Interoperability: inherit the distinction, not a simple origin story

KQML / FIPA

KQML and FIPA-ACL are related strands of agent communication research, but KQML was not a FIPA product [Finin et al. 1994; FIPA 2002]. They address communicative acts and interaction semantics as well as message structure. Directory services, including FIPA's Directory Facilitator specifications, help agents find services; they are not identical to a modern static Agent Card.

Concrete example 25. KQML: a query names its language and ontology

Specimen • original paper's message, whitespace normalized. Finin et al. (1994), Section 3, PDF page 4, use this stock-price query:

```
(ask-one
  :content (PRICE IBM ?price)
  :receiver stock-server
  :language LPROLOG
  :ontology NYSE-TICKS)
```

75. [Software-company entry point](#)

the paper's protocol example, not an observed market transaction.

Connection to reality. Mars Climate Orbiter's unit mismatch shows why agreeing on message structure is insufficient for shared meaning. KQML makes language and vocabulary declarations explicit, but their presence does not prove that every value is interpreted correctly.

Verification. Compared with the original paper; expression structure checked. No KQML server or LPROLOG interpreter was run.⁷⁶

MCP and A2A define lifecycle, transport, security, task, and capability conventions (Section 12.1). Schema validation addresses structure; shared interpretation and legitimate action require additional contracts. Registry freshness, identity verification, authorization, and advertised-versus-demonstrated capability remain live operational concerns.

Concrete example 26. A2A: discovery and task inspection are separate operations

Specimen • selected Agent Card fields. This object selects the name and first supported interface from Section 8.5.⁷⁷

```
{
  "name": "GeoSpatial Route Planner Agent",
  "supportedInterfaces": [{
    "url": "https://georoute-agent.example.com/a2a/v1",
    "protocolBinding": "JSONRPC",
    "protocolVersion": "1.0"
  }]
}
```

Specimen • complete task query from Section 9.4.3.

```
{
  "jsonrpc": "2.0",
  "id": 2,
  "method": "GetTask",
  "params": {"id": "task-uuid", "historyLength": 10}
}
```

Read the artifacts. Discovery advertises where and how to connect; inspection asks for existing work. The outer `id` correlates request and response; the inner `id` identifies the task. Advertising a skill does not demonstrate competence, and knowing a task identifier does not grant access.

Source and verification. The Agent Card selection omits other interfaces and required fields. Selected fields and JSON were checked against the release; sample endpoints were not invoked.

76. Finin, Fritzson, McKay, and McEntire, "KQML as an Agent Communication Language," CIKM 1994, Section 3.

https://ebiquity.umbc.edu/_file_directory_/papers/318.pdf.

77. Agent2Agent Project, *Agent2Agent Protocol Specification*, release 1.0.0, Sections 6.1, 8.5, and 9.4.3, inspected 15 September 2026.

Apache-2.0 documentation. <https://a2a-protocol.org/v1.0.0/specification/>.

Contemporary pattern	Useful historical comparison	What must not be conflated
Supervisor	Hierarchy and exception routing	A prompt-based manager is not a complete organizational model
Direct handoff	Delegation and distributed allocation	A handoff is not Contract Net if no announcement, bidding, and award occur
Group chat with speaker selection	Blackboard control and scheduling	Conversation history need not be a structured shared problem state
Agent Card or registry	Directory and matchmaking services	A signed advertisement does not establish task competence
Recursive subagents	Holarchies and nested responsibilities	Spawning children alone does not create holonic organization
Reflection with retained examples	Case-based reasoning and episodic learning	Verbal feedback is not identical to a full case-based reasoning method

PROSA and ADACOR are important holonic manufacturing precedents [Van Brussel et al. 1998; Leitão and Restivo 2006]. Their distinctions concern resource, product, order, and coordination responsibilities, including adaptation, not merely a recursion-depth setting. The practical question remains when decomposition reduces complexity and when coupled subtasks need to remain together.

⚠ Caution – Compare mechanisms, not resemblances. Use classical work to recover failure analyses and design alternatives, but do not claim that every modern handoff is a diminished Contract Net or that every group chat is Hearsay-II. Compare the actual state, control policy, commitment semantics, and information exchanged. Sometimes the simpler modern mechanism is the correct choice because allocation or rich semantics is not the problem being solved.

Concrete example 27. JSON-RPC: correlation is not business completion

Specimen • paired messages from JSON-RPC 2.0, Section 7. Source direction arrows are replaced with separate blocks.⁷⁸

```
{"jsonrpc": "2.0", "method": "subtract", "params": [42, 23], "id": 1}
```

```
{"jsonrpc": "2.0", "result": 19, "id": 1}
```

Read the artifact. Equal identifiers associate this result with its request. The protocol supplies envelopes; the application supplies the method's meaning. A notification omits `id` and receives no response under the specification. Neither correlation nor notification establishes idempotency, authorization,

78. JSON-RPC Working Group, *JSON-RPC 2.0 Specification*, Section 7, "rpc call with positional parameters," updated 4 January 2013. <https://www.jsonrpc.org/specification>.

ion, or independent verification of an external effect.

Verification. JSON, identifier correspondence, and arithmetic checked locally; no remote method invoked. MCP and A2A add semantics rather than inheriting an organizational contract from JSON-RPC.

12.7 Authoring, observability, and human intervention

Natural-language authoring can lower the entry cost of specifying roles, policies, and workflows. It does not eliminate the burden of resolving ambiguity, validating constraints, maintaining shared vocabulary, or making revisions safe. Treat generated formal structure as a proposed compilation that requires inspection and tests, not as legislation automatically authorized by fluent prose.

✎ **Design implication – Make specification authoring useful from the first bounded rule.** Show the interpreted scope, affected actors, decision examples, rejected cases, and consequences of changing a rule. Preserve versions and reversibility of configuration. If humans can edit both natural language and formal representations, declare which is authoritative and how divergence is detected; unrestricted two-way editing can create two incompatible policies.

Contemporary runtimes and tools provide substantial infrastructure. LangGraph, CrewAI, AutoGen, the OpenAI Agents SDK, Semantic Kernel, Anthropic's agent tools, and Amazon Strands illustrate runtime and orchestration families. Their versions, primitives, and lifecycle guarantees differ. A toolkit's omission of a first-class organizational object does not prove that applications built with it cannot implement one.

LangSmith, Langfuse, Braintrust, Comet Opik, provider tracing, Semantic Kernel telemetry, and DSPy-related evaluation workflows illustrate observability and evaluation options. Check whether the chosen release records the inputs, outputs, timing, policy decisions, and errors needed for the actual audit. A trace is not automatically an institutional record, and a model's emitted reasoning is not automatically a faithful causal account.

Durable workflows such as Temporal, Restate, and Inngest, browser-automation tools such as browser-use and Skyvern, and policy or guardrail tools such as Guardrails AI, NeMo Guardrails, and Granite Guardian address different pieces of execution. Sandboxes, branch-per-attempt workflows, approval hooks, interrupts, and replay can be useful containment mechanisms. None automatically defines who may approve or whether replay is safe for external side effects.

✎ **Design implication – Reuse infrastructure, specify the institutional contract.** Before building another runtime, inventory available execution, identity, storage, tracing, human-gate, and policy capabilities. Then test the missing organization-specific behavior: scope, obligations, validity intervals, acceptance, decision ownership, and evidence continuity. Prefer integration when it satisfies those tests; do not force a bespoke layer where existing enterprise systems already provide the needed controls.



AutoGen
LangGraph

Temporal /
durable
workflows



Concrete example 28. OpenAI Agents SDK: a handoff is visible in the program

Specimen • official “Basic usage” example, reformatted. A triage agent can hand control to billing or refund specialists.⁷⁹

```
from agents import Agent, handoff

billing_agent = Agent(name="Billing agent")
refund_agent = Agent(name="Refund agent")

triage_agent = Agent(
    name="Triage agent",
    handoffs=[billing_agent, handoff(refund_agent)]
)
```

Read the artifact. The triage agent has two explicit destinations. The SDK represents handoffs as tools available to the model. This declaration alone neither runs a conversation nor grants refund permission.

Connect to the public case. Air Canada's dispute concerns responsibility for customer-facing advice. Routing to a refund-named agent does not settle that responsibility or establish the applicable policy. Inspect history, authority checks, and consequential operations separately.

Source and verification. Names are the documentation's illustrative fixtures. Python syntax and source statements checked; no model call or refund workflow executed.

12.8 Failure taxonomies and adversarial evaluation

MAST identifies 14 failure modes grouped around system design, inter-agent misalignment, and task verification [Cemri et al. 2025]. Its reviewed revision distinguishes the traces used to develop the taxonomy from the larger MAST-Data collection: version 3 describes 150 traces used to develop the taxonomy and a collection of 1,642 annotated traces across seven frameworks. A taxonomy derived from selected traces is **not an estimate of failure prevalence** in all deployments. Use it to design tests for bad decomposition, unproductive interaction, missing verification, and incorrect termination, then measure incidence locally.

79. OpenAI Agents SDK for Python, “Handoffs,” “Basic usage,” inspected 15 September 2026. <https://openai.github.io/openai-agents-python/handoffs/>.

 Example**An edit reported but not applied.**

Insight. In the published HyperAgent trace, a confident report of a completed edit conflicted with the executor's test, showing why completion claims need artifact-level verification.

In a recorded HyperAgent attempt on Flask issue 4992, the editor reported: "The modification has been successfully applied to the `from_file()` method in the `src/flask/config.py` file." The executor subsequently reported that its test failed because `from_file()` did not recognize the proposed `mode` parameter [Cemri et al. 2025, v3, Appendix N.13].

The editor's claim and the executor's observation disagree. An acceptance check needs the changed artifact and test result, not just the message saying the work is done. This **claim-to-state mismatch** comes from the researchers' published trace, not a production incident or a run reproduced here; it does not establish that every missing edit has the same cause.

AgentDojo
CaMeL

AgentDojo supplies an extensible environment for prompt-injection evaluation with tool-using agents and untrusted data [Debenedetti et al. 2024]. CaMeL separates control and data flow and applies capabilities to constrain data movement [Debenedetti et al. 2025]. Its security guarantees concern the modeled system, policy, and threat assumptions; they are not a claim that every application assembled around it is immune to every attack.

Agentic
misalignment

Anthropic's agentic-misalignment research deliberately stress-tests models in controlled fictional corporate situations [Lynch et al. 2025]. It demonstrates risks under the constructed conditions, not observed prevalence in real companies. The distinction matters when reasoning about **drift** versus goal-conflicting behavior. Some failures arise without strategic intent through context changes, stale state, tools, or distribution shift; others may involve objective pursuit that conflicts with constraints. Diagnose the mechanism; state drift and goal conflict should not receive the same treatment by default.

 **Caution – Separate ordinary failure, malicious input, and goal conflict.** They can produce similar surface traces but require different tests and mitigations. State hygiene and reset may help accumulated-context failures; they do not replace access control, independent evidence, or evaluation of behavior under conflicting incentives. A natural-language prohibition alone is not an adequate safety boundary.

12.9 The commercial landscape as a capability audit

The relevant comparison is not “which product calls itself an organization?” but “which responsibilities, records, and guarantees can this deployment actually supply?” Keep public product claims, tested capabilities, and inferred market gaps separate. The following landscape is a reading guide, not an endorsement or a verified comparative product trial.

Product or category	Why a reader should investigate it	Verification boundary
Workday Agent System of Record	Agent inventory, lifecycle, organizational ownership	Validate current documentation and licensed capabilities; do not infer them from the product name
ServiceNow AI Control Tower	Discovery, governance, observability, risk, and business-context integration	Public claims do not establish complete control over every external agent
Microsoft Agent 365	Agent identity, administrative oversight, and ecosystem integration	Confirm release, availability, supported identity surfaces, and actual policy enforcement
OpenAI Frontier	Shared business context, agent execution, evaluation, permissions, and deployment integration	The introduction describes broader enterprise ambitions than “developer-only runtime”; deployed guarantees need testing
Relevance AI	Agent-team and workflow authoring for business processes	Evaluate persistence, review, export, authority, and failure handling in the intended workflow
Twinorg AI and AgentOrg	Organization-oriented product framing and potential prior art	Maturity and delivered capabilities remain unverified here

ServiceNow, Microsoft, and OpenAI explicitly describe capabilities overlapping organizational governance and business context [ServiceNow 2026; Microsoft 2026; OpenAI 2026]. Microsoft's documentation states commercial general availability from 1 May 2026, with licensing prerequisites; availability is not a measured organizational outcome. It would therefore be premature to infer a market gap from terminology alone. Conversely, a marketing description does not establish that a product satisfies the institutional and operational tests in this chapter.

Existing accounting, enterprise resource planning (ERP), human-resources information systems (HRIS), workflow, and communication tools may already own the relevant records. A new executive layer must justify integration cost, duplicated records, operator attention, and maintenance burden. A feature gap is a hypothesis to test, not an automatic business case. If the organization uses Workday, ServiceNow, NetSuite, or another authoritative record, an additional application should integrate with that record rather than silently create a competing one.



 **Design implication – Buy, integrate, or build against an acceptance test.** Ask whether a candidate preserves obligations through replacement, scopes authority by consequence, retains dissent, resolves active policy, and makes outcomes independently inspectable. Test uncertain claims in an isolated environment. An organization-themed interface is useful only if it improves decisions and operations over the available alternatives.

An inspectable product comparison. As an editorial evaluation protocol, keep one record per tested requirement: product and version, configuration, actor and authority, starting state, attempted operation, observed result, retained evidence, and unresolved limitation. Separate an advertised feat-

ure from an observed behavior. For recovery, interrupt a disposable task after an external effect but before its acknowledgment; inspect what the resumed system knows before allowing a retry. For authority, revoke permission while work is pending and observe the next attempted consequential action. Run these tests only in an isolated environment with reversible effects. A pass establishes behavior for that configuration and scenario, not universal safety. A failure should identify the unmet requirement rather than become a general verdict about the vendor. The comparison then supports a specific purchase, integration, or implementation decision that another reviewer can inspect.

Practical resources (vendor claims, not independent evaluations): AgentOrg⁸⁰; MCP announcement⁸¹; Cognition's design argument⁸²; A2A announcement⁸³; Microsoft Agent 365⁸⁴; OpenAI Frontier⁸⁵; Relevance AI⁸⁶; ServiceNow AI Control Tower⁸⁷; Twinorg AI⁸⁸; Workday agent records⁸⁹

12.10 Historical failures as design hypotheses

Earlier work also reveals costs: rich semantics can be difficult to author, trust models can lack outcome feedback, and lessons repositories can be poorly connected to later decisions. Their prevalence and causes differ by system. These are specific risks to investigate, rather than reasons to declare the whole tradition unsuccessful. Classical MAS produced working industrial and research systems as well as approaches that were costly to adopt.

The transferable lesson is **time to useful value**: a specification should pay for itself at the scale where it is adopted. Trial one role boundary, one policy, one exception class, and one acceptance check before committing to a large ontology. Cognition's warning about multi-agent construction is a vendor engineering position worth comparing with empirical alternatives, not a theorem that collaboration is always inferior [Cognition 2025].

 **Caution – Synthetic bureaucracy is a measurable failure mode.** Cheap generation can produce more meetings, reports, and approvals without better outcomes. Measure accepted results, human review time, recovery costs, and avoided errors. Retire procedures that consume attention without improving those measures. More organizational vocabulary is not necessarily more organizational effectiveness.

80. AgentOrg. Product website, maturity unverified. <https://agentorg.run/>

81. Anthropic. (2024). "Introducing the Model Context Protocol." 25 November 2024. <https://www.anthropic.com/news/model-context-protocol>

82. Cognition / Yan, W. (2025). "Don't Build Multi-Agents." 12 June 2025. Vendor engineering argument. <https://cognition.com/blog/dont-build-multi-agents>

83. Google Developers. (2025). "A2A: A New Era of Agent Interoperability." <https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/>

84. Microsoft. (2026). *Microsoft Agent 365 Overview*. Documentation, checked 15 September 2026. <https://learn.microsoft.com/en-us/microsoft-agent-365/overview>

85. OpenAI. (2026). "Introducing OpenAI Frontier." 5 February 2026. <https://openai.com/index/introducing-openai-frontier/>

86. Relevance AI. Product website; advertised agent-team capabilities, not independently evaluated here. <https://relevanceai.com/>

87. ServiceNow. (2026). *AI Control Tower*. Product description, checked 15 September 2026. <https://www.servicenow.com/products/ai-control-tower.html>

88. Twinorg AI. Product website, maturity unverified. <https://twinorgai.com/>

89. Workday. (2025). *Agent System of Record* announcement, 11 February 2025. The earlier cited newsroom URL was unavailable during this revision; current product details require confirmation from Workday documentation.

Failures crossing workers, tools, policies, and state are difficult to reconstruct without linked records. Invest in event lineage, reproducible attempts, controlled replay, and observability early. The next section makes those recovery boundaries concrete through coordination services and transactions.

12.11 Coordination services, transactions, and external effects

Google's Chubby service illustrates these ideas in operation. The Google File System (GFS) and Bigtable storage system use it for functions including master election, service discovery, and small reliable metadata [Burrows 2006]. ZooKeeper provides primitives from which clients build locks, elections, membership, and other recipes [Hunt et al. 2010]. These are not complete organization engines or bulk artifact stores.

The ZooKeeper lock recipe explicitly handles successful sequential-node creation whose reply is lost: the client uses a GUID to locate its node after reconnection. Its predecessor-watch pattern avoids waking every contender for each release. A timeout is therefore not proof of no effect, and a watch is not a durable history of every state change.⁹⁰

Boundary	Mechanism to evaluate	Failure the mechanism must expose
Work admission	Atomic claim plus durable attempt identity	Worker stops after removing available work
Ownership change	Receiver-enforced generation or sequencer	Paused former owner resumes after takeover
External operation	Receiver-supported operation identity and reconciliation	Effect succeeds but acknowledgement is lost
Local state and publication	Transactional outbox where one transaction can cover both	Committed work is never published, or publication is repeated
Result and acceptance	Separate submission and verification records	Self-reported completion is mistaken for outcome
Revocation and authority	Current checks at consequential boundaries	An old prompt or claim widens current permissions

A transaction protects only participating resources. A JavaSpaces transaction or database transaction cannot roll back an unrelated service simply because a call was made while the transaction was open. An outbox makes local publication recoverable, not every remote effect exactly once. A saga's compensation is a further action with its own consequences [Garcia-Molina and Salem 1987]. Keep unknown external outcomes visible until reconciled.

90. Apache ZooKeeper, "Recipes and Solutions," especially recoverable creation errors and locks, inspected 15 September 2026. <https://zookeeper.apache.org/doc/current/recipes.html>. Exact deployment guarantees require the selected release's programmer's guide and tests; no distributed cluster was run for this study.

 Key concept**Coordinate the invariant, not every observation.**

Concurrent evidence collection can often be decoupled, while exclusive ownership, revocation, and scarce-resource admission need agreement or correctly allocated rights [Bailis et al. 2014]. Choose the smallest coordination boundary that preserves the actual requirement. Do not build a bespoke agent queue when an established transactional or workflow substrate provides the required contract.

 Concrete example 29. JavaSpaces: entry lifetime, transaction, and wait are separate fields

Specimen • Apache River 2.2.2 API signature reductions. Entry lifetime, transaction and wait appear as separate parameters.⁹¹

```
Lease write(Entry entry, Transaction txn, long lease)
Entry take(Entry tmpl, Transaction txn, long timeout)
```

Read the artifact. `write` returns a lease for an entry; `take` consumes a matching entry under an optional transaction and a waiting timeout. The entry's lease does not lease a worker's authority. A transaction over space operations does not automatically cover an unrelated remote service. Matching public fields against a template is an access mechanism, not proof of institutional permission.

`takeIfExists` is also not simply synonymous with “instant”: the API allows waiting for transactional state to settle. Inspect the contract, not just the method name.

Source and verification. These signature reductions omit declared exceptions and are not standalone compilation units. Signatures and semantics were checked against the versioned API documentation; no JavaSpaces service or transaction coordinator was run.

12.12 Further afield

Top recommended readings

Direct infrastructure foundations, ranked for reliable effects, stale authority, and the boundary of necessary coordination.

1. Garcia-Molina and Salem (1987), *Sagas*.

This database work explains how long-lived activity can be divided into smaller transactions with compensating actions. It is foundational for the chapter's distinction between retry, compensation, and a global rollback that may not exist. Read the recovery contract, not just the pattern name. Compensation is another meaningful action; it need not erase every external consequence.

91. Apache River 2.2.2, `net.jini.space.JavaSpace`, methods `write`, `take`, and `takeIfExists`, inspected 15 September 2026. <https://river.apache.org/release-doc/2.2.2/api/net/jini/space/JavaSpace.html>.

2. Burrows (2006), *The Chubby Lock Service for Loosely-Coupled Distributed Systems*.

Chubby grounds distributed coordination in an operating service and its failure assumptions. Section 2.4 is particularly important: the receiving service must validate sequencers so an obsolete holder cannot act with stale authority. Read this before treating lease expiry or a generation number in a log as fencing. The protected boundary must implement the check.

3. Bailis and colleagues (2014), *Coordination Avoidance in Database Systems*.

The invariant-confluence perspective asks when coordination is necessary to preserve an application requirement, rather than assuming every update needs agreement. Read the examples and conditions alongside the book's ownership and resource constraints. It offers a principled reason to narrow coordination, not permission to discard the transactions or rights that a nonconfluent invariant needs.

Broader explorations

Danger signals rather than a simple insider/outsider divide. Matzinger's immunological essay proposed focusing on danger and signals from tissues rather than treating self/non-self discrimination as the immune system's sole organizing principle [Matzinger 1994]. It is a theoretical challenge within immunology, not a security architecture for agent systems.

As a speculative design exercise, compare defenses based only on where a message came from with defenses that also examine the state change it requests and the evidence of harm it could produce. Internal content can be compromised; external content can be useful. The analogy suggests questions about contextual signals and layered response, but it does not justify weakening access control or treating suspicious behavior as proof of hostile intent. Read it to challenge a mental model, not to borrow biological authority for an untested safeguard.

Complex interactions can defeat component-level reassurance. Charles Perrow's *Normal Accidents* studies systems in which interactive complexity and tight coupling make failures difficult to anticipate and contain [Perrow 1984]. Read it beside an architecture composed of individually plausible services. Which interactions can occur before an operator understands what happened, and how much time or slack exists for recovery? The book's historical arguments do not establish that all complex agent systems inevitably fail. They suggest examining coupling and the consequences of unexpected combinations, rather than treating component test coverage as a complete account of system safety.

Risk management crosses organizational levels. Rasmussen's "Risk Management in a Dynamic Society" analyzes control in changing sociotechnical settings [Rasmussen 1997]. Its relevance is that pressure, adaptation, and feedback operate across managers, operators, technical systems, and wider institutions. A local guard may be correct while the broader arrangement steadily pushes work toward unsafe conditions. Ask which constraints are communicated, which feedback reaches decision makers, and how changes alter the boundary of acceptable performance. This perspective supports a system-level research design; it is not a substitute for testing the concrete tool permissions, state transitions, and failure histories of the implementation.

Safety as control, not only component reliability. Nancy Leveson's *Engineering a Safer World* develops a systems-theoretic approach to safety [Leveson 2012]. Read it to separate "the component did not break" from "the system constrained hazardous behavior." A functioning controller can still issue an unsafe action because its model of the process is wrong or feedback is missing. In an artificial organization, examine the relationship between the worker, policy mechanism, environment, and operator. The analogy suggests hazard and control analysis alongside ordinary tests. It does not establish that adding a policy engine supplies the safety properties of a properly analyzed control structure.

Infrastructure is also an organizational relationship. Star and Ruhleder's study of a collaborative scientific system connects local use with larger infrastructural change [Star and Ruhleder 1996]. It is worth revisiting during runtime selection: a feature that works in a demonstration may require support, shared conventions, training, and compatible institutional practices to work reliably elsewhere. Map these requirements alongside APIs and storage. An integration can fail because the surrounding organization cannot sustain its assumptions, even when its transport and schema are correct. This is a reason to include operational adoption in evaluation, not to dismiss technical contracts or treat every deployment problem as purely social.

13 A construction method

© **Core point — Build one governed outcome end to end.** Define acceptance and authority before dispatch, test failure and recovery, and only then expand. A collection of capable agents is not a substitute for one demonstrably correct organizational transaction.

This method deliberately begins with a real outcome and adds organization only where a simpler mechanism fails. Use the ten steps as the working sequence; Sections 13.1–13.3 provide acceptance checks and limits for a design review.

Step 1: define the institutional boundary

Name the legal operator, affected people, data, systems, jurisdictions, and external effects. Decide what the artificial organization can observe and what it can change. **Do not start with an org chart.**

Step 2: choose one bounded outcome

Specify an observable result and time horizon. “Improve marketing” is not an outcome. “Verify that a database snapshot restores the required records” names a testable property. GitLab's postmortem shows why the required records and recovery point must be explicit: restoring service did not restore every lost database change.⁹² Define the time horizon and acceptable loss separately.

Step 3: map dependencies and consequences

List inputs, decisions, actions, handoffs, irreversible effects, affected rights, and acceptance tests. Use deterministic functions and workflows for stable transformations. Mark where judgment is genuinely required.

Step 4: define roles only at accountability boundaries

Create a role when obligations must persist independently of a particular agent, when authority differs, or when incompatible duties require separation. Define cardinality and compatibility. Keep the first organization small.

92. GitLab, “Postmortem of database outage of January 31,” 10 February 2017, especially “Broken recovery procedures,” “Recovering GitLab.com,” and “Root cause analysis.” This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.

Step 5: create commitments and authority leases

For every delegation specify the mission, evidence, deadline, review triggers, and fallback. Allocate authority separately for acquisition, analysis, selection, and execution. Bind technical credentials to the narrower institutional lease. The stage distinction follows Parasuraman, Sheridan, and Wickens (2000); the lease and recordkeeping requirements are this book's engineering synthesis.

Step 6: encode policy at the right strength

Regiment catastrophic or irreversible boundaries. **Enforce** contextual rules with monitoring and reparation. Version policies and constitutive mappings. Provide a human exception path where legitimate exceptions exist.

Step 7: design evidence and acceptance before dispatch

Define what the worker must return and how an independent party can check it. Capture sources and actions mechanically. Separate submitted from accepted and accepted from observed outcome. PROV-DM supplies the lineage vocabulary [W3C 2013], not the acceptance judgment.

Step 8: design attention as a measured control system

Assign interruption channels, deadlines, and safe no-response defaults. Run triggers in shadow mode. Label alerts after the fact to estimate PPV and sample below-threshold events to estimate misses. Budget review demand before admitting concurrent work.

Step 9: evaluate representative situations

Include concurrent histories: two workers claim the same work; a publisher loses its acknowledgment; a consumer stops after claim; a stale owner resumes; a remote effect succeeds before the caller times out. Check durable state after reconnect, not only the immediate response. The Rinda source box supplies a small semantic check; it is not a substitute for these substrate-specific tests. For multi-step effects, the saga tradition makes compensation explicit rather than assuming a global undo [Garcia-Molina and Salem 1987]. Its transaction assumptions must be checked against the actual external operations.

Test ordinary success, ambiguity, tool denial, stale evidence, agent replacement, policy change, timeout, duplicate dispatch, malicious input, disagreement, rollback, and external side-effect failure. Measure outcome accuracy, policy violations, missed escalations, false alarms, review time, recovery time, and harm detected through delayed follow-up or independent sampling. Record what these checks cannot observe; absence of detected harm is not proof of absence.

Step 10: expand authority empirically

Begin in observe or recommend mode. Increase authority only for a named task class after sufficient representative evidence. Reset confidence after changing models, prompts, tools, retrieval, policy, or environment. Retire processes that do not justify their attention and maintenance cost.

Reusable decision questions

When evaluating an unfamiliar technique, ask:

1. What concrete dependency or risk does it solve?
2. Which assumptions does it make about error independence, observability, stationarity, and human response?
3. Is this an organization problem, a workflow problem, or a tool problem?
4. Can deterministic machinery solve it more clearly?
5. What authority does it create, and who can revoke it?
6. What happens when the agent, monitor, or human is wrong or silent?
7. How is success accepted independently of self-report?
8. Which evidence survives compression and handoff?
9. What new failure boundary and review burden does the technique add?
10. What measurement would cause us to remove or replace it?

Practice note

Retrofitting a running agent system.

Observe one consequential workflow without widening access. Identify its writers, external effects, and acceptance practice. Add work and attempt identifiers, then record proposed policy decisions in shadow mode. Compare them with actual outcomes before enabling enforcement.

Migrate one action class at a time. Drain or explicitly adopt in-flight work; do not let old and new controllers spend independently against the same commitment. Configuration rollback is not reversal of an external act. This migration heuristic derives from the preceding runtime requirements and needs tests on the system being migrated, not just a clean-install demonstration.

13.1 A surface-by-surface acceptance checklist

The design ideas become useful when each has an observable acceptance condition. The following checklist consolidates the institutional requirements without requiring every organization to adopt the same interface or mathematical model.

Surface	Required question	Concrete acceptance check
Decision queue	Why does this need this person's attention now?	Each item has an owner, reason, deadline, default, and relevant evidence
Brief	What supports or defeats the recommendation?	Claims, warrants, qualifiers, and unresolved objections remain inspectable
Evidence record	What happened, using which sources and authority?	Recorded lineage can reconstruct an attempt independently of narration
Organization model	Which duties and rights survive replacement?	Role compatibility, enactment intervals, and obligations pass replacement tests
Policy engine	Which rule applies to this act at this time?	Applicable version, validity, conflict resolution, denial, and remedy are recorded
Deliberation	How does discussion become an authorized commitment?	Confirmation creates an attributable decision and its required follow-through
Memory	Which record is valid and why is it relevant now?	A recent proposal cannot override an active rule; re-opening retrieves applicable rationale
Adapters	What institutional meaning does an external operation have?	Typed payloads, scoped identity, evidence capture, and reconciliation are tested
Human control	Can the operator resume understanding and action?	A time-bounded handoff or interruption test succeeds under realistic workload
Recovery	What if the worker, monitor, or remote service fails?	Uncertain side effects, retries, compensation, and escalation have tested paths



Design implication – Keep the important facts in the decision surface. An executive view should expose outcomes, exceptions, uncertainty, requested authority, and unresolved dissent. Raw traces belong in supporting inspection, not in place of the business decision. Choose a compact evidence packet rather than a ceremony of mandatory formalisms when the smaller representation is sufficient.

13.2 From a first workflow to routine operation

Use the ten steps to build the first bounded workflow, then use the acceptance table to decide whether it is ready for routine use. A successful demonstration is only the start: repeat the checks under representative workload, replacement, policy changes, and partial failure. Record who owns each control and what evidence would require narrowing or suspending the delegated authority.

Adopt only the mechanisms the task requires, but retain the distinctions they protect when changing implementations. A smaller initial system may use simple records and deterministic guards; a regulated or safety-critical system may need substantially more assurance. Passing this review does not, by itself, establish that the product is economically viable.

13.3 Which assumptions actually change with artificial workers?

Error dependence, attention limits, information compression, conflicting objectives, and small-sample learning occur in both human and artificial organizations. The transition does not replace an ideal human organization with a uniquely fallible computational one. What changes is the distribution of capabilities, costs, observability, permissions, and failure mechanisms.

Test shared-model errors; test human shared assumptions too. Measure workload instead of importing a span-of-control number. Preserve evidence despite summarization even when storage is cheap. Investigate drift, malicious input, and conflicting objective pursuit rather than assuming only one exists. Evaluate the effective sample size instead of inferring it from workforce size.

 **Caution – Reliability can increase the value of governance without proving that another layer is worthwhile.** Controls may make imperfect workers usable, but they also introduce failure points and review costs. Compare net accepted outcomes, harms, and total cost against a simpler baseline. “Governance helps when workers fail” is a hypothesis about a particular control, not a universal return-on-investment guarantee.

13.4 Further afield

Top recommended readings

Direct foundations of this chapter's construction method, ranked by the design decisions they make inspectable. The ten-step method is a synthesis, not a recipe claimed by any one of these sources.

1. **Parasuraman, Sheridan, and Wickens (2000), *A Model for Types and Levels of Human Interaction with Automation*.**

Read this first when translating a desired outcome into delegated rights. The supervisory-control framework distinguishes observing, analyzing, selecting, and executing, so construction need not begin with one global autonomy setting. Its importance is a better allocation question, not a universal allocation answer. Map each stage to an enforceable boundary and a representative evaluation.

2. **W3C (2013), *PROV-DM: The PROV Data Model*.**

A standard lineage model gives the evidence record concrete objects and relations before a worker is dispatched. Trace an artifact through its inputs, producing activity, and responsible actor, then ask what the acceptance check still needs. Historically, provenance work supplies an interoperable descriptive foundation; it does not decide whether an authorized action achieved the intended outcome.

3. Garcia-Molina and Salem (1987), Sagas.

Read this when the acceptance test includes partial execution and recovery. The long-lived-transaction problem predates agentic workflows and makes clear why completed suboperations may require compensation rather than rollback. Use the paper to state recovery obligations explicitly, but do not assume every external effect has a valid compensating operation or that compensation restores the original world.

Broader explorations

The option to wait is itself a design choice. Dixit and Pindyck's *Investment under Uncertainty* studies investment where uncertainty and irreversibility matter [Dixit and Pindyck 1994]. It offers an economic perspective on why committing immediately and retaining an option to act later are not equivalent, even when the eventual project looks attractive.

For artificial organizations, explore staged authority as an option: a small, reversible deployment may buy information before an irreversible commitment. Waiting also has costs, and a pilot that cannot resolve the relevant uncertainty may buy little. Compare the information gained, the harm still possible, and the opportunities forgone. This is an analogy to investment under specified models, not a universal rule to delay automation or a numerical valuation of governance without an explicit economic model.

Design starts with relations among requirements. Christopher Alexander's *Notes on the Synthesis of Form* examines the relationship between a form and the context it must fit [Alexander 1964]. It offers a useful architectural analogy for decomposing a system: separating responsibilities well requires understanding which requirements interact. A seemingly neat component boundary can force continual repair when the underlying dependencies cross it. Read the work to ask what evidence supports the chosen decomposition and what misfits it creates. Its formal and design arguments are not a ready-made algorithm for organizing agent teams; they suggest a way to examine why a particular boundary helps or hinders the whole.

Designed systems deserve their own science. Simon's *The Sciences of the Artificial* develops a perspective on systems made to meet purposes within environments [Simon 1996]. Read it beside the construction method to examine the relationship between the internal arrangement and the conditions in which it must operate. A design can be coherent internally while poorly adapted to its actual environment. Ask which assumptions about users, resources, institutions, and feedback make the arrangement viable. The work supports studying design seriously; it does not mean that every plausible artifact or elegant specification has already been validated by being constructed.

Improvement can crowd out exploration. March's "Exploration and Exploitation in Organizational Learning" studies the tension between refining familiar activities and investigating new possibilities [March 1991]. This is relevant to a staged rollout that expands only what currently performs well. Such discipline can improve reliability but also narrow what the organization learns. Distinguish production authority from a protected experimental budget, and observe the effects over time. The paper models particular learning situations; it does not supply a universal exploration ratio. Its practical question is which short-term feedback mechanisms might make the organization increasingly efficient at pursuing an inadequate set of possibilities.

Incremental choice is not simply failed optimization. Lindblom's "The Science of 'Muddling Through'" contrasts comprehensive rational analysis with successive limited comparisons in policy making [Lindblom 1959]. It offers an adjacent perspective on beginning with one bounded workflow. Sometimes the available knowledge and political agreement support comparison of a few feasible changes better than a supposedly complete redesign. Ask what a small change can reveal and what entrenched assumptions it leaves untouched. The connection is not an excuse to avoid long-term consequences or foundational reform; it helps make the scope and limits of incremental experimentation explicit rather than treating every pilot as a miniature proof of the whole.

14 Putting the concepts to work

© **Core point — Connect the controls to the outcome.** Roles, coordination, policy, evidence, memory, and review are useful together, at a particular boundary of work. An intelligible handoff is not necessarily authorized; a completed action is not necessarily an accepted result; an accepted result does not erase every downstream loss.

The preceding chapters separated mechanisms so that their contributions could be understood. Here, six concrete situations bring them together. Each receives the same treatment: the situation, the connections among organizational concepts, the evidence needed for closure, and the limits of what can be concluded. No case is a complete demonstration of an artificial organization, and no one case supplies the template for all the others.

The first four are documented incidents or a legal decision. The last two combine published evaluations and a local library check with explicitly proposed design responses. **Reported facts are not the same as proposed controls:** the analysis does not invent historical staffing, authority chains, or agent deployments. Earlier diagrams and specimens supply detail; this chapter follows the reasoning across those boundaries.

14.1 An engineering handoff: Mars Climate Orbiter

Insight. The Orbiter case shows that division of labor must transfer verification duties and shared meaning along with the data needed by the next team.

Situation. Spacecraft and navigation teams exchanged data, but incompatible units crossed the interface without correction. NASA/JPL's preliminary account identifies the failure of checks and systems engineering as central.⁹³ The data transfer worked in a narrow technical sense; the receiving activity did not obtain the meaning on which its calculation depended.

Putting the concepts together. Division of labor creates a dependency between the producing and receiving roles. Their obligation is not merely to send and receive a file: the handoff must preserve units, interpretation, and the evidence supporting acceptance. A shared vocabulary helps name the quantities; a schema can constrain their representation; a check must still establish that the actual values satisfy the agreed interpretation. Neither a role description nor a message transport supplies that entire contract.

93. NASA/JPL, "Mars Climate Orbiter Team Finds Likely Cause of Loss," 30 September 1999, release 99-083. This is a preliminary institutional account, not the final investigation. <https://www.jpl.nasa.gov/news/mars-climate-orbiter-team-finds-likely-cause-of-loss/>.

In an agent-assisted version of such work, a summarizer should preserve the quantity definitions and unresolved discrepancies instead of smoothing them into a confident narrative. A receiving role needs a way to reject or escalate an incompatible artifact. These are proposed uses of loss-accounted compression, role obligations, and action-boundary checks, not controls demonstrated by the historical incident.

Closure and limits. Retain the transmitted artifact, its version and unit conventions, the relevant transformation, and the result of an independent consistency check. Acceptance concerns the interpretation needed for the next operation, not just successful parsing. The case does not prove that adding a generic schema, reviewer, or LLM would have prevented the loss. Its transferable lesson is that coordination must carry **meaning and verification duties** as well as data.

14.2 Public advice: Air Canada's bereavement policy

Insight. Air Canada's policy and its chatbot's advice diverged, and the resulting customer loss remained a responsibility to remedy rather than merely a statement to correct.

Situation. A customer relied on chatbot advice that conflicted with the airline's bereavement policy. The tribunal found misleading advice and reasonable detrimental reliance; pointing to the correct policy page did not dispose of the claim.⁹⁴ The decision concerns this dispute and jurisdiction, not a universal rule for every generated statement.

Putting the concepts together. Institutional policy and its customer-facing representation are connected but distinct. A knowledge store can contain the correct rule while an interface still misstates it. The responsible organization therefore needs maintained ownership of both the rule and its operational representations. For a proposed agent system, policy versions, applicability dates, and the actual advice given belong in the durable record; replacing a worker or updating its prompt must not erase the unresolved customer commitment.

The judgment does not establish a retrieval-system failure, so the diagnosis must not assume one. The defensible design response is broader: reconcile the representations, preserve what was communicated, route uncertain cases to a role able to resolve them, and evaluate the answer against the applicable policy. An assistant's ability to speak for a service is not permission to invent its terms.

Closure and limits. Correcting future advice, verifying that correction, and remedying an existing customer's loss are different outcomes. The judgment records that the airline said it could update its chatbot; the dispute still proceeded to a compensation decision. A review should retain the advice, applicable terms, customer reliance, correction evidence, and disposition of the claim. A newly accurate answer cannot retroactively make the earlier interaction accurate.

94. *Moffatt v. Air Canada*, 2024 BCCRT 149, Civil Resolution Tribunal, 14 February 2024: paragraphs 14–23 describe the chatbot evidence and policy conflict; paragraphs 24–32 explain negligent misrepresentation; paragraphs 40–44 give the remedy. Primary decision inspected 15 September 2026: <https://decisions.civilresolutionbc.ca/crt/crtd/en/item/525448/index.do>. The case does not identify an LLM architecture.

14.3 Consequential execution: Knight Capital

Insight. Knight Capital's automation could execute orders and emit error messages while still lacking effective controls over the exposure it created.

Situation. Erroneous automated orders reached the market. The SEC found inadequate controls and described 97 automated pre-market emails identifying an error, without action on those messages that day. The messages were not designed as system alerts.⁹⁵ Effective technical execution therefore coexisted with inadequate control and response.

Putting the concepts together. Admission to an interface, authorization for a particular act, and enforcement of exposure limits are separate boundaries. A worker that can submit an order has not thereby demonstrated that the order is within its permitted scope. A proposed agent-assisted process needs guards where the consequential action occurs, rather than treating a manager's earlier instruction as a lasting guarantee about every later request.

Human oversight also needs an operational contract: an accountable recipient, an interpretable signal, a feasible response, and enough time for intervention to matter. More messages do not supply these properties. The count of 97 emails cannot establish positive predictive value or supervisory capacity: it lacks the labeled denominator and operating conditions needed for those estimates. This is not evidence of a particular approval waterfall or measured habituation.

Closure and limits. Review the effective guard, the action and its target, the exposure created, and the response actually taken. Record ambiguous or remaining effects rather than closing the incident when a notification is sent. Detection performance requires measurements of prevalence, misses, false positives, response time, and consequences. The case does not show that all automation is unsafe; it shows why **execution, control, and attention** must be evaluated together without substituting one for the others.

14.4 Recovering a service: GitLab's database outage

Insight. GitLab restored service while losing some database changes, demonstrating that restored availability and recovered data are distinct outcomes.

Situation. GitLab's 31 January 2017 recovery used a roughly six-hour-old staging snapshot rather than an almost day-old alternative. Replication was broken, and the ordinary logical-backup route had failed. Its PostgreSQL version could not dump the newer production database; error notifications were rejected by the receiving mail system.⁹⁶ A configured job and a quiet inbox were not evidence of a usable recovery artifact.

95. U.S. Securities and Exchange Commission, "SEC Charges Knight Capital With Violations of Market Access Rule," 16 October 2013, release 2013-222; links to administrative order 34-70694. Knight consented without admitting or denying the findings. <https://www.sec.gov/newsroom/press-releases/2013-222>.

96. GitLab, "Postmortem of database outage of January 31," 10 February 2017, especially "Broken recovery procedures," "Recovering GitLab.com," and "Root cause analysis." This is the operator's retrospective account. <https://about.gitlab.com/blog/postmortem-of-data-base-outage-of-january-31/>.

Putting the concepts together. The postmortem links missing backup tests to lack of ownership. It also records a destructive operation against the primary while the operator believed it targeted the secondary. Durable responsibility, target identity, and recovery capability address different weaknesses. Preventing one dangerous command would not cover disk corruption or every other cause of loss; the source emphasizes recoverability as well as environment identification.

Recovery itself had dependencies: parallel copying, separate webhook restoration, sequence adjustment, and service checks, as shown in Section 12.2. Copying took roughly 18 hours because storage throughput was the bottleneck. A decision packet should preserve recovery points, feasibility, likely data loss, and uncertainty. That is an editorial recommendation, not an invented historical approval model.

Closure and limits. Restored availability, restored auxiliary records, and recovered user data are different outcomes. Some application-database changes were unrecoverable; repositories and wikis were stored separately. A deterministic restore drill can test specified properties without requiring debating agents. Judgment remains useful for ambiguous options, but the postmortem does not establish that a manager agent would have prevented the incident. Completion must retain the loss interval and affected data classes, not compress them into the single word “recovered.”

14.5 Collective judgment: agreement and independent evidence

Insight. Sycophancy and evaluator self-preference show how favorable judgments can reflect shared biases rather than independent confirmation.

Situation. Published studies of sycophancy and evaluator self-preference identify ways in which agreement or favorable ratings can mislead [Sharma et al. 2023; Panickssery et al. 2024]. They do not establish that every same-model team fails, or that switching model families necessarily creates independence. The experimental conditions matter as much as the appealing label “multi-agent review.”

Putting the concepts together. A panel distributes contributions, but its topology does not determine how independent their evidence is. A critic needs relevant evidence access and a route for an objection to change the decision. Argumentation distinguishes a challenged premise from a challenged inference; the decision record should retain that distinction and the strongest unresolved objection. A majority vote does not settle a disputed fact or authorize a choice among competing values.

For a proposed evaluation, retain model and provider lineage, prompts or scaffolds, shared sources, whether participants saw each other's answers, and the evidence each supplied. Compare the arrangement with a strong single-worker baseline and independent outcome checks. Blind authorship where relevant and vary presentation order. These controls test particular explanations of a result; they are not a claim that a different judge is automatically neutral.

Closure and limits. An accepted recommendation should identify its decision owner, supporting evidence, qualifications, and conditions for reopening it. Organizational memory should retain the later outcome and the unresolved issues, not simply the winning answer. Agreement is useful when

it reflects warranted convergence, but counting votes or negative comments cannot establish that the institution has performed substantive scrutiny.

14.6 Claiming work: the Rinda tuple-space example

Insight. The Rinda check excludes two consumers from taking the same tuple but leaves duplicate publication and verified completion as separate problems.

Situation. The local Rinda check demonstrates matching, reading, consuming, and duplicate publication through a real library. With one tuple available, one consuming operation succeeds; a second consumer cannot take that same tuple. The source example in Section 6.5 documents the narrow test. It is not a demonstration of exactly-once completion of a business outcome.

Putting the concepts together. Associative discovery answers how a participant finds work; consumption answers which participant takes a particular record. Neither establishes what happens if that participant stops before producing a result. Duplicate publication can also admit another attempt. The organizational obligation must survive those changes in worker or attempt: “claimed” is an execution state, not acceptance of the promised outcome.

A proposed work protocol therefore records publication identity, ownership, attempt state, submitted evidence, and acceptance separately. Recovery after an interruption must distinguish an unperformed action from a performed action whose reply was lost. Receiver-supported operation identity, fencing where applicable, and reconciliation address different parts of that problem. A tuple space does not supply them merely because its consuming operation is atomic.

Closure and limits. Retain the exact task identity, attempts, outcome checks, and any unresolved external effect. A replacement worker must acquire current authority rather than inherit it from an abandoned local variable. Further failure-injection tests could stop a consumer after claim, repeat publication, or lose an acknowledgement; those are proposed tests, not histories exercised by the simple Rinda check. The example connects coordination to commitments, durable memory, and acceptance without confusing a library primitive with an entire organization.

14.7 What the examples establish together

The same review questions recur, but their answers are specific to the work. Engineering transfer concerns interpreted quantities; public advice concerns representations and remedy; execution concerns permitted effects; recovery concerns several distinct properties; deliberation concerns warrant-judgment; claiming concerns continuing responsibility. **No single completion signal can stand for all of these outcomes.**

Situation	Connected mechanisms	Evidence needed for accountable closure
Engineering handoff	Shared meaning, role obligations, acceptance	Exact artifact, units, transformation, consistency checks
Public advice	Applicable policy, representation, continuing responsibility	Advice given, governing terms, verified correction, customer remedy
Market execution	Scoped authority, enforced limits, timely oversight	Actual action, exposure, effective guard, response and remaining effects
Service recovery	Ownership, dependency-aware workflow, outcome verification	Recovery point, restored objects, integrity checks, loss and uncertainty
Collective judgment	Independent evidence, consequential criticism, decision rights	Evaluation conditions, reasons, dissent, decision and later outcome
Claimed work	Atomic claim, durable commitment, recovery and acceptance	Task identity, attempts, current authority, result and external-effect status

These are applications of the book's distinctions, not six equivalent experiments or a common numerical risk estimate. The public cases establish narrower events and consequences; the proposed controls still need evaluation in their intended setting. A simple deterministic operation is preferable when it answers the required question. Add coordination and judgment when the dependency requires them, not to imitate an organizational chart.

Key concept

Test the distinctions under pressure.

Credentials are not permission; consensus is not independence; recency is not validity; a completed attempt is not an achieved outcome. An evaluation should expose the missing boundary and its consequences, not merely recognize the terminology.

A **quiet queue** likewise needs interpretation. It may indicate healthy operation, low activity, a failed trigger, or an unavailable evidence source. Recent successful checks, stale observations, and work awaiting acceptance must remain distinguishable even when no decision is presently pending.

 **Design implication – Silence needs a positive account.** “No action needed” should mean that the required checks ran, their evidence is sufficiently fresh, and no applicable threshold or review condition requires action. Otherwise show the missing or uncertain observation. Routine sampled audits and suitable review cadence should follow the task's risk; quietness alone is not success.

14.8 Further afield

Top recommended readings

Direct case records, ranked for revisiting how several mechanisms interact in a single outcome. These are primary accounts, not a hierarchy of universal templates.

1. GitLab (2017), *Postmortem of database outage of January 31*.

Return to this account for its explicit chain of recovery artifacts, dependencies, service checks, and observed loss. It supports the chapter's detailed exercise in joining coordination with accountable closure. Read the sequence against the diagram and distinguish restored availability from recovered data. The operator's retrospective remains a case record, not comparative evidence that the proposed organizational controls would have prevented the incident.

2. NASA/JPL (1999), *Mars Climate Orbiter Team Finds Likely Cause of Loss*.

The preliminary account makes a foundational integration question concrete: which meaning and verification obligations must cross a handoff? Read it against the chapter's combination of roles, contracts, evidence, and review. The historical lesson is not that every failure needs central supervision. Different checks protect different properties, and proposed alternatives require their own evidence.

3. British Columbia Civil Resolution Tribunal (2024), *Moffatt v. Air Canada*.

This public decision brings organizational responsibility into contact with automated advice. Read the reasons concerning the chatbot's representation and the separate policy page; they constrain what a successful response or citation can establish. Its importance here is the accountability boundary, not a general legal rule for every chatbot, jurisdiction, or future dispute.

Broader explorations

Return to normal, or preserve the ability to function? Holling's ecological analysis distinguishes stability around an equilibrium from resilience in the face of disturbance [Holling 1973]. Those questions can diverge: small visible variation is not the same as the persistence of essential relationships under a large disruption.

Across the examples, restored service, corrected advice, controlled exposure, and capacity to respond to another incident are different properties. A study could compare recovery plans optimized for rapid restoration with plans that preserve more future recovery options. Ecological resilience is not a ready-made software metric, and undesirable arrangements can persist too. The useful connection is the change in question, not the assumption that persistence always means success.

Investigate normalization before the incident. Vaughan's *The Challenger Launch Decision* examines how organizational practices and interpretations developed before a catastrophic outcome [Vaughan 1996]. Its value for reading a recovery postmortem is to shift attention from the final visible mistake to the conditions that made it possible and intelligible at the time. What warnings had become routine, what exceptions had been accepted, and how was uncertainty communicated? An agent-operation study could inspect repeated responses to partial failures before a major loss. The comparison remains analytical: it does not establish a shared causal history between the shuttle programme and a database outage.

Recovery depends on defenses as well as operators. Reason's *Human Error* provides a framework for examining how different forms of error interact with the surrounding system [Reason 1990]. Read it to avoid a postmortem that ends with “the operator should have been more careful.” That statement does not explain target identification, notification delivery, test ownership, or the availability of a usable recovery path. Compare interventions at those different points and ask what each leaves unprotected. The aim is to understand how failures become consequential, not to remove personal responsibility or to assume that all errors can be prevented by adding another check.

Failures can incubate in institutional arrangements. Barry Turner's “The Organizational and Interorganizational Development of Disasters” considers how accidents develop through information and organizational processes over time [Turner 1976]. It is a useful companion to the durable-record requirement: what was noticed, how was it categorized, and which observations failed to reach the people who could act? A review should include the history before the recovery timeline, not only the sequence after the alarm. Turner's analysis suggests questions about accumulated discrepancies; it does not turn an archive into a guarantee that an institution will recognize and respond to those discrepancies.

Resilience has several meanings. Woods' “Four Concepts for Resilience” distinguishes uses of the term that are often compressed into a single positive label [Woods 2015]. Returning to a previous state, absorbing disturbance, extending adaptive capacity, and sustaining that capacity pose different questions. Across these examples, distinguish continuity of service, repair of affected records or representations, revised procedures, and preparation for a further disruption. A resilience claim should name the capacity and disturbance involved. The article helps refine what to measure; it does not provide a single number that can summarize every aspect of an organization's ability to recover.

From resilient operation to warranted conclusions. The following works complement the preceding resilience literature with ways to make disagreement, testing, and revision consequential. Together these research routes address both what an organization can withstand and what it can responsibly conclude.

Disagreement can design the experiment. Mellers, Hertwig, and Kahneman's adversarial collaboration examined a dispute about conjunction judgments by agreeing on empirical tests and using an arbiter [Mellers et al. 2001]. The participants did not have to begin with one interpretation, and the findings did not erase all interpretive disagreement.

This is a useful alternative to asking an agent critic merely to object to a proposal. Ask proponents and critics to specify a test that each would regard as informative, including the observations that would change their positions. Preserve their predictions before running it. The analogy is to a research practice, not an assumption that LLM debate supplies authentic independence. Its value is a stronger link between disagreement and evidence, rather than a more theatrical argument.

Design tests that can contradict the explanation. Popper's *The Logic of Scientific Discovery* places severe testing and falsifiability at the center of a philosophy of empirical science [Popper 1959]. Read it as a challenge to demonstrations that can be interpreted as successful whatever happens. Before running a test, state which observation would count against the proposed mechanism. A failed prediction may expose an auxiliary assumption or a measurement problem, so interpretation still requires care. The useful transfer is disciplined vulnerability to evidence, not the simplistic idea that one failed run mechanically settles an entire research programme.

Evaluate a programme of explanations over time. Lakatos' *The Methodology of Scientific Research Programmes* examines how related theories develop, including the difference between productive change and adjustments that merely accommodate known difficulties [Lakatos 1978]. It offers a broader frame for iterating on an agent architecture. Does the revision predict or explain something new, or only rename the failure that just occurred? Preserve the old prediction, the change, and the next test. The analogy is methodological, not a scoring rule for product releases; practical systems may need repair for immediate reasons even when that repair establishes little about a general theory.

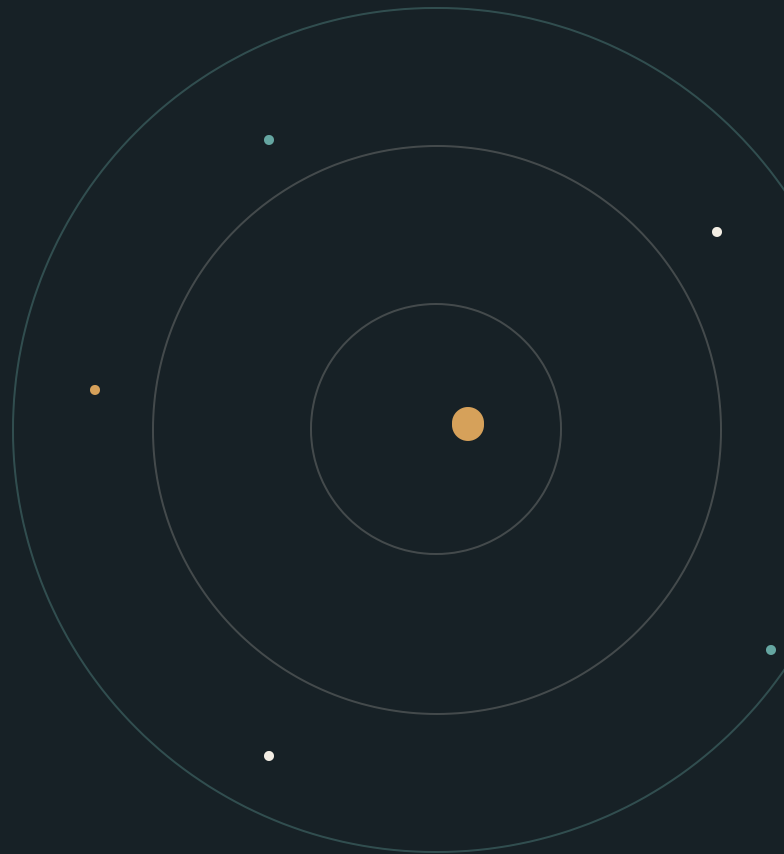
Criticism requires a receptive institution. Longino's *Science as Social Knowledge* asks how critical interaction contributes to inquiry [Longino 1990]. It complements adversarial collaboration by emphasizing the arrangements that make objections consequential. Who can question the test design, propose alternative interpretations, and obtain a response? An agent critic given permission to speak but no route to change the decision may contribute little scrutiny. A useful evaluation would trace how objections affect experiments and revisions, not merely count negative comments. This is a philosophical framework for examining inquiry, not evidence that additional critics automatically improve the accuracy of a system.

Replication is not repetition of the same bias. Ioannidis' analysis of published findings highlights the importance of study conditions, selection, and bias in interpreting apparent success [Ioannidis 2005]. Repeating one benchmark with shared prompts, data, and evaluators may produce stable numbers without testing generality. Ask which parts of the evidence are independent and what remained fixed. A stronger check varies meaningful conditions while preserving a clear question and comparator. The paper's model is not a verdict on every agent result; it supplies reasons to report the research process and negative outcomes that a success-only demonstration would conceal.

PART V

Frontier and synthesis

What contemporary systems should inherit, which questions remain open, and where the evidence leads.



15 What LLM-agent systems should inherit

© Core point — Better workers do not replace organizational mechanisms.

Decades of MAS research made role enactment, commitments, coordination, and institutional rules explicit. LLM systems make new kinds of work practical, but many familiar organizational labels still need those mechanisms beneath them. Recover the useful contract, not necessarily the old implementation.

15.1 The disconnect: vocabulary travels faster than mechanisms

An LLM team can contain a manager, a reviewer, a memory store, and a task board while leaving important questions unanswered: who owes what after a worker is replaced, which actions count as approvals, and what happens when an accepted obligation becomes impossible? These are not newly discovered consequences of language-model unreliability. Organization-oriented MAS, social commitments, joint intentions, and electronic institutions made them research subjects long before LLMs [Ferber and Gutknecht 1998; Singh 1998; Cohen and Levesque 1991; Esteva et al. 2004].

The concern has contemporary advocates. *Large Language Models Miss the Multi-Agent Mark* argues that parts of LLM-agent research adopt MAS vocabulary without adequately engaging its foundations, identifying problems in social agency, environment design, coordination, and evaluation of emergence [La Malfa et al. 2025]. That is a position paper, not a census of researchers' knowledge. This chapter examines the **engineering consequence of incomplete transfer**: an application may reconstruct a weaker version of an old mechanism, or overlook a useful alternative, because the earlier design question has disappeared from its comparison set.

Three claims must remain distinct. **Missing citation** is a bibliographic observation. **Missing mechanism** is a claim about represented state and enforced behavior. **Costly omission** requires evidence that the mechanism would improve the relevant system. None automatically proves the next. Calling the entire LLM community ignorant is neither necessary nor justified by the examples below; identifying a missing obligation lifecycle is both more precise and more useful.

15.2 How to read the comparisons

This is a structured, selective comparison, not an exhaustive systematic review or a performance leaderboard. The classical side uses the primary works and versioned specimens already developed in Chapters 5–12. The LLM side samples published architectures, implementation excerpts, and protoc-

Social
commitments

ol contracts because each exposes a different layer. **A toolkit feature is not the whole application**, and an example's silence does not establish that its host framework cannot implement a richer contract.

The sample spans conversation frameworks, simulated agents, task orchestration, failure analysis, and institutional design from 2023–2026. Comparisons use the paper revisions listed in the source note and this book's pinned implementation specimens, including A2A 1.0.0 and MCP's June 2025 specification. Their dates identify what was inspected; they do not imply that an older result remains today's leading performance.⁹⁷

Concern	Earlier mechanism	Contemporary comparison	What must be supplied when the concern matters
Durable responsibility	AGR role enactment; MOISE roles, missions, and duties	MetaGPT role classes and team assembly	Occupancy intervals, duties independent of workers, and replacement rules
Accepted obligations	Directed social commitments and lifecycle	SDK handoff; A2A task state	Debtor, creditor, content, acceptance, discharge, and repair
Meaning and permission	Counts-as, norms, and electronic institutions	Tool schemas, approval interrupts, guardrails	Recognition rules, authority checks, deadlines, and violation handling
Selecting work	Contract Net allocation; black-board control	Speaker scheduling, orchestration, subagents	Allocation policy, shared task state, admission, and ownership
Maintaining teamwork	Joint intentions, STEAM conventions and communication decisions	Progress monitoring and replanning	Who must learn that a goal failed, became impossible, or changed
Learning from experience	Organizational and transactive memory; case-based reasoning	Retrieved episodes, reflections, summaries	Applicability, provenance, revision, and evaluation of reuse
Deliberation	Argument structures and explicit acceptance semantics	Debate and model reviewers	Preserved objections, evidence, decision rule, and accountable closure
Institutional performance	Evaluate the organizational arrangement and its assumptions	Task completion and failure taxonomies	Tests that vary governance while holding worker capability and resources comparable

97. Inspection scope, 16 September 2026: AutoGen <https://arxiv.org/html/2308.08155v2>, MetaGPT <https://arxiv.org/html/2308.00352v7>, Generative Agents <https://arxiv.org/html/2304.03442v2>, Magentic-One <https://arxiv.org/html/2411.04468v1>, MAST <https://arxiv.org/html/2503.13657v3>, Waites <https://arxiv.org/html/2602.13275v1>, and La Malfa et al. <https://arxiv.org/html/2505.21298v4>. Architectural descriptions and relevant related-work passages were inspected; this chapter does not report new executions of these systems. Source-box commits and documentation dates identify the separate implementation specimens.

The last column is a design obligation for systems that need the property, not a demand that every toolkit provide every feature. A single bounded analysis may need none of the heavier machinery.

15.3 Roles: a job title versus an institutional position

MetaGPT

The [MOISE specimen](#) declares a `bricklayer` role, a `build_walls` mission, and an obligation linking them. The [MetaGPT specimen](#) constructs a team with roles such as `ProductManager()` and `Architect()` and then runs it. Both organize work, but the inspected statements answer different questions: **what duty belongs to this role?** versus **which implemented participants should join this run?**

MetaGPT's paper emphasizes standard operating procedures, structured intermediate outputs, and verification in a software-development workflow [Hong et al. 2024]. These organize production, but do not, by themselves, establish durable role occupancy, compatible positions, or the migration of unfinished duties when staffing changes.

 Practice note

Replace the architect, not the responsibility.

The MetaGPT source box makes the Architect participant concrete. The MOISE box makes the separation of role, mission, and obligation concrete. Combining those ideas suggests a useful test: interrupt a worker, replace its instance, and inspect whether the same outstanding duty still has an authorized occupant and its prior evidence. Merely restoring conversation history answers a different question. This is a proposed transfer test, not a reported MetaGPT failure.

The useful inheritance is the separation of **position, occupant, duty, and execution attempt**. It can be implemented with ordinary typed records and guards; adopting the entire MOISE toolchain is a separate decision. Where the work ends with one disposable run, such persistence may add cost without benefit.

15.4 Handoffs: passing control versus accepting a commitment

Social-commitment research represents an undertaking directed from a debtor to a creditor, with content and, where relevant, conditions and lifecycle [Singh 1998]. The [OpenAI Agents SDK handoff](#) exposes control transfer to another agent. That is useful execution behavior, but passing control alone does not say that the recipient accepted a duty to a named party.

A2A

The distinction is visible in the [A2A response](#): it carries a task identifier, `TASK_STATE_COMPLETED`, and an artifact. The schema supports interoperable task exchange. Those fields do not establish that a creditor accepted the delivered result under an organization's review policy. A2A also separates task mechanisms from implementation-specific authorization [A2A 2026]. The adopting organization must therefore connect protocol state to its own acceptance and authority rules.

The missing application-level contract matters when a request is declined, delegated onward, cancelled, delivered late, or disputed. Retain the accepted undertaking separately from the provider task and its artifact. Test whether a worker's "completed" response can prematurely close an obligation. A social commitment is a computational institutional object; its representation alone does not make it a legally enforceable contract.

15.5 Rules: stopping execution versus recognizing an official act

InstAL

The [InstAL rules](#) distinguish an observed `wave` from the institutional `greetRoom` event and represent an obligation with performance, deadline, and violation. The example is intentionally small, but its distinction is substantial: **what happened** and **what it counts as** are different objects. Electronic-institution research also studied permitted interaction sequences and infrastructure-mediated enforcement [Jones and Sergot 1996; Esteva et al. 2004; Padget et al. 2016].

LangGraph

By comparison, [LangGraph's `interrupt`](#) pauses execution and makes a value available for a later resumption. A resume value can carry an approval, but the pause does not determine which identity may approve, whether its authority has expired, or whether the approval concerns the same artifact now being executed. A guardrail can reject selected content without representing the duty to repair an earlier action.

The recoverable idea is to join execution controls to **versioned institutional recognition**. Resolve actor, scope, artifact revision, and applicable rule at the transition; distinguish rejected actions from effective-but-forbidden acts and unresolved obligations. Replay an old approval after a policy change as a test. The newer toolkit may host these checks well: the question is whether the application defines and enforces them, not whether it uses old terminology.

15.6 Coordination: choosing a speaker is not choosing an owner

Contract Net
STEAM teamwork

Contract Net separates task announcement, proposals, selection, and award [Smith 1980]. Blackboard systems expose partial solutions and a control policy for choosing contributions [Erman et al. 1980]. Joint intentions and STEAM add questions about maintaining team commitments when members learn that a goal has succeeded, failed, or become impossible [Cohen and Levesque 1991; Tambe 1997]. These mechanisms solve different problems even within classical MAS.

AutoGen

The [AutoGen round-robin excerpt](#) advances an index modulo the participant count. It selects the next speaker, deterministically. There is no bidding or award in that method. This is not evidence that AutoGen lacks other schedulers: its paper explicitly supports programmable interaction patterns. It means that a team constructed with round-robin scheduling has not acquired capability-based allocation merely by being called multi-agent [Wu et al. 2023].

Magentic-One provides a stronger contemporary comparison. Its Orchestrator plans, tracks progress, directs specialized agents, and replans after errors [Fourney et al. 2024]. That is a real advance over a fixed conversational cycle. It still leaves a separate question for an adopting application: what obligations and authorizations persist when the plan changes? Replanning and institutional continuity can complement one another rather than compete.

Illustrative failure pair. A planner may correctly select a replacement specialist while the original worker still holds active credentials and an unfinished execution attempt. The new plan is sensible, but two actors may now try to produce the same external effect. Conversely, an authority guard may prevent duplicate action while leaving the replacement unable to discover what the old attempt already did. The first problem concerns the right to act; the second concerns the information required to resume. This is a proposed diagnostic comparison, not a reported failure of Magentic-One or AutoGen. Evaluate the plan, the handover of authority, and the recovery record separately. A persuasive explanation of the new plan does not settle either operational question, and a correct access check does not by itself supply the missing history.

 Practice note

The unavailable specialist.

A coordination test can make an assigned specialist unavailable after accepting work. Speaker scheduling asks who can respond next; allocation asks who should take the work; teamwork asks which dependent participants must learn the original plan is no longer viable. Resuming the conversation is only one of those tasks. This test is an editorial application of the cited mechanisms, not an invented historical case.

Do not force auctions onto tasks with one obvious executor. Use allocation protocols when capability, cost, availability, or independently controlled participants make selection a real problem. Likewise, a shared document does not inherit Linda's atomic consumption or a replicated service's failure semantics; the concurrent-systems foundations in Chapters 6 and 12 remain relevant alongside MAS, not subsumed by it.

15.7 Memory and review: plausible behavior versus accountable reuse

Generative Agents builds observation, memory retrieval, reflection, and planning into an interactive simulation. Its paper explicitly connects to earlier believable-agent and cognitive-architecture work; it is not evidence that all LLM research forgot agent history [Park et al. 2023]. Its stated evaluation concerns believable behavior in a sandbox, not the validity of a company's policy or the discharge of a customer's claim.

The [memory-constructor specimen](#) and [Reflexion loop](#) make retention and feedback concrete. They do not turn a retrieved episode into an authoritative precedent. Organizational-memory and case-based reasoning traditions ask about retained experience, applicability, adaptation, and reuse; formal organization models supply a further distinction between a remembered rule and one currently in force. For an institutional application, test an old but active policy against a newer superseded discussion, and preserve the evidence behind a reflection rather than treating the reflection as its own proof.

Deliberation has the same boundary. Dung's argumentation framework makes attacks and acceptance semantics explicit; a model debate generates and assesses prose [Dung 1995; Khan et al. 2024]. The useful transfer is not to pretend that every debate is an argumentation solver. It is to retain objections and their grounds, make the decision rule explicit, and test whether additional agents contribute different evidence. The IETF example in Section 10.2 illustrates why an unanswered objection survives a favorable headcount.

15.8 What is genuinely new, and what is being recovered?

LLMs make broad natural-language interpretation, document synthesis, tool use, and flexible role behavior more accessible than many hand-authored earlier systems. Modern runtimes add practical persistence, browser interaction, observability, and integration. Those gains change which applications are feasible and how much authoring they require. Classical systems had working agents too; their domains, knowledge engineering, and assumptions differed.

Historical engagement is uneven rather than absent. MetaGPT cites Wooldridge and Jennings' 1998 discussion of agent-development pitfalls; Generative Agents engages cognitive architectures and earlier simulated agents. La Malfa et al. explicitly calls for reconnecting contemporary work with MAS. These counterexamples prevent the legitimate critique from becoming an indiscriminate accusation.

Waites's 2026 *Artificial Organisations* offers a concrete institutional direction: a Composer, a source-access Corroborator, and a source-restricted Critic, with differentiated access and iterative review. The value is in the implemented distinction between instructions and available operations, not the novelty of discovering separation of duties. The paper's observational, human-supervised composition setting does not isolate the causal benefit of that architecture or establish general organizational reliability [Waites 2026].

 **Caution – A recovered idea still needs an evaluation.** A verifier's rejection rate does not establish its accuracy; a higher internal review score does not independently establish better work. Different information access does not prove statistically independent errors. Verify both the architectural boundary and the quality of the judgments made within it.

Mature commercial systems may already provide identity, approvals, audit, and workflow state. Recovering an old research distinction should first prompt a search for the existing control that owns it. It need not produce another framework. Conversely, a convenient contemporary API should not determine which organizational questions are allowed to be asked.

15.9 Turn the critique into a testable research programme

A plausible explanation for incomplete transfer is that the research traditions optimize different things: capability on new tasks, explicit social semantics, simulation believability, or interoperability. Different vocabularies and the cost of operating historical toolchains can reinforce the separation. These are explanations to investigate, not measured causes of researchers' choices.

The practical remedy is a **mechanism-level comparison**. Hold the model, tools, task set, and resource limits comparable, then add one institutional mechanism at a time. Compare against both a competent single-worker baseline and a well-engineered modern workflow; a deliberately weak baseline would only make the historical method look useful.

Intervention	Situation that exposes its value	Measure alongside task quality
Durable role-to-duty records	Replace a worker with unfinished work	Orphaned or duplicated duties; recovery time
Commitment lifecycle	Decline, delegate, cancel, or dispute a result	False closure; unresolved creditor claims
Versioned authority and recognition	Replay approval after policy or artifact change	Unauthorized effects; incorrect recognition
Explicit allocation and team notifications	Remove a specialist or invalidate a dependency	Stranded work; stale plans; communication cost
Applicability-aware memory	Mix superseded rules with relevant episodes	Invalid retrieval; lost evidence; review effort
Independent acceptance evidence	Supply plausible but incorrect completion claims	False acceptance, false rejection, and human workload

Run nominal situations as well as failures: an institutional mechanism that prevents rare mistakes but makes ordinary work unusable may still be the wrong choice. Record its authoring burden, maintenance cost, latency, and exceptions. Publish negative results and the exact versions used. A benefit in one domain justifies a bounded claim, not the universal superiority of formal organization.

The strongest conclusion is therefore constructive. **The pre-LLM literature contains reusable questions, models, protocols, and failure analyses that contemporary systems underuse when they substitute organizational vocabulary for explicit contracts.** The research opportunity is to combine those assets with today's capable workers and test the resulting system, rather than choose between nostalgia and novelty.

15.10 Further afield

Top recommended readings

Direct comparison literature, ranked to read a contemporary challenge against an earlier mechanism and a substantive modern architecture.

1. **La Malfa and colleagues (2025), *Large Language Models Miss the Multi-Agent Mark*, version 4.**

This contemporary critique directly frames the chapter's question: what is lost when familiar MAS vocabulary travels without its underlying commitments and interaction mechanisms? Read the specific arguments and research directions, not the title as a verdict on every LLM system. It is a position requiring comparison with actual implementations and evaluations, not independent proof of universal failure.

2. **Smith (1980), *The Contract Net Protocol*.**

This early distributed-AI mechanism is a concrete test of historical inheritance. Announcement, proposals, evaluation, and award answer allocation questions that conversational turn-taking does not. Read the actual protocol before comparing a modern manager-worker arrangement with it. Recover the useful contract when selection is a real problem; importing auctions into an obvious assignment adds no automatic benefit.

3. **Fourney and colleagues (2024), *Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks*.**

This orchestration architecture provides a stronger contemporary comparison than a fixed speaker cycle: planning, progress tracking, direction, and replanning are explicit mechanisms. Read those mechanisms and evaluation conditions before asking what institutional continuity the adopting application must add. The architecture's contribution is real, but its evaluation does not certify an open-ended company.

Broader explorations

A citation can acknowledge, use, contrast, or criticize. Teufel, Siddharthan, and Tidhar studied automatic classification of citation function: the role a reference plays in the citing author's argument [Teufel et al. 2006]. This opens a connection from the historical-inheritance debate to bibliometrics, scientific rhetoric, and the sociology of knowledge.

A stronger study of the MAS–LLM disconnect would classify both how prior work is cited and whether its mechanisms are actually implemented or evaluated. Compare papers that name an earlier theory, adopt its distinctions without naming it, and cite it only to reject an assumption. Manual inspection remains important: neither citation counts nor an automatic classifier can establish what researchers knew. This route could test the chapter's historical hypothesis instead of allowing a persuasive narrative about rediscovery to become its own evidence.

Research traditions define exemplary problems. Kuhn's *The Structure of Scientific Revolutions* examines paradigms, normal science, and changes in scientific practice [Kuhn 1996]. Read it to investigate why two communities can use similar words while valuing different tasks, evidence, and solutions. Which examples train researchers to recognize a worthwhile problem? A study of classical MAS and LLM-agent work could compare their exemplary evaluations, not just their reference lists. Kuhn's historical account is not proof that the current situation is a scientific revolution. It offers questions about disciplinary learning and comparison without requiring that every change be cast as a dramatic replacement.

Recognition is distributed unevenly. Merton's "The Matthew Effect in Science" examines how established reputation can affect the visibility and credit given to contributions [Merton 1968]. It is a useful counterweight to a history told only through famous names or highly cited systems. Look for earlier, less visible implementations and collaborators, and distinguish priority from influence. In contemporary research, one could compare how equivalent mechanisms are described when associated with different institutions or product brands. The analysis does not imply that prominence is undeserved; it warns that reputation and citation counts are themselves social processes, not direct measures of a contribution's technical substance.

Follow the network of papers. Derek de Solla Price's "Networks of Scientific Papers" treats scientific literature as a connected structure rather than a flat reading list [Price 1965]. This is useful for studying intellectual inheritance: which clusters cite one another, which works connect them, and which links disappear when vocabulary changes? A research map can combine citation tracing with inspection of mechanisms and examples. Missing edges remain ambiguous: they can reflect different questions, incomplete search, or unattributed reuse. Network structure helps locate cases to examine; it cannot establish what an author knew or whether an omission harmed the resulting system.

Technical communities also make authority claims. Gieryn's work on boundary-work examines the practical distinction of science from non-science [Gieryn 1983]. It can be extended cautiously to analyze how research communities define serious methods, legitimate evaluation, or "real" agents. Compare those claims with the behavior and evidence actually supplied. A historical mechanism should not be dismissed because it uses old vocabulary, but neither should its scholarly pedigree exempt it from current testing. The productive question is how boundaries shape the comparison set. This remains a sociological research direction, not a license to infer motives from terminology alone.

16 What remains open

© **Core point — Design around uncertainty, not promises.** The reviewed literature does not establish reliable open-ended company autonomy, a universal supervision capacity, or one best memory architecture. Instrument the limits locally and state what evidence would change your conclusion.

The following are research questions, not missing product checkboxes:

1. **[Open] Human fan-out.** This study found no generally validated curve relating one person's decision quality to the number of concurrent reasoning agents, escalation rate, review debt, and deadline clustering.
2. **[Open] Error correlation.** We lack broad, controlled measurements of task- error correlation across model families, tools, retrievers, and scaffolds.
3. **[Open] Conversational institutional authoring.** Natural language lowers authoring cost, but it is not established that conversationally derived roles and policies are complete, consistent, or usable enough to govern.
4. **[Open] Organizational memory and forgetting.** Retrieval systems do not yet provide a generally accepted method for validity-aware institutional memory, conflict, consolidation, and principled forgetting.
5. **[Open] Rationale reuse.** Agent generation lowers capture cost, but may not solve the historical failure to retrieve and use decision rationale later.
6. **[Open] Reorganization semantics.** Changing roles and authority while work is live can orphan commitments or widen access. Safe migration methods need more evidence.
7. **[Disputed] Scalable oversight.** Debate, critics, weak-to-strong methods, and prover-verifier training help in selected settings, but none establishes reliable oversight of an open-ended artificial company.
8. **[Open] Legal treatment.** Responsibility, agency, employment, liability, recordkeeping, and sectoral duties remain jurisdiction- and use-specific.

A responsible implementation instruments these unknowns locally instead of hiding them behind a maturity label.

Three strands from the earlier chapters make these gaps concrete. Crandall and colleagues' multitasking study supplies workload concepts, not an LLM-company supervision curve [Crandall et al. 2005]. QOC makes retained decision structure inspectable without proving that people will reuse it [MacLean et al. 1991]. MAST supplies a taxonomy of failures in evaluated multi-agent LLM systems, not a demonstrated repair for every class [Cemri et al. 2025]. The agenda below asks what additional evidence would turn such contributions into dependable practice.

16.1 A measurable research agenda

Open question	A useful study design	What would count against the proposed approach?
Does conversational authoring produce governable structure?	Compare generated interpretations with expert-reviewed cases, ambiguity tests, revisions, and role changes	Silent scope widening, inconsistent norms, or users unable to detect mistakes
How independent are reviewers?	Common held-out tasks, controlled budgets, blind identity, multiple families and deterministic checks	Strong residual joint error despite nominal diversity
How much work can one person oversee?	Measure neglect tolerance, handling time, deadline bursts, task phases, and recovery performance	Missed consequential exceptions or growing review debt at the admitted load
Are escalation triggers useful in practice?	Label raised alerts, sample suppressed events, and track response and consequence	Low PPV, unobserved misses, or alerts that arrive too late to change outcomes
Can drift and goal conflict be distinguished?	Perturb context, tools, policy, feedback, and incentives while preserving comparable tasks	Detectors react to benign variation but miss consequential departures
Is rationale actually reused?	Observe reopening events, retrieval relevance, and decision changes against a baseline	Large archives with negligible use or misleading precedent retrieval
Can organizational facts be reconstructed from traces?	Independent reconstruction after replacement, partial failure, concurrent attempts, and policy change	Missing actors, unverifiable authority, ambiguous external effects, or orphaned commitments

These questions connect the theory to deployment without pretending that one product, benchmark, or interface study settles them. Instrument them from the first bounded use case. An unknown can be an acceptable research risk, a reason to narrow authority, or a deployment blocker depending on the consequences.

16.2 Further afield

Top recommended readings

Direct research foundations, ranked for making the chapter's unknowns measurable. The recent work is selected for its relevance, not an established long-term influence claim.

1. **Crandall, Goodrich, Olsen, and Nielsen (2005)**, *Validating Human-Robot Interaction Schemes in Multitasking Environments*.

Start here to turn a vague question about human fan-out into quantities such as neglect tolerance and interaction demand. Its human-robot setting belongs to an earlier supervisory-control tradition with concrete tasks and conditions. Read the measurement design before transferring any capacity estimate; concurrent reasoning agents, deadline bursts, and different consequences require new tests.

2. **MacLean, Young, Bellotti, and Moran (1991)**, *Questions, Options, and Criteria: Elements of Design Space Analysis*.

QOC makes the rationale-reuse problem specific: a stored choice should preserve alternatives and criteria that can matter when conditions change. Read the representation with later retrieval and use in mind, rather than equating capture with learning. Its historical role is to supply an inspectable design space; whether an agent-generated record improves a later decision remains empirical.

3. **Cemri and colleagues (2025)**, *Why Do Multi-Agent LLM Systems Fail?*, **version 3**.

The MAST taxonomy and associated data turn broad dissatisfaction into distinguishable failure classes. Use that specificity to formulate interventions and adverse tests. As recent evidence from evaluated systems, it cannot establish timeless failure rates or show that adding a governance mechanism will help. Test the proposed correction against capable baselines and report its new costs and failure modes.

Broader explorations

Uncertainty is not always a well-estimated lottery. Knight's *Risk, Uncertainty, and Profit*, especially Part III, Chapter VII, distinguishes measurable risk from judgment about situations that resist reliable statistical classification [Knight 1921]. His account is a useful challenge when a dashboard turns every unknown into a precise probability.

Explore what an organization should do when the event classes themselves are unclear: preserve options, seek disconfirming observations, or impose limits that do not depend on a confident estimate. Those choices require their own tradeoffs; calling a problem "Knightian" does not settle them. The connection invites a richer account of uncertainty without abandoning measurement where well-defined events and relevant observations make it useful.

Risk judgments have social organization. Douglas and Wildavsky's *Risk and Culture* examines how social arrangements and values influence which risks receive attention [Douglas and Wildavsky 1982]. This is relevant when a system treats its risk priorities as if they followed automatically from technical probabilities. Which harms are visible, whose losses count, and who can challenge the ranking? A comparative study could hold a risk inventory fixed while examining different institutional responses. The connection does not make hazards merely subjective or excuse inaccurate measurement. It adds the question of how organizations select and interpret the risks they choose to govern.

High stakes and uncertain knowledge change the inquiry. Funtowicz and Ravetz's "Science for the Post-Normal Age" addresses situations in which uncertainty, disputed values, high stakes, and urgent decisions complicate ordinary expert assessment [Funtowicz and Ravetz 1993]. Read it when a deployment decision cannot wait for a complete evidence base. Ask how quality is assessed, which perspectives are included, and what uncertainties remain consequential. The framework does not license weak evidence or remove the need for expertise. It suggests examining the process by which a decision is made when technical findings alone cannot settle the legitimate tradeoffs.

Do not erase uncertainty by overcompressing it. Andy Stirling's "Keep It Complex" argues against reducing difficult policy questions to unjustifiably simple representations [Stirling 2010]. It is a useful companion to the book's qualified summaries and decision surfaces. A single confidence or risk score may conceal disagreement about outcomes, values, and the models being used. Ask which dimensions need to remain visible for a responsible decision, and which can genuinely be summarized. This is not an argument for unreadable dashboards or endless deliberation. The challenge is to make complexity tractable without pretending that a consequential disagreement has been resolved by formatting.

Public understanding is also a relationship of trust. Brian Wynne's "Misunderstood Misunderstanding" studies Cumbrian farmers' responses to scientific advice after the Chernobyl fallout [Wynne 1992]. It shows why reception of expert knowledge cannot simply be reduced to whether the public has received enough facts. Social relationships, local knowledge, and institutional credibility matter. For an artificial organization, investigate how explanations and uncertainty disclosures are received by the people affected, including those whose experience conflicts with the official categories. The case is not evidence that local knowledge always defeats expertise; it challenges a one-way model of communication.

17 Synthesis

© **Core point — Coherent action needs structure and correction.** Keep legitimate action inspectable, preserve evidence that summaries omit, invite correction at consequential decisions, and expand autonomy only where representative local results justify it.

Retain these ideas:

- 1. An organization is not a crowd of agents.** It is a durable system of purpose, roles, obligations, authority, coordination, evidence, and learning. Simon's decision-centered account explains why arrangements matter for limited participants; it is an explanatory foundation, not a modern runtime design [Simon 1947].
- 2. Classical theory and modern infrastructure are complementary.** Classical MAS and organization science provide institutional concepts; concurrent systems explain synchronization, shared state, and recovery. Current agents, workflow engines, MCP, and A2A provide complementary execution and exchange. Compare their actual mechanisms, as in Chapter 15, rather than equating a familiar name with an inherited contract.
- 3. Roles outlive agents.** Bind obligations to roles and missions, then record which agent enacts a role during which interval. MOISE+'s linked structural, functional, and deontic descriptions make this separation explicit [Hübner et al. 2002b].
- 4. Use the simplest suitable arrangement.** Keep work human-led where agent capability or risk is unsuitable. Use deterministic workflows for known procedures and agents where demonstrated judgment or flexibility helps. Multiple workers need a benefit that repays coordination cost.
- 5. Authority is not capability and not a global dial.** Allocate acquisition, analysis, selection, and execution separately through narrow, revocable leases. The automation-stage framework motivates separate allocations; revocable institutional leases are an additional design requirement [Parasuraman et al. 2000].
- 6. Human attention is a scarce control resource.** Measure alert precision and missed events, distinguish mandatory approvals from anomaly detection, choose interruption channels, and budget review demand.
- 7. Govern actions at institutional boundaries.** Combine regimentation, enforcement, constitutive rules, policy lifecycle, and repair according to consequence and context.
- 8. Rationale is not provenance; completion is not success.** Preserve sources and actions, distinguish submission from acceptance, and observe outcomes, including adverse or inconclusive ones. Use independent review where the consequences warrant it; a second agent is not automatically independent.

- 9. Dissent must be real enough to correct error.** Diversity of labels is not diversity of evidence or failure mode.
- 10. Expand autonomy only from local evidence.** Design for improving agents, but ship for demonstrated reliability in the actual organization.

The field's enduring design problem is compression with accountability: enough structure to let many fallible workers act coherently, enough evidence to recover what summaries omit, and enough restraint that the organization never confuses fluent activity with legitimate accomplishment.

17.1 Further afield

Top recommended readings

Three direct foundations to reread after the synthesis, ranked for their combined explanatory reach rather than as a canon excluding the rest of the book.

- 1. Simon (1947), *Administrative Behavior*.**

Return to the decision-centered account with the book's mechanisms in view. Its foundational question is why limited participants need arrangements that shape information, attention, and choice. That perspective keeps organization from becoming an elaborate collection of agent names. Read for the conditions under which structure helps and what it obscures, not for a timeless optimum org chart.

- 2. Hübner, Sichman, and Boissier (2002b), *A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems*.**

MOISE+ turns part of that organizational problem into distinct computational descriptions. After the synthesis, trace how a role, mission, and obligation remain connected when an occupant changes. The historically important move is explicit organizational structure beyond individual behavior; formal consistency still does not establish participant competence, lawful deployment, or successful outcomes.

- 3. Parasuraman, Sheridan, and Wickens (2000), *A Model for Types and Levels of Human Interaction with Automation*.**

This framework provides the practical restraint needed alongside growing agent capability: automation can be allocated differently across observation, analysis, selection, and action. Read it again when deciding what to delegate next. Neither a capable worker nor a mature interface supplies a universal answer; representative local evidence must justify the chosen scope and oversight burden.

Broader explorations

Whose agency is the organization expanding? Sen distinguishes people's well-being from their agency and freedom to pursue valued goals [Sen 1985]. This perspective shifts evaluation beyond the number of tasks completed or the preferences inferred from observed behavior.

An artificial organization might increase output while reducing a person's ability to understand a decision, challenge it, or choose a different course. A worthwhile research direction is to evaluate those possibilities alongside efficiency and reliability. This is a normative connection, not a claim that software agents possess the human agency discussed by Sen. It leaves the reader with a question that technical performance alone cannot answer: what forms of human action should the institution make possible, and for whom?

Evaluate real opportunities, not only completed tasks. Martha Nussbaum's *Creating Capabilities* develops an account of human development concerned with what people are actually able to do and be [Nussbaum 2011]. It offers a normative counterpart to operational measures such as throughput and accepted outcomes. Ask whether an artificial organization expands people's opportunities to understand, participate, learn, and challenge decisions, or makes them more dependent on a system they cannot influence. The capabilities approach does not supply an automatic technical score. It requires choices about valued opportunities and whose situation is being assessed, which must remain explicit in the evaluation.

Justify the arrangement to those subject to it. Rawls' *A Theory of Justice* asks how principles for social institutions might be justified under a conception of fairness [Rawls 1971]. Read it as an invitation to examine distribution and institutional design, not just aggregate efficiency. Who bears the risks, who receives the benefits, and what rights constrain the pursuit of a shared objective? Applying this perspective to an artificial organization requires attention to its actual legal and social context. A thought experiment about fairness is not proof that a particular automated decision is just, but it can expose assumptions hidden by a single organizational success metric.

Correction needs both voice and alternatives. Hirschman's *Exit, Voice, and Loyalty* provides a practical way to revisit accountability at the end of the study [Hirschman 1970]. Can affected people object, receive an answer, and change an arrangement, or can they only stop using it? What makes either option costly or ineffective? A system can have excellent internal audit logs while offering weak remedies to outsiders. Evaluate the available routes of correction alongside the technical quality of decisions. The framework does not make complaints conclusive or switching costless; it helps distinguish mechanisms that a generic claim of “user control” would otherwise merge.

Preserve people as contributors to knowledge. Fricker's *Epistemic Injustice* asks how people can be wronged as knowers [Fricker 2007]. It closes an important loop in this monograph: evidence is not only something a system stores, but something institutions recognize, interpret, and sometimes discount. Ask whose testimony is missing, which experiences the categories cannot express, and how an affected person can correct the record. The aim is not to assign a moral status to software agents by analogy. It is to ensure that increasing organizational capability does not reduce the ability of people to contribute knowledge about the decisions that shape their lives.

References

The works cited in the text and Further afield sections are grouped below by subject. Links lead to available texts or publication records; some require access. News reports and practical resources are also cited in footnotes beside the relevant passages. Image credits follow as a separate section.

Interdisciplinary exploration

- Anderson, M. C., Bjork, R. A., & Bjork, E. L. (1994). "Remembering Can Cause Forgetting: Retrieval Dynamics in Long-Term Memory." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(5), 1063–1087. <https://doi.org/10.1037/0278-7393.20.5.1063>.
- Ashforth, B. E., & Mael, F. (1989). "Social Identity Theory and the Organization." *Academy of Management Review*, 14(1), 20–39. <https://doi.org/10.5465/amr.1989.4278999>.
- Clark, A., & Chalmers, D. J. (1998). "The Extended Mind." *Analysis*, 58(1), 7–19. <https://doi.org/10.1093/analys/58.1.7>. Author-hosted text.
- Cronbach, L. J., & Meehl, P. E. (1955). "Construct Validity in Psychological Tests." *Psychological Bulletin*, 52(4), 281–302. <https://doi.org/10.1037/h0040957>.
- Dixit, A. K., & Pindyck, R. S. (1994). *Investment under Uncertainty*. Princeton University Press. <https://doi.org/10.1515/9781400830176>.
- Einstein, G. O., & McDaniel, M. A. (1990). "Normal Aging and Prospective Memory." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16(4), 717–726. <https://doi.org/10.1037/0278-7393.16.4.717>.
- Fischhoff, B. (1975). "Hindsight Is Not Equal to Foresight: The Effect of Outcome Knowledge on Judgment under Uncertainty." *Journal of Experimental Psychology: Human Perception and Performance*, 1(3), 288–299. <https://doi.org/10.1037/0096-1523.1.3.288>.
- Holling, C. S. (1973). "Resilience and Stability of Ecological Systems." *Annual Review of Ecology and Systematics*, 4, 1–23. <https://doi.org/10.1146/annurev.es.04.110173.000245>.
- Knight, F. H. (1921). *Risk, Uncertainty, and Profit*. Houghton Mifflin. Part III, Chapter VII.
- Matzinger, P. (1994). "Tolerance, Danger, and the Extended Family." *Annual Review of Immunology*, 12, 991–1045. <https://doi.org/10.1146/annurev.iy.12.040194.005015>.
- Mellers, B., Hertwig, R., & Kahneman, D. (2001). "Do Frequency Representations Eliminate Conjunction Effects? An Exercise in Adversarial Collaboration." *Psychological Science*, 12(4), 269–275. <https://doi.org/10.1111/1467-9280.00350>.
- Ostrom, E. (1990). *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press. [Accessible scholarly overview](#).
- Sen, A. (1985). "Well-Being, Agency and Freedom: The Dewey Lectures 1984." *Journal of Philosophy*, 82(4), 169–221. <https://doi.org/10.2307/2026184>.
- Star, S. L., & Griesemer, J. R. (1989). "Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907–39." *Social Studies of Science*, 19(3), 387–420. <https://doi.org/10.1177/030631289019003001>.

- Stasser, G., & Titus, W. (1985). "Pooling of Unshared Information in Group Decision Making: Biased Information Sampling during Discussion." *Journal of Personality and Social Psychology*, 48(6), 1467–1478. <https://doi.org/10.1037/0022-3514.48.6.1467>.
- Teufel, S., Siddharthan, A., & Tidhar, D. (2006). "Automatic Classification of Citation Function." *Proceedings of EMNLP 2006*, 103–110. <https://aclanthology.org/W06-1613/>.
- Theraulaz, G., & Bonabeau, E. (1999). "A Brief History of Stigmergy." *Artificial Life*, 5(2), 97–116. <https://doi.org/10.1162/106454699568700>.
- Tyler, T. R. (2006). "Psychological Perspectives on Legitimacy and Legitimation." *Annual Review of Psychology*, 57, 375–400. <https://doi.org/10.1146/annurev.psych.57.102904.190038>.

Further interdisciplinary reading

- Abbott, A. (1988). *The System of Professions: An Essay on the Division of Expert Labor*. University of Chicago Press. <https://doi.org/10.7208/chicago/9780226189666.001.0001>.
- Alexander, C. (1964). *Notes on the Synthesis of Form*. Harvard University Press. ISBN 9780674627512. Bibliographic reading lead; no full-text inspection claimed.
- Argyris, C., & Schön, D. A. (1978). *Organizational Learning: A Theory of Action Perspective*. Addison-Wesley. ISBN 0201001748.
- Arrow, K. J. (1963). *Social Choice and Individual Values*, 2nd ed. Yale University Press; first edition 1951. Digital reissue: <https://doi.org/10.12987/9780300186987>.
- Aumann, R. J. (1976). "Agreeing to Disagree." *The Annals of Statistics*, 4(6), 1236–1239. <https://doi.org/10.1214/aos/1176343654>.
- Austin, J. L. (1975). *How to Do Things with Words*, 2nd ed., J. O. Urmson and M. Sbisà, eds. Oxford University Press; first edition 1962. <https://doi.org/10.1093/acprof:oso/9780198245537.001.0001>.
- Bowker, G. C., & Star, S. L. (1999). *Sorting Things Out: Classification and Its Consequences*. MIT Press. <https://doi.org/10.7551/mitpress/6352.001.0001>.
- Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton University Press. Digital reissue: <https://doi.org/10.2307/j.ctvzxx9tx>.
- Campbell, D. T., & Stanley, J. C. (1966). *Experimental and Quasi-Experimental Designs for Research*. Houghton Mifflin; originally published as a handbook chapter in 1963. ISBN 0395307872.
- Crawford, S. E. S., & Ostrom, E. (1995). "A Grammar of Institutions." *American Political Science Review*, 89(3), 582–600. <https://doi.org/10.2307/2082975>.
- Douglas, M., & Wildavsky, A. (1982). *Risk and Culture: An Essay on the Selection of Technological and Environmental Dangers*. University of California Press. Digital reissue: <https://doi.org/10.1525/9780520907393>.
- Endsley, M. R. (1995). "Toward a Theory of Situation Awareness in Dynamic Systems." *Human Factors*, 37(1), 32–64. <https://doi.org/10.1518/001872095779049543>.
- Fricker, M. (2007). *Epistemic Injustice: Power and the Ethics of Knowing*. Oxford University Press. <https://doi.org/10.1093/acprof:oso/9780198237907.001.0001>.
- Fuller, L. L. (1969). *The Morality of Law*, revised ed. Yale University Press; first edition 1964. ISBN 9780300010701.
- Funtowicz, S. O., & Ravetz, J. R. (1993). "Science for the Post-Normal Age." *Futures*, 25(7), 739–755. [https://doi.org/10.1016/0016-3287\(93\)90022-L](https://doi.org/10.1016/0016-3287(93)90022-L).
- Gieryn, T. F. (1983). "Boundary-Work and the Demarcation of Science from Non-Science: Strains and Interests in Professional Ideologies of Scientists." *American Sociological Review*, 48(6), 781–795. <https://doi.org/10.2307/2095325>.

- Goffman, E. (1959). *The Presentation of Self in Everyday Life*. Doubleday Anchor. ISBN 9780385094023; the earlier Edinburgh edition appeared in 1956.
- Granovetter, M. (1978). "Threshold Models of Collective Behavior." *American Journal of Sociology*, 83(6), 1420–1443. <https://doi.org/10.1086/226707>.
- Hart, H. L. A. (1961). *The Concept of Law*. Clarendon Press, Oxford. Later editions add a postscript and editorial material; consult the edition cited.
- Hayek, F. A. (1945). "The Use of Knowledge in Society." *American Economic Review*, 35(4), 519–530. <https://www.econlib.org/library/Essays/hyk-Knw.html>.
- Hirschman, A. O. (1970). *Exit, Voice, and Loyalty: Responses to Decline in Firms, Organizations, and States*. Harvard University Press. ISBN 9780674276604.
- Hong, L., & Page, S. E. (2004). "Groups of Diverse Problem Solvers Can Outperform Groups of High-Ability Problem Solvers." *PNAS*, 101(46), 16385–16389. <https://doi.org/10.1073/pnas.0403723101>.
- Hutchins, E. (1995). *Cognition in the Wild*. MIT Press. <https://doi.org/10.7551/mitpress/1881.001.0001>.
- Ioannidis, J. P. A. (2005). "Why Most Published Research Findings Are False." *PLoS Medicine*, 2(8), e124. <https://doi.org/10.1371/journal.pmed.0020124>.
- Kuhn, T. S. (1996). *The Structure of Scientific Revolutions*, 3rd ed. University of Chicago Press; first edition 1962. <https://doi.org/10.7208/chicago/9780226458106.001.0001>.
- Lakatos, I. (1978). *The Methodology of Scientific Research Programmes*, J. Worrall and G. Currie, eds. Cambridge University Press. <https://doi.org/10.1017/CBO9780511621123>.
- Latour, B. (2005). *Reassembling the Social: An Introduction to Actor-Network Theory*. Oxford University Press. <https://doi.org/10.1093/oso/9780199256044.001.0001>.
- Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511815355>.
- Leveson, N. G. (2012). *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press. <https://doi.org/10.7551/mitpress/8179.001.0001>.
- Levitt, B., & March, J. G. (1988). "Organizational Learning." *Annual Review of Sociology*, 14, 319–338. <https://doi.org/10.1146/annurev.so.14.080188.001535>.
- Lindblom, C. E. (1959). "The Science of 'Muddling Through.'" *Public Administration Review*, 19(2), 79–88. <https://doi.org/10.2307/973677>.
- Longino, H. E. (1990). *Science as Social Knowledge: Values and Objectivity in Scientific Inquiry*. Princeton University Press. Digital reissue: <https://doi.org/10.2307/j.ctvx5wbfbz>.
- March, J. G. (1991). "Exploration and Exploitation in Organizational Learning." *Organization Science*, 2(1), 71–87. <https://doi.org/10.1287/orsc.2.1.71>.
- Merton, R. K. (1968). "The Matthew Effect in Science." *Science*, 159(3810), 56–63. <https://doi.org/10.1126/science.159.3810.56>.
- Norman, D. A. (2013). *The Design of Everyday Things*, revised and expanded ed. Basic Books. ISBN 9780465050659.
- Nussbaum, M. C. (2011). *Creating Capabilities: The Human Development Approach*. Harvard University Press. <https://doi.org/10.4159/harvard.9780674061200>.
- Olson, M. (1965). *The Logic of Collective Action: Public Goods and the Theory of Groups*. Harvard University Press. ISBN 9780674537514.
- Pearl, J. (2009). "Causal Inference in Statistics: An Overview." *Statistics Surveys*, 3, 96–146. <https://doi.org/10.1214/09-SS057>.
- Perrow, C. (1984). *Normal Accidents: Living with High-Risk Technologies*. Basic Books; Princeton edition with a new afterword, 1999. Digital reissue: <https://doi.org/10.2307/j.ctt7srgf>.
- Polanyi, M. (1966). *The Tacit Dimension*. Doubleday; University of Chicago Press reissue, 2009. ISBN 9780226672984.

- Popper, K. R. (1959). *The Logic of Scientific Discovery*. Hutchinson; English edition of the earlier German work. Routledge digital reissue: <https://doi.org/10.4324/9780203994627>.
- Price, D. J. de Solla (1965). "Networks of Scientific Papers." *Science*, 149(3683), 510–515. <https://doi.org/10.1126/science.149.3683.510>.
- Rasmussen, J. (1997). "Risk Management in a Dynamic Society: A Modelling Problem." *Safety Science*, 27(2–3), 183–213. [https://doi.org/10.1016/S0925-7535\(97\)00052-0](https://doi.org/10.1016/S0925-7535(97)00052-0).
- Rawls, J. (1971). *A Theory of Justice*. Harvard University Press. <https://doi.org/10.4159/9780674042605>.
- Reason, J. (1990). *Human Error*. Cambridge University Press. <https://doi.org/10.1017/CBO9781139062367>.
- Schelling, T. C. (1971). "Dynamic Models of Segregation." *Journal of Mathematical Sociology*, 1(2), 143–186. <https://doi.org/10.1080/0022250X.1971.9989794>.
- Scott, J. C. (1998). *Seeing Like a State: How Certain Schemes to Improve the Human Condition Have Failed*. Yale University Press. Digital reissue: <https://doi.org/10.12987/9780300252989>.
- Simon, H. A. (1996). *The Sciences of the Artificial*, 3rd ed. MIT Press. Reissue with a new introduction, 2019: <https://doi.org/10.7551/mitpress/12107.001.0001>.
- Star, S. L., & Ruhleder, K. (1996). "Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces." *Information Systems Research*, 7(1), 111–134. <https://doi.org/10.1287/isre.7.1.111>.
- Stirling, A. (2010). "Keep It Complex." *Nature*, 468, 1029–1031. <https://doi.org/10.1038/4681029a>.
- Strauss, A. (1985). "Work and the Division of Labor." *The Sociological Quarterly*, 26(1), 1–19. <https://doi.org/10.1111/j.1533-8525.1985.tb00212.x>.
- Strogatz, S. H., & Stewart, I. (1993). "Coupled Oscillators and Biological Synchronization." *Scientific American*, 269(6), 102–109. <https://doi.org/10.1038/scientificamerican1293-102>.
- Suchman, L. (2007). *Human-Machine Reconfigurations: Plans and Situated Actions*, 2nd ed. Cambridge University Press. <https://doi.org/10.1017/CBO9780511808418>.
- Sweller, J. (1988). "Cognitive Load During Problem Solving: Effects on Learning." *Cognitive Science*, 12(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4.
- Turner, B. A. (1976). "The Organizational and Interorganizational Development of Disasters." *Administrative Science Quarterly*, 21(3), 378–397. <https://doi.org/10.2307/2391850>.
- Tversky, A., & Kahneman, D. (1974). "Judgment under Uncertainty: Heuristics and Biases." *Science*, 185(4157), 1124–1131. <https://doi.org/10.1126/science.185.4157.1124>.
- Vaughan, D. (1996). *The Challenger Launch Decision: Risky Technology, Culture, and Deviance at NASA*. University of Chicago Press. <https://doi.org/10.7208/chicago/9780226346960.001.0001>.
- Wickens, C. D. (2002). "Multiple Resources and Performance Prediction." *Theoretical Issues in Ergonomics Science*, 3(2), 159–177. <https://doi.org/10.1080/14639220210123806>.
- Woods, D. D. (2015). "Four Concepts for Resilience and the Implications for the Future of Resilience Engineering." *Reliability Engineering & System Safety*, 141, 5–9. <https://doi.org/10.1016/j.ress.2015.03.018>.
- Wynne, B. (1992). "Misunderstood Misunderstanding: Social Identities and Public Uptake of Science." *Public Understanding of Science*, 1(3), 281–304. <https://doi.org/10.1088/0963-6625/1/3/004>.

Organization science and human factors

- Bainbridge, L. (1983). "Ironies of Automation." *Automatica*, 19(6), 775–779.
[https://doi.org/10.1016/0005-1098\(83\)90046-8](https://doi.org/10.1016/0005-1098(83)90046-8)
- Bliss, J. P., Gilson, R. D., & Deaton, J. E. (1995). "Human Probability Matching Behaviour in Response to Alarms of Varying Reliability." *Ergonomics*, 38(11), 2300–2312.
<https://doi.org/10.1080/00140139508925269>
- Bolton, P., & Dewatripont, M. (1994). "The Firm as a Communication Network." *Quarterly Journal of Economics*, 109(4), 809–839.
<https://doi.org/10.2307/2118349>
- Drogoul, A., & Collinot, A. (1998). "Applying an Agent-Oriented Methodology to the Design of Artificial Organizations: A Case Study in Robotic Soccer." *Autonomous Agents and Multi-Agent Systems*, 1(1), 113–129.
<https://doi.org/10.1023/A:1010098623921>
- Fourney, A. et al. (2024). *Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks*. arXiv:2411.04468v1.
<https://arxiv.org/abs/2411.04468v1>
- Garicano, L. (2000). "Hierarchies and the Organization of Knowledge in Production." *Journal of Political Economy*, 108(5), 874–904.
<https://doi.org/10.1086/317671>
- Goodhart, C. A. E. (1984). "Problems of Monetary Management: The U.K. Experience." In *Monetary Theory and Practice*. Macmillan.
- La Malfa, E., La Malfa, G., Marro, S., Zhang, J. M., Black, E., Luck, M., Torr, P., & Wooldridge, M. (2025). *Large Language Models Miss the Multi-Agent Mark*. arXiv:2505.21298v4, 6 December 2025.
<https://arxiv.org/abs/2505.21298v4>
- March, J. G., Sproull, L. S., & Tamuz, M. (1991). "Learning from Samples of One or Fewer." *Organization Science*, 2(1), 1–13.
<https://doi.org/10.1287/orsc.2.1.1>
- McFarlane, D. C. (2002). "Comparison of Four Primary Methods for Coordinating the Interruption of People in Human–Computer Interaction." *Human–Computer Interaction*, 17(1), 63–139. https://doi.org/10.1207/S15327051HCI1701_2
- Meyer, J. (2001). "Effects of Warning Validity and Proximity on Responses to Warnings." *Human Factors*, 43(4), 563–572.
<https://doi.org/10.1518/001872001775870395>
- Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). "A Model for Types and Levels of Human Interaction with Automation." *IEEE Transactions on Systems, Man, and Cybernetics A*, 30(3), 286–297.
<https://doi.org/10.1109/3468.844354>
- Radner, R. (1993). "The Organization of Decentralized Information Processing." *Econometrica*, 61(5), 1109–1146.
<https://doi.org/10.2307/2951495>
- Ridgway, V. F. (1956). "Dysfunctional Consequences of Performance Measurements." *Administrative Science Quarterly*, 1(2), 240–247.
<https://doi.org/10.2307/2390989>
- Sah, R. K., & Stiglitz, J. E. (1986). "The Architecture of Economic Systems: Hierarchies and Polyarchies." *American Economic Review*, 76(4), 716–727. <https://www.nber.org/papers/w1334>
- Simon, H. A. (1947). *Administrative Behavior*. Macmillan. First edition; the linked catalogue describes the 1957 second edition.
<https://archive.org/details/administrativebe00si-mo>
- Simon, H. A. (1955). "A Behavioral Model of Rational Choice." *Quarterly Journal of Economics*, 69(1), 99–118. <https://doi.org/10.2307/1884852>
- Waites, W. (2026). *Artificial Organisations*. arXiv:2602.13275v1.
<https://arxiv.org/abs/2602.13275v1>.
Contemporary institutional-design proposal with observational evidence, not a standard definition of the field.

- Walsh, J. P., & Ungson, G. R. (1991). "Organizational Memory." *Academy of Management Review*, 16(1), 57–91. <https://doi.org/10.2307/258607>
- Wegner, D. M. (1987). "Transactive Memory: A Contemporary Analysis of the Group Mind." In B. Mullen & G. R. Goethals (Eds.), *Theories of Group Behavior*. Springer. https://doi.org/10.1007/978-1-4612-4634-3_9
- Ye, M., & Carley, K. M. (1995). "RADAR-Soar: Towards an Artificial Organization Composed of Intelligent Agents." *Journal of Mathematical Sociology*, 20(2–3), 219–246. <https://doi.org/10.1080/0022250X.1995.9990163>

Organizations, institutions, and coordination in MAS

- Cohen, P. R., & Levesque, H. J. (1991). "Teamwork." *Noûs*, 25(4), 487–512. <https://doi.org/10.2307/2216075>
- Erman, L. D., Hayes-Roth, F., Lesser, V. R., & Reddy, D. R. (1980). "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty." *ACM Computing Surveys*, 12(2), 213–253. <https://doi.org/10.1145/356810.356816>
- Ferber, J., & Gutknecht, O. (1998). "A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems." ICMAS 1998. <https://doi.org/10.1109/ICMAS.1998.699041>
- Fox, M. S., Barbuceanu, M., & Grüninger, M. (1996). "An Organisation Ontology for Enterprise Modeling: Preliminary Concepts for Linking Structure and Behaviour." *Computers in Industry*, 29(1–2), 123–134. [https://doi.org/10.1016/0166-3615\(95\)2900079-8](https://doi.org/10.1016/0166-3615(95)2900079-8)
- Hübner, J. F., Sichman, J. S., & Boissier, O. (2002a). "MOISE+: Towards a Structural, Functional, and Deontic Model for MAS Organization." AAMAS 2002. <https://doi.org/10.1145/544741.544858>
- Jennings, N. R. (1993). "Commitments and Conventions: The Foundation of Coordination in Multi-Agent Systems." *Knowledge Engineering Review*, 8(3), 223–250.
- Jones, A. J. I., & Sergot, M. (1996). "A Formal Characterisation of Institutionalised Power." *Logic Journal of the IGPL*, 4(3), 427–443. <https://doi.org/10.1093/jigpal/4.3.427>
- Singh, M. P. (1998). "Agent Communication Languages: Rethinking the Principles." *Computer*, 31(12), 40–47. <https://doi.org/10.1109/2.735849>
- Smith, R. G. (1980). "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver." *IEEE Transactions on Computers*, C-29(12), 1104–1113. <https://doi.org/10.1109/TC.1980.1675516>
- Tambe, M. (1997). "Towards Flexible Teamwork." *Journal of Artificial Intelligence Research*, 7, 83–124. <https://www.jair.org/index.php/jair/article/view/10193>

Evidence, argumentation, and decisions

- Dung, P. M. (1995). "On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games." *Artificial Intelligence*, 77(2), 321–357. [https://doi.org/10.1016/0004-3702\(95\)2900041-X](https://doi.org/10.1016/0004-3702(95)2900041-X)
- Jøsang, A. (2016). *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Springer. <https://doi.org/10.1007/978-3-319-42337-1>
- Kunz, W., & Rittel, H. W. J. (1970). *Issues as Elements of Information Systems*. University of California, Berkeley. https://web.archive.org/web/20251116061957id_/https://escholarship.org/content/qt5cj786v8/qt5cj786v8.pdf

- MacLean, A., Young, R. M., Bellotti, V. M. E., & Moran, T. P. (1991). "Questions, Options, and Criteria: Elements of Design Space Analysis." *Human-Computer Interaction*, 6(3-4), 201-250.
- McBurney, P., & Parsons, S. (2002). "Games That Agents Play: A Formal Framework for Dialogues Between Autonomous Agents." *Journal of Logic, Language and Information*, 11, 315-334. <https://doi.org/10.1023/A:1015586128739>
- McBurney, P., Hitchcock, D., & Parsons, S. (2007; online 2006). "The Eightfold Way of Deliberation Dialogue." *International Journal of Intelligent Systems*, 22(1), 95-132. <https://doi.org/10.1002/int.20191>
- Nemeth, C., Brown, K., & Rogers, J. (2001). "Devil's Advocate versus Authentic Dissent: Stimulating Quantity and Quality." *European Journal of Social Psychology*, 31(6), 707-720. <https://doi.org/10.1002/ejsp.58>
- Searle, J. R. (1995). *The Construction of Social Reality*. Free Press.
- Toulmin, S. E. (1958). *The Uses of Argument*. Cambridge University Press. <https://archive.org/details/usesofargument0000-toul>
- W3C. (2013). *PROV-DM: The PROV Data Model*. W3C Recommendation. <https://www.w3.org/TR/prov-dm/>
- Backlund, A., & Petersson, L. (2025). *Vending-Bench: A Benchmark for Long-Term Coherence of Autonomous Agents*. <https://arxiv.org/abs/2502.15840>
- Hong, S. et al. (2024; preprint 2023). "MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework." ICLR 2024. <https://arxiv.org/abs/2308.00352>
- Khan, A. et al. (2024). "Debating with More Persuasive LLMs Leads to More Truthful Answers." <https://arxiv.org/abs/2402.06782>
- Kim, S. S. Y. et al. (2025). "Fostering Appropriate Reliance on Large Language Models: The Role of Explanations, Sources, and Inconsistencies." CHI 2025. <https://arxiv.org/abs/2502.08554>
- Kwa, T. et al. (2025). "Measuring AI Ability to Complete Long Tasks." <https://arxiv.org/abs/2503.14499>
- Lanham, T. et al. (2023). "Measuring Faithfulness in Chain-of-Thought Reasoning." <https://arxiv.org/abs/2307.13702>
- Park, J. S. et al. (2023). "Generative Agents: Interactive Simulacra of Human Behavior." UIST 2023. <https://arxiv.org/abs/2304.03442>
- Shinn, N. et al. (2023). "Reflexion: Language Agents with Verbal Reinforcement Learning." NeurIPS 2023. <https://arxiv.org/abs/2303.11366>
- Turpin, M. et al. (2023). "Language Models Don't Always Say What They Think." NeurIPS 2023. <https://arxiv.org/abs/2305.04388>
- Wang, Q. et al. (2024). "Rethinking the Bounds of LLM Reasoning: Are Multi-Agent Discussions the Key?" ACL 2024. <https://arxiv.org/abs/2402.18272>
- Wu, Q. et al. (2023). "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation." <https://arxiv.org/abs/2308.08155>
- Xie, T. et al. (2024). *OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments*. <https://arxiv.org/abs/2404.07972>
- Yao, S. et al. (2023; preprint 2022). "ReAct: Synergizing Reasoning and Acting in Language Models." ICLR 2023. <https://arxiv.org/abs/2210.03629>

Further foundations used in the deeper treatments

- Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). "Tropos: An Agent-Oriented Software Development Methodology." *Autonomous Agents and Multi-Agent Systems*, 8, 203–236.
<https://doi.org/10.1023/B:AGNT.0000018806.20944.ef>
- Brier, G. W. (1950). "Verification of Forecasts Expressed in Terms of Probability." *Monthly Weather Review*, 78(1), 1–3.
- Coase, R. H. (1937). "The Nature of the Firm." *Economica*, 4(16), 386–405.
<https://doi.org/10.1111/j.1468-0335.1937.tb00002.x>
- Cyert, R. M., & March, J. G. (1963). *A Behavioral Theory of the Firm*. Prentice-Hall.
- Dignum, V. (2004). *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. PhD thesis, Utrecht University.
- Garcia-Molina, H., & Salem, K. (1987). "Sagas." *Proceedings of ACM SIGMOD*, 249–259.
<https://doi.org/10.1145/38714.38742>
- Grossman, S. J., & Hart, O. D. (1986). "The Costs and Benefits of Ownership: A Theory of Vertical and Lateral Integration." *Journal of Political Economy*, 94(4), 691–719.
<https://doi.org/10.1086/261404>
- Horling, B., & Lesser, V. (2004). "A Survey of Multi-Agent Organizational Paradigms." *The Knowledge Engineering Review*, 19(4), 281–316.
<https://doi.org/10.1017/S0269888905000317>
- Joskow, P. L. (1985). "Vertical Integration and Long-term Contracts: The Case of Coal-burning Electric Generating Plants." *Journal of Law, Economics, and Organization*, 1(1), 33–80.
<https://doi.org/10.1093/oxfordjournal-s.jleo.a036889>
- Malone, T. W., & Crowston, K. (1994). "The Interdisciplinary Study of Coordination." *ACM Computing Surveys*, 26(1), 87–119.
<https://doi.org/10.1145/174666.174668>
- March, J. G., & Simon, H. A. (1958). *Organizations*. Wiley.
- Padget, J., ElDeen Elakehal, E., Li, T., & De Vos, M. (2016). "InstAL: An Institutional Action Language." In *Social Coordination Frameworks for Social Technical Systems*, 101–124.
- Williamson, O. E. (1979). "Transaction-Cost Economics: The Governance of Contractual Relations." *Journal of Law and Economics*, 22(2), 233–261. <https://doi.org/10.1086/466942>
- Zambonelli, F., Jennings, N. R., & Wooldridge, M. (2003). "Developing Multiagent Systems: The Gaia Methodology." *ACM Transactions on Software Engineering and Methodology*, 12(3), 317–370.
<https://doi.org/10.1145/958961.958963>

Concurrent and distributed coordination

- Arbab, F. (2004). "Reo: A Channel-Based Coordination Model for Component Composition." *Mathematical Structures in Computer Science*, 14(3), 329–366.
<https://doi.org/10.1017/S0960129504004153>
- Bailis, P., Fekete, A., Franklin, M. J., Ghodsi, A., Hellerstein, J. M., & Stoica, I. (2014). "Coordination Avoidance in Database Systems." <https://arxiv.org/abs/1402.2237v4>
- Burrows, M. (2006). "The Chubby Lock Service for Loosely-Coupled Distributed Systems." *OSDI*, 335–350. <https://storage.googleapis.com/gweb-research2023-media/pubtools/4444.pdf>
- Carriero, N., Gelernter, D., Mattson, T., & Sherman, A. (1994). "The Linda Alternative to Message-Passing Systems." *Parallel Computing*, 20, 633–655.
<https://heather.miller.am/teaching/cs7680/pdfs/Linda-Alternative-to-Message-Passing.pdf>

- Gelernter, D. (1985). “Generative Communication in Linda.” *ACM TOPLAS*, 7(1), 80–112.
<https://doi.org/10.1145/2363.2433>
- Gelernter, D., & Carriero, N. (1992). “Coordination Languages and Their Significance.” *Communications of the ACM*, 35(2), 97–107.
<https://doi.org/10.1145/129630.129635>
- Hellerstein, J. M., & Alvaro, P. (2019). “Keeping CALM: When Distributed Consistency Is Easy.”
<https://arxiv.org/abs/1901.01930v2>
- Hoare, C. A. R. (1978). “Communicating Sequential Processes.” *Communications of the ACM*, 21(8), 666–677. <https://www.cs.cmu.edu/~crary/819-f09/Hoare78.pdf>
- Hunt, P., Konar, M., Junqueira, F. P., & Reed, B. (2010). “ZooKeeper: Wait-free Coordination for Internet-scale Systems.” *USENIX ATC*.
<https://www.usenix.org/conference/usenix-atc-10/zookeeper-wait-free-coordination-internet-scale-systems>
- Milner, R., Parrow, J., & Walker, D. (1992). “A Calculus of Mobile Processes, I.” *Information and Computation*, 100(1), 1–40.
[https://doi.org/10.1016/0890-5401\(92\)90008-4](https://doi.org/10.1016/0890-5401(92)90008-4). Earlier technical report.
- Omicini, A., & Denti, E. (2001). “From Tuple Spaces to Tuple Centres.” *Science of Computer Programming*, 41(3), 277–294. [https://doi.org/10.1016/S0167-6423\(01\)00011-9](https://doi.org/10.1016/S0167-6423(01)00011-9). Detailed runtime semantics remain a further-reading lead; this update checked the bibliographic record.
- Shapiro, M., Preguiça, N., Baquero, C., & Zawirski, M. (2011). “A Comprehensive Study of Convergent and Commutative Replicated Data Types.” INRIA RR-7506.
<https://inria.hal.science/inria-00555588v1/file/techreport.pdf>

Business processes, workflows, and agent participation

- Chopra, A. K., Christie, S. H., V, & Singh, M. P. (2020). “An Evaluation of Communication Protocol Languages for Engineering Multiagent Systems.” Version inspected: arXiv:1901.08441v2.
<https://arxiv.org/html/1901.08441v2>
- Jennings, N. R., Faratin, P., Johnson, M. J., Norman, T. J., O'Brien, P., & Wiegand, M. E. (1996). “Agent-Based Business Process Management.” *International Journal of Cooperative Information Systems*.
<https://doi.org/10.1142/S0218843096000051>. Author manuscript.
- Kourani, H., Berti, A., Schuster, D., & van der Aalst, W. M. P. (2024b). “Evaluating Large Language Models on Business Process Modeling: Framework, Benchmark, and Self-Improvement Analysis.” arXiv:2412.00023v1.
<https://arxiv.org/html/2412.00023v1>
- Nigam, A., & Caswell, N. S. (2003). “Business Artifacts: An Approach to Operational Specification.” *IBM Systems Journal*, 42(3), 428–445. <https://doi.org/10.1147/sj.423.0428>. Bibliographic identity checked; detailed formal semantics are a further-reading lead.
- Object Management Group. (2014). *Business Process Model and Notation (BPMN), Version 2.0.2*.
<https://www.omg.org/spec/BPMN/2.0.2/PDF>
- Object Management Group. (2016). *Case Management Model and Notation (CMMN), Version 1.1*.
<https://www.omg.org/spec/CMMN/1.1/PDF>
- Object Management Group. (2024). *Decision Model and Notation (DMN), Version 1.5*.
<https://www.omg.org/spec/DMN/1.5/About-DMN>

- Pfeiffer, P., Rombach, A., Majlatow, M., & Mehdiyev, N. (2025). "From Theory to Practice: Real-World Use Cases on Trustworthy LLM-Driven Process Modeling, Prediction and Automation." arXiv:2506.03801v1; SIGMOD 2025 workshop contribution.
<https://arxiv.org/html/2506.03801v1>
- Schluntz, E., & Zhang, B. (2024). "Building Effective Agents." Anthropic Engineering, 19 December; live page inspected 20 September 2026.
<https://www.anthropic.com/engineering/building-effective-agents>
- van der Aalst, W. M. P. (1998). "The Application of Petri Nets to Workflow Management." *Journal of Circuits, Systems and Computers*, 8(1), 21–66.
<https://doi.org/10.1142/S0218126698000043>. Institutional record.
- van der Aalst, W. M. P., & ter Hofstede, A. H. M. (2005). "YAWL: Yet Another Workflow Language." *Information Systems*, 30(4), 245–275.
<https://doi.org/10.1016/j.is.2004.02.002>
- van der Aalst, W. M. P., ter Hofstede, A. H. M., Kiepuszewski, B., & Barros, A. P. (2003). "Workflow Patterns." *Distributed and Parallel Databases*, 14, 5–51. <https://doi.org/10.1023/A:1022883727209>. Publisher PDF in institutional repository.
- van der Aalst, W. M. P., Pesic, M., & Schonenberg, H. (2009). "Declarative Workflows: Balancing between Flexibility and Support." *Computer Science — Research and Development*, 23, 99–113.
<https://doi.org/10.1007/s00450-009-0057-9>
- van der Aalst, W. et al. (2012). "Process Mining Manifesto." *Business Process Management Workshops*, LNBIP 99, 169–194. Manifesto developed in 2011; proceedings publication in 2012.
https://doi.org/10.1007/978-3-642-28108-2_19
- Xiao, R., Ma, W., Wang, K., Wu, Y., Zhao, J., Wang, H., Huang, F., & Li, Y. (2024). "FlowBench: Revisiting and Benchmarking Workflow-Guided Planning for LLM-based Agents." arXiv:2406.14884v1.
<https://arxiv.org/html/2406.14884v1>
- Yao, S., Shinn, N., Razavi, P., & Narasimhan, K. (2024). "tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains." arXiv:2406.12045v1.
<https://arxiv.org/html/2406.12045v1>
- Ziche, C., & Apruzzese, G. (2024). "LLM4PM: A Case Study on Using Large Language Models for Process Modeling in Enterprise Organizations." BPM 2024 industry forum; arXiv:2407.17478v1.
https://doi.org/10.1007/978-3-031-70445-1_35. Inspected manuscript.

Additional classical sources: organization and supervision

- Chen, J. Y. C., Procci, K., Boyce, M., Wright, J., Garcia, A., & Barnes, M. (2014). *Situation Awareness-Based Agent Transparency*. U.S. Army Research Laboratory, ARL-TR-6905.
- Crandall, J. W., Goodrich, M. A., Olsen, D. R., Jr., & Nielsen, C. W. (2005). "Validating Human–Robot Interaction Schemes in Multitasking Environments." *IEEE Transactions on Systems, Man, and Cybernetics A*, 35(4), 438–449. <https://scholar.archive.byu.edu/cgi/viewcontent.cgi?article=1362&context=facpub>
- Galbraith, J. R. (1973). *Designing Complex Organizations*. Addison-Wesley.
<https://archive.org/details/designingcomplex00-galb>
- Holt, C. C., Modigliani, F., Muth, J. F., & Simon, H. A. (1960). *Planning Production, Inventories, and Work Force*. Prentice-Hall.
<https://archive.org/details/PlanningProductionInventoriesAndWorkForce>
- Lee, J. D., & See, K. A. (2004). "Trust in Automation: Designing for Appropriate Reliance." *Human Factors*, 46(1), 50–80.
https://doi.org/10.1518/hfes.46.1.50_30392
- Meyer, J. (2004). "Conceptual Issues in the Study of Dynamic Hazard Warnings." *Human Factors*, 46(2), 196–204.
- Mintzberg, H. (1979). *The Structuring of Organizations*. Prentice-Hall. <https://archive.org/details/structuringoforg0000mint>

- Selznick, P. (1949). *TVA and the Grass Roots: A Study in the Sociology of Formal Organization*. University of California Press. <https://archive.org/details/in.ernet.dli.2015.286>
- Sheridan, T. B., & Verplank, W. L. (1978). *Human and Computer Control of Undersea Teleoperators*. MIT Man-Machine Systems Laboratory. https://archive.org/details/DTIC_ADA057655
- Simon, H. A. (1962). "The Architecture of Complexity." *Proceedings of the American Philosophical Society*, 106(6), 467–482.
- Thompson, J. D. (1967). *Organizations in Action*. McGraw-Hill. <https://archive.org/details/organizationsina0000thom>

Additional classical sources: institutions and coordination

- Bernstein, D. S., Zilberstein, S., & Immerman, N. (2000). "The Complexity of Decentralized Control of Markov Decision Processes." *UAI*, 32–37. Conference version; deposited in 2013. <https://arxiv.org/abs/1301.3836>
- Broersen, J., Dastani, M., Hulstijn, J., Huang, Z., & van der Torre, L. (2001). "The BOID Architecture: Conflicts Between Beliefs, Obligations, Intentions and Desires." *Proceedings of Agents '01*, 9–16. <https://dspace.library.uu.nl/handle/1874/19919>
- Dechter, R. (1999). "Bucket Elimination: A Unifying Framework for Reasoning." *Artificial Intelligence*, 113, 41–85. <https://www.ics.uci.edu/~dechter/publications/r76A.pdf>
- Dignum, V., Vázquez-Salceda, J., & Dignum, F. (2005). "OMNI: Introducing Social Structure, Norms and Ontologies into Agent Organizations." In *Programming Multi-Agent Systems*, ProMAS 2004 post-proceedings, 181–198. <https://dspace.library.uu.nl/handle/1874/11498>
- Esteva, M., de la Cruz, D., & Sierra, C. (2002). "ISLANDER: An Electronic Institutions Editor." *AAMAS '02*. <https://doi.org/10.1145/545056.545069>
- Esteva, M., Rodríguez-Aguilar, J. A., Sierra, C., Garcia, P., & Arcos, J.-L. (2001). "On the Formal Specification of Electronic Institutions." In *Agent-Mediated Electronic Commerce*, LNAI 1991. https://doi.org/10.1007/3-540-44682-6_8
- Esteva, M., Rosell, B., Rodríguez-Aguilar, J. A., & Arcos, J. L. (2004). "AMELI: An Agent-based Middleware for Electronic Institutions." *AAMAS 2004*. <https://doi.org/10.65109/fefg8977>
- Finin, T., Fritzson, R., McKay, D., & McEntire, R. (1994). "KQML as an Agent Communication Language." *CIKM '94*. https://ebiquity.umbc.edu/_file_directory_/papers/318.pdf
- Fioretto, F., Pontelli, E., & Yeoh, W. (2018). "Distributed Constraint Optimization Problems and Applications: A Survey." *JAIR*, 61, 623–698. <https://arxiv.org/abs/1602.06347v4>
- Foundation for Intelligent Physical Agents. (2002). *FIPA ACL Message Structure Specification and FIPA Communicative Act Library Specification*. Documents SC00061G and SC00037J. The former FIPA website no longer supplied the standards when checked on 15 September 2026; a usable archival copy remains a research lead rather than a verified specimen in this edition.
- Grossi, D. (2007). *Designing Invisible Handcuffs: Formal Investigations in Institutions and Organizations for Multi-Agent Systems*. PhD thesis, Utrecht. <https://dspace.library.uu.nl/handle/1874/22838>
- Grossi, D., Meyer, J.-J., & Dignum, F. (2006). "Classificatory Aspects of Counts-as: An Analysis in Modal Logic." *Journal of Logic and Computation*, 16(5). <https://doi.org/10.1093/logcom/exl027>
- Guestrin, C., Koller, D., & Parr, R. (2001). "Multi-agent Planning with Factored MDPs." *Advances in Neural Information Processing Systems*, 14. https://proceedings.neurips.cc/paper_files/paper/2001/hash/7af6266cc52234b5aa339b16695f7fc4-Abstract.html

- Hübner, J. F., Sichman, J. S., & Boissier, O. (2002b). "A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems." *SBIA 2002*, LNCS 2507, 118–128. Companion to the AAMAS 2002 extended abstract listed above as 2002a. <https://hal-emse.ccsd.cnrs.fr/emse-00745177v1>
- Leitão, P., & Restivo, F. (2006). "ADACOR: A Holonic Architecture for Agile and Adaptive Manufacturing Control." *Computers in Industry*, 57(2), 121–130.
- Oliehoek, F. A., & Amato, C. (2016). *A Concise Introduction to Decentralized POMDPs*. Springer. Author preprint dated July 2015. <https://www.fransoliehoek.net/docs/Oliehoek-Amato16book.pdf>
- Saltzer, J. H., & Schroeder, M. D. (1975). "The Protection of Information in Computer Systems." *Proceedings of the IEEE*, 63(9), 1278–1308. <https://web.mit.edu/Saltzer/www/publications/protection/>
- Singh, M. P. (1999). "An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts." *Artificial Intelligence and Law*, 7. <https://doi.org/10.1023/A:1008319631231>
- Stone, P., & Veloso, M. (2000). "Multiagent Systems: A Survey from a Machine Learning Perspective." *Autonomous Robots*, 8(3), 345–383. <https://www.cs.utexas.edu/~pstone/Papers/bib2html-links/MASsurvey.pdf>
- Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., & Peeters, P. (1998). "Reference Architecture for Holonic Manufacturing Systems: PROSA." *Computers in Industry*, 37(3), 255–274.
- Wooldridge, M., & Jennings, N. R. (1998). "Pitfalls of Agent-Oriented Development." *Proceedings of the Second International Conference on Autonomous Agents*, 385–391. <https://www.cs.ox.ac.uk/people/michael-wooldridge/pubs/agents98.pdf>
- Wurman, P. R., D'Andrea, R., & Mountz, M. (2008). "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses." *AI Magazine*, 29(1), 9–20. <https://doi.org/10.1609/aimag.v29i1.2082>
- Yeoh, W., & Yokoo, M. (2012). "Distributed Problem Solving." *AI Magazine*, 33(3), 53–65. <https://doi.org/10.1609/aimag.v33i3.2429>
- Yolum, P., & Singh, M. P. (2002). "Commitment Machines." In *Intelligent Agents VIII*. https://doi.org/10.1007/3-540-45448-9_17

Additional classical sources: trust, argumentation, and memory

- Admiralty-style source and information grading. Historical institutional practice; definitions vary by manual. The secondary overview below is a finding aid, not a substitute for the applicable intelligence standard. https://en.wikipedia.org/wiki/Admiralty_code
- Bench-Capon, T. J. M. (2003). "Persuasion in Practical Argument Using Value-based Argumentation Frameworks." *Journal of Logic and Computation*, 13(3), 429–448. <https://doi.org/10.1093/logcom/13.3.429>
- Heuer, R. J., Jr. (1999). *Psychology of Intelligence Analysis*. CIA Center for the Study of Intelligence. See the Analysis of Competing Hypotheses chapter. <https://www.cia.gov/resources/csi/static/Psychology-of-Intelligence-Analysis.pdf>
- Huynh, T. D., Jennings, N. R., & Shadbolt, N. R. (2006). "An Integrated Trust and Reputation Model for Open Multi-Agent Systems." *Autonomous Agents and Multi-Agent Systems*, 13(2), 119–154. <https://doi.org/10.1007/s10458-005-6825-4>
- Jøsang, A. (2001). "A Logic for Uncertain Probabilities." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3), 279–311. <https://folk.universitetetioslo.no/josang/papers/Jos2001-IJUFKS.pdf>
- Jøsang, A., & Ismail, R. (2002). "The Beta Reputation System." *Proceedings of the 15th Bled Electronic Commerce Conference*. <https://folk.universitetetioslo.no/josang/papers/JI2002-Bled.pdf>

- Kent, S. (1964). "Words of Estimative Probability." Reprinted in *Sherman Kent and the Board of National Estimates: Collected Essays*, 1994. <https://www.cia.gov/resources/csi/static/Words-of-Estimative-Probability.pdf>
- Lee, J. (1990). "SIBYL: A Tool for Managing Group Design Rationale." *CSCW '90*, 79–92. <https://doi.org/10.1145/99332.99344>
- Lee, J. (1991). "Extending the Potts and Bruns Model for Recording Design Rationale." *ICSE '91*, 114–125. <https://doi.org/10.1109/ICSE.1991.130629>
- Masolo, C., Borgo, S., Gangemi, A., Guarino, N., & Oltramari, A. (2003). *Ontology Library*. WonderWeb Deliverable D18; includes DOLCE.
- Mellers, B. et al. (2014). "Psychological Strategies for Winning a Geopolitical Forecasting Tournament." *Psychological Science*, 25(5), 1106–1115. <https://doi.org/10.1177/0956797614524255>
- Moran, T. P., & Carroll, J. M. (eds.). (1996). *Design Rationale: Concepts, Techniques, and Use*. Lawrence Erlbaum Associates.
- Niles, I., & Pease, A. (2001). "Towards a Standard Upper Ontology." *Proceedings of FOIS '01*, 2–9; introduces SUMO.
- Office of the Director of National Intelligence. (2015). *Intelligence Community Directive 203: Analytic Standards*. Revised from the 2007 directive.
- Sabater, J., & Sierra, C. (2001). "REGRET: Reputation in Gregarious Societies." *Proceedings of Agents '01*, 194–195. <https://doi.org/10.1145/375735.376110>
- Sabater, J., & Sierra, C. (2005). "Review on Computational Trust and Reputation Models." *Artificial Intelligence Review*, 24(1), 33–60. <https://www.iiia.csic.es/~jsabater/Publications/2005-AIR.pdf>
- Schwenk, C. R. (1990). "Effects of Devil's Advocacy and Dialectical Inquiry on Decision Making: A Meta-analysis." *Organizational Behavior and Human Decision Processes*, 47(1), 161–176. <https://doi.org/10.1016/0749-5978%2890%2990051-A>
- Teacy, W. T. L., Patel, J., Jennings, N. R., & Luck, M. (2006). "TRAVOS: Trust and Reputation in the Context of Inaccurate Information Sources." *Autonomous Agents and Multi-Agent Systems*, 12(2), 183–198. <https://doi.org/10.1007/s10458-006-5952-x>
- Tetlock, P. E. (2005). *Expert Political Judgment: How Good Is It? How Can We Know?* Princeton University Press. <https://archive.org/details/expertpoliticalj0000tetl>
- Walton, D., Reed, C., & Macagno, F. (2008). *Argumentation Schemes*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511802034>
- Wang, R. Y., & Strong, D. M. (1996). "Beyond Accuracy: What Data Quality Means to Data Consumers." *Journal of Management Information Systems*, 12(4), 5–33. <https://doi.org/10.1080/07421222.1996.11518099>

Additional contemporary research and evaluations

- Becker, J., Rush, N., Barnes, B., & Rein, D. (2025). "Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity." *METR*. <https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/>
- Cemri, M. et al. (2025). "Why Do Multi-Agent LLM Systems Fail?" *MAST and MAST-Data*. Version 3, 26 October 2025. <https://arxiv.org/abs/2503.13657v3>
- Chan, C. et al. (2024; preprint 2023). "ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate." <https://arxiv.org/abs/2308.07201>
- Debenedetti, E. et al. (2024). "AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents." <https://arxiv.org/abs/2406.13352>

- Debenedetti, E. et al. (2025). “Defeating Prompt Injections by Design.” CaMeL. <https://arxiv.org/abs/2503.18813>
- Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., & Mordatch, I. (2024; preprint 2023). “Improving Factuality and Reasoning in Language Models through Multiagent Debate.” <https://arxiv.org/abs/2305.14325>
- Dylan, D. et al. (2026). “At Equal Inference Cost, Multi-Agent Structure Does Not Beat a Single Frozen Agent.” Preprint, version 1. <https://arxiv.org/abs/2609.04217v1>
- Edge, D. et al. (2024). “From Local to Global: A Graph RAG Approach to Query-Focused Summarization.” <https://arxiv.org/abs/2404.16130>
- Gao, M., Li, Y., Liu, B., Yu, Y., Wang, P., Lin, C.-Y., & Lai, F. (2025). “Single-agent or Multi-agent Systems? Why Not Both?” Preprint. <https://arxiv.org/abs/2505.18286v1>
- Hadfield, J., Zhang, B., Lien, K., Scholz, F., Fox, J., & Ford, D. (2025). “How We Built Our Multi-Agent Research System.” Anthropic, 13 June. Vendor engineering account. <https://www.anthropic.com/engineering/multi-agent-research-system>
- Hong, K., Troynikov, A., & Huber, J. (2025). “Context Rot: How Increasing Input Tokens Impacts LLM Performance.” Chroma Research, July 2025. <https://www.trychroma.com/research/context-rot>
- Jimenez, C. E. et al. (2024; preprint 2023). “SWE-bench: Can Language Models Resolve Real-World GitHub Issues?” <https://arxiv.org/abs/2310.06770>
- Kapoor, S., Stroebl, B., Siegel, Z. S., Nadgir, N., & Narayanan, A. (2024). “AI Agents That Matter.” <https://arxiv.org/abs/2407.01502v1>
- Kim, Y. et al. (2026; first preprint 2025). “Towards a Science of Scaling Agent Systems.” Preprint, version 3, 8 April 2026. <https://arxiv.org/abs/2512.08296v3>
- Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024; preprint 2023). “Lost in the Middle: How Language Models Use Long Contexts.” *TACL*, 12, 157–173. <https://arxiv.org/abs/2307.03172v3>
- Lynch, A. et al. (2025). “Agentic Misalignment: How LLMs Could Be an Insider Threat.” Anthropic Research; controlled simulated scenarios. <https://www.anthropic.com/research/agentic-misalignment>
- METR. (2026). “We Are Changing Our Developer Productivity Experiment Design.” 24 February 2026; discusses selection and measurement limitations. <https://metr.org/blog/2026-02-24-uplift-update/>
- Packer, C. et al. (2023). “MemGPT: Towards LLMs as Operating Systems.” <https://arxiv.org/abs/2310.08560>
- Panickssery, A., Bowman, S. R., & Feng, S. (2024). “LLM Evaluators Recognize and Favor Their Own Generations.” <https://arxiv.org/abs/2404.13076>
- Sharma, M. et al. (2023). “Towards Understanding Sycophancy in Language Models.” <https://arxiv.org/abs/2310.13548>
- Shi, J., & Zhang, T. (2026). “What Do CAE Simulation Agents Really Need Beyond a Generic Harness?” Preprint, version 1, 3 September. <https://arxiv.org/abs/2609.03718v1>
- Wang, J. et al. (2024). “Mixture-of-Agents Enhances Large Language Model Capabilities.” <https://arxiv.org/abs/2406.04692>
- Wang, X. et al. (2023). “Self-Consistency Improves Chain of Thought Reasoning in Language Models.” *ICLR*. <https://arxiv.org/abs/2203.11171v4>
- Wang, Z., Mao, S., Wu, W., Ge, T., Wei, F., & Ji, H. (2024). “Unleashing the Emergent Cognitive Synergy in Large Language Models: A Task-Solving Agent through Multi-Persona Self-Collaboration.” *NAACL*. <https://arxiv.org/abs/2307.05300v4>
- Xu, F. F. et al. (2025; preprint 2024). “TheAgent-Company: Benchmarking LLM Agents on Consequential Real World Tasks.” Revision of 10 September 2025. <https://arxiv.org/abs/2412.14161v3>

Yao, S. et al. (2024). “ τ -bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains.” <https://arxiv.org/abs/2406.12045>

Zhang, J. et al. (2026). “Rethinking the Evaluation of Efficiency Methods for Multi-Agent Systems.” Available preprint, version 1, 5 September. <https://arxiv.org/abs/2609.05933v1>

Standards, protocols, and regulation

Agent2Agent Project. (2026). *Agent2Agent Protocol Specification*, version 1.0.0. <https://a2a-protocol.org/v1.0.0/specification/>

European Parliament and Council. (2024). Regulation (EU) 2024/1689, *Artificial Intelligence Act*. <https://eur-lex.europa.eu/eli/reg/2024/1689/oj>

Model Context Protocol. (2025). *Architecture and Base Protocol Overview*, specification dated 18 June 2025. <https://modelcontextprotocol.io/specification/2025-06-18/architecture>

National Institute of Standards and Technology. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1. <https://doi.org/10.6028/NIST.AI.100-1>

National Institute of Standards and Technology. (2024). *Generative Artificial Intelligence Profile*, NIST AI 600-1. <https://doi.org/10.6028/NIST.AI.600-1>

Historical image credits

The photographs and reproduced research figure below accompany the concepts they help explain. These credits identify their sources, reuse terms, and any modifications. They are used for editorial and educational purposes; no photographer, institution, or depicted person endorses this publication.

H1 – Mars Climate Orbiter during tests. NASA, 27 May 1998, GPN-2000-000498 (alternate ID M98ORBITER). The item describes acoustic tests simulating launch conditions. Public domain in the United States (PD-USGov-NASA in the item record); NASA's media guidelines allow factual educational and editorial use without implied endorsement. The downloaded JPEG is unchanged; only display size is adjusted. [Item record and rights declaration](#); [NASA media-use guidelines](#). NASA is the source of the photograph, not the author or reviewer of this manuscript's interpretations.

H2 – TVA fertilizer test field. *Results of Fertilizer*, 1942. Franklin D. Roosevelt Presidential Library and Museum, record 53227(1828), image 27-0921a. Individual photographer not identified in the inspected record. Public domain in the United States as a federal-government work; the item also carries a Public Domain Mark. This is the 590-by-471-pixel Commons version, whose uploader cropped the archival image in 2011. No further cropping, retouching, or upscaling was applied to the file. [Item record, version history, and rights declaration](#).

H3 – Mine-to-plant coal transport. Lyntha Scott Eiler, *Trucks Haul Coal from the Navajo Mine to the Four Corners Generating Plant*, May 1972. EPA/DOCUMERICA, U.S. National Archives, NAID 544169. Public domain in the United States as a federal-government work (PD-USGov-EPA in the item record). The archival JPEG is reproduced unchanged, including its slide border; display size only is adjusted. [Item record and rights declaration](#).

H4 – Apollo 11 launch-control test. NASA, KSC-69P-3253, 1969. NASA's media-use page identifies the scene as engineers monitoring an overall vehicle test in Firing Room 1. Used under NASA's permission for factual educational and editorial publication, with acknowledgment and no implied endorsement. No cropping or retouching; display size only is adjusted. [Caption and media-use terms](#); [NASA image](#).

R1 – TheAgentCompany environment. Frank F. Xu and coauthors, *TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks*, arXiv:2412.14161v3, 10 September 2025, Figure 1. Reproduced unchanged under [CC BY 4.0](#), the license linked by the [versioned article record](#). [Original figure and caption](#). Product marks remain part of the attributed research figure; their inclusion does not imply endorsement or grant trademark rights.

H5 – TVA central files. TVA Web Team, 1936 - *Central files*, K-0897, via its Flickr collection and Wikimedia Commons. [CC BY 2.0](#), verified by the Commons license review dated 25 September 2013. [Item and rights record](#). The historical year comes from the title; the 2005 camera timestamp concerns the digital reproduction. Individual historical photographer not identified. The downloaded JPEG is unchanged; display size only is adjusted.

Portrait credits

Herbert A. Simon. Portrait from *News and Events*, Rochester Institute of Technology, issue of 19 March 1981. The file record identifies a United States public-domain basis for publication without notice; the newsletter date is not asserted to be the exposure date. Reproduced from the existing Commons extraction without further cropping or retouching. [Source and rights record](#).

Tony Hoare. Rama, Lausanne, 20 June 2011, *Sir Tony Hoare IMG 5125*. Reproduced unchanged under [CC BY-SA 2.0 France](#); display size only is adjusted. [Source record](#).

Elinor Ostrom. Holger Motzkau, Stockholm press conference, 7 December 2009, *Nobel Prize 2009-Press Conference KVA-30*. Reproduced unchanged under [CC BY-SA 3.0](#), display size only adjusted. [Source record](#).

Robin Milner. The [official portrait gallery](#) credits the University of Cambridge and specifies reproduction by permission. No permission for this publication was verified, so that portrait is linked rather than reproduced. The omission does not diminish his biographical treatment.

Information-box icons. Lucide contributors, Lucide 0.468.0, used under the ISC license with the upstream Lucide/Feather notices retained in the distributed [license file](#). [Project and license](#). Icons identify reading purposes, not evidence strength or institutional authority.

Core points and key concepts

The chapter takeaways, gathered in reading order. Each entry repeats its original wording; page references lead to the full discussion. The [glossary and index](#) provides alphabetical lookup.

1. The field and its boundaries

Institutions, not agent headcount. An artificial organization adds durable responsibilities, legitimate action, and accountable closure to computational agency. More agents do not, by themselves, supply any of these.

Core point
p. 8

2. Cases and examples used in this book

Choose the example for the question. Recurring incidents connect the chapters; organizational practices show how work is arranged; research studies and source specimens expose particular mechanisms. Each supplies a different kind of evidence. No single case demonstrates everything an artificial organization needs.

Core point
p. 13

3. A map of the field

Separate the layers, then connect them. Purpose explains why work exists; roles allocate responsibility; governance constrains action; runtime systems execute it; evidence tests whether the outcome occurred. These connections need not correspond to separate products.

Core point
p. 18

4. Why organize at all?

Organization must repay its overhead. Specialization and exception routing can save expertise, but handoffs add delay, information loss, and failure boundaries. Compare against a competent single worker or a deterministic workflow before adding management.

Core point
p. 23

Bounded rationality. A decision maker cannot examine every alternative, consequence, and piece of evidence. Simon makes those limits part of the explanation of choice, rather than treating them as an incidental defect [Simon 1947]. In an artificial organization, assess the decision process against its actual information, capability, time, and resource constraints.

Key concept
p. 24

Satisficing. Search can stop when an option meets an aspiration or acceptability threshold, rather than after proving it globally optimal [Simon 1955]. This is not permission to accept arbitrary quality. For an engineered workflow, make the threshold and stopping condition explicit; whether the threshold is good enough remains a separate design question.	Key concept p. 24
Near-decomposability. Simon's important distinction is stronger interaction within subsystems and weaker interaction between them , not complete independence [Simon 1962]. It explains why some complex systems can be understood in parts. A task boundary is useful only if it respects the interactions that still cross it.	Key concept p. 25
Factorization. Express a problem through smaller components while retaining the interactions needed for a correct combined result [Dechter 1999; Guestrin et al. 2001]. Independent components can be optimized separately; coupled components require coordination. The saving comes from exploiting structure, which either a central program or several agents may do.	Key concept p. 26
Uncertainty absorption. Recipients often receive an inference instead of the observations that produced it [March and Simon 1958]. This economizes on attention but can turn tentative judgments into apparently settled facts. Preserve the assumptions and contrary evidence needed to reopen the inference; forwarding everything is not the only alternative.	Key concept p. 28
Information-processing need versus capacity. Galbraith's two responses are to reduce the coordination information a task requires or increase the organization's capacity to process it [Galbraith 1973]. Better task boundaries and slack address the first; information systems and lateral relationships address the second. More reporting does not necessarily do either.	Key concept p. 28
Task interdependence. Pooled work shares an enterprise or resource base; sequential work consumes upstream outputs; reciprocal work repeatedly changes other work's inputs [Thompson 1967]. The distinction helps choose standardization, planning, or mutual adjustment. A reciprocal task does not become a pipeline merely because the interface displays it in columns.	Key concept p. 29
Coordination mechanisms. Standardizing a process, an output, or a skill provides different assurance; direct supervision and mutual adjustment coordinate in other ways [Mintzberg 1979]. Specify what is actually standardized or checked. A role title cannot substitute for demonstrated skill, and procedural compliance cannot establish that an outcome was achieved.	Key concept p. 29
Distributed agency. Several actors retain distinct local state, capabilities or decision rights and coordinate their contributions [Stone and Veloso 2000]. The reason for separation should be identifiable: capacity, local knowledge, independent interests or bounded authority. Several roles may share one model; several conversations may still belong to one principal. Model count and legitimate independence are different things.	Key concept p. 32

Knowledge hierarchies. Route routine problems to general workers and exceptional problems to scarce expertise [Garicano 2000]. The rationale is conserving specialized knowledge, not assuming that a superior position always knows more. The advantage depends on the task distribution and the costs of learning and communicating.

Key concept
p. 34

Screening architectures. Requiring every screen to approve and allowing any channel to approve trade false acceptance against false rejection in different ways [Sah and Stiglitz 1986]. The preferred rule depends on losses and error dependence. Adding reviewers is not a neutral increase in safety; it changes which mistakes the organization tends to make.

Key concept
p. 34

Capacity comes with organizational commitments. The factory panel connects staffing, production and inventory in one plan because a local saving can create costs elsewhere. The TVA panel connects local reach with the partners' influence over which farmers and interests the programme serves. An intermediary supplies capacity and can shape its use. These studies are not interchangeable evidence for one ideal hierarchy [Holt et al. 1960; Selznick 1949].

Key concept
p. 40

Loss-accounted compression. A useful summary reduces reading effort while preserving a route to its evidence, omissions, and unresolved disagreement. This book's design heuristic asks what a summary loses and how that loss can be inspected; it is not a named theorem from Simon or a claim that every detail must remain in the executive view.

Key concept
p. 41

Transaction costs and firm boundaries. Compare the cost of obtaining the same useful result through different arrangements, not the supplier's price against a supposedly free internal instruction. Internal organization saves some search, bargaining, and adaptation costs but adds administration, monitoring, decision errors, and incentive problems. Coase's boundary question is marginal: would the next activity be coordinated more economically inside this firm, through the market, or by another firm? There is no implication that all activity belongs in one large hierarchy.

Key concept
p. 42

5. Roles, missions, and obligations

Roles outlive workers. Bind duties to durable positions and missions, then record their occupants and enactment intervals. Replacement is safe only when responsibility, access, and unfinished work transfer explicitly.

Core point
p. 49

Role, occupant, and duty. A role defines an institutional position; enactment records who occupies it; a deontic relation binds a role to a mission [Ferber and Gutknecht 1998; Hübner et al. 2002b]. Changing an occupant need not change the duty. Model those changes separately so replacing a worker cannot silently erase responsibility.

Key concept
p. 51

Landmarks. Specify a state that must be reached without prescribing every intermediate action [Dignum 2004]. This permits different participants and plans to satisfy the same organizational requirement. The flexibility concerns the route; it does not waive permissions, evidence requirements, or constraints on how that state may be reached.

Key concept
p. 52

Dimensions versus refinement levels. OMNI separates organizational, normative, and ontological concerns from abstract, concrete, and implementation-level descriptions [Dignum et al. 2005]. A detailed schema can still leave a policy question unanswered. Check both what concern a statement addresses and how far it has been made operational.

Key concept
p. 54

6. Coordination and organizational topology

Topology follows dependency. Pipelines move artifacts; hierarchies route assignments and summaries; blackboards coordinate through shared state; contracts allocate work; deliberation exchanges reasons. Choose the dependency mechanism before choosing the number of agents.

Core point
p. 59

Blackboard coordination. Specialists contribute to an explicit shared problem representation; a control policy selects useful next contributions [Erman et al. 1980]. The board is neither another manager nor an unstructured chat history. Its advantage is opportunistic coordination; scheduling and shared-state consistency remain real work.

Key concept
p. 62

Blackboard is not tuple space. A blackboard architecture organizes problem-solving around shared partial solutions and a selection policy for contributions. Linda supplies associative publication, observation, and consumption operations [Gelernter 1985; Carriero et al. 1994]. A tuple space can support part of a blackboard, but it does not automatically supply its scheduler, task semantics, or independent acceptance criteria.

Key concept
p. 67

Atomic claim is not reliable work. One consuming operation can exclude a second taker of the same tuple. A stopped consumer can still leave the work unfinished, and two duplicate publications can admit two consumers. Keep publication identity, durable attempt ownership, result submission, and outcome acceptance distinct. Linda's small operation set is useful precisely because its scope can be stated clearly.

Key concept
p. 70

Business process. A coordinated set of activities, events, decisions and interactions directed toward an organizational outcome. Its model describes possible behavior; a process instance is one particular unfolding case. A workflow makes aspects of the routing and execution rules explicit enough to support or automate them. Neither the model nor its successful execution proves that the intended business outcome occurred [van der Aalst et al. 2003; OMG 2014].

Key concept
p. 76

Orchestration and choreography. Orchestration describes the control of work from one process's perspective. Choreography describes the exchanges expected among participants without supplying all their private implementations. A choreography can constrain autonomous parties, but a diagram of their exchanges does not guarantee compatible implementations, delivery, authorization or eventual completion [OMG 2014; Chopra et al. 2020].

Key concept
p. 77

Workflow soundness. In its classic workflow-net setting, completion remains possible from every reachable state, reaching the final state leaves no residual work, and no modeled transition is permanently unusable. This is a property of the modeled behavior, not a guarantee of deadlines, factual correctness, legal compliance or business value. An indefinitely repeated permitted loop can still require a termination policy in the implementation [van der Aalst 1998; van der Aalst et al. 2009].

Key concept
p. 78

Declarative process. A process specified through constraints on acceptable behavior rather than a single complete procedure. The next step can be chosen at runtime while the constraints remain in force. Underspecifying the constraints can admit unwanted behavior; specifying incompatible ones can leave no acceptable continuation. Flexibility therefore shifts the modeling burden rather than removing it [van der Aalst et al. 2009].

Key concept
p. 78

7. Delegation, authority, and human attention

Delegate rights, not just tasks. State what the worker may observe, analyze, select, and execute, with scope, expiry, and a no-response default. Escalation quality depends on consequences and evidence, not on a confidence phrase or an alert rule's apparent accuracy.

Core point
p. 83

Stage-specific automation. Acquiring information, analyzing it, selecting a decision, and implementing an action can have different allocations of human and machine control [Parasuraman et al. 2000]. Permission to analyze is not permission to execute. A single autonomy score hides this distinction precisely where consequences become real.

Key concept
p. 84

Access, power, permission. Technical access answers whether an interface can be reached. Institutional power answers whether the act can create a recognized effect. Permission answers whether the act is allowed in this case [Jones and Sergot 1996]. A technically possible or institutionally effective act may still violate a rule.

Key concept
p. 84

Lease expiry needs action-side enforcement. An expired lease does not stop a paused or disconnected participant from resuming. Chubby's sequencers carry lock identity, mode, and generation to the receiving service, which must validate them [Burrows 2006, Section 2.4]. For agentic work, require the effect-owning boundary to reject obsolete authority. A generation number in a log, or a check followed by an unguarded remote call, is not fencing.

Key concept
p. 85

Expected-loss communication. The reason to notify someone is the loss that informing them can prevent, compared with the cost of doing so [Tambe 1997]. Low confidence alone does not establish that a human should be interrupted. Consequence, recipient knowledge, and the opportunity to act must enter the judgment.

Key concept
p. 85

Positive predictive value. The fraction of alerts that are real depends on prevalence as well as sensitivity and specificity. For rare events, false positives can dominate a seemingly accurate detector's queue [Bliss et al. 1995; Meyer 2001]. Estimate the actual review burden, not just the detector's headline accuracy.

Key concept
p. 85

Appropriate reliance. Responding to an alarm and trusting silence are different behaviors [Meyer 2004]. The aim is reliance suited to the system's capabilities and context, not maximal trust [Lee and See 2004]. Evaluate actions on false alarms and harmful misses during silence separately.

Key concept
p. 90

Neglect tolerance and interaction time. Supervisory capacity depends on how long work can proceed without attention and how much attention each intervention requires [Crandall et al. 2005]. Simultaneous exceptions and context switching can invalidate a simple average. More parallel workers do not create more parallel human attention.

Key concept
p. 90

8. Norms, policy, and institutional facts

Permission, power, and evidence are different gates. A rule may forbid an action, define an official act, or prescribe repair after a violation. Decide which mechanism is needed and which boundary enforces it.

Core point
p. 96

Constitutive and regulative rules. A counts-as mapping explains when an observed act acquires an institutional meaning; a regulative rule concerns what is permitted, prohibited, or required. Recognizing an act is not the same as approving it. The institutional-power and counts-as sources discussed here keep those questions distinct.

Key concept
p. 99

Regimentation versus enforcement. Preventing a prohibited transition and detecting or responding to a violation are different controls. Prevention cannot undo an external act that already occurred; a violation record does not itself provide prevention. Select controls according to which actions the institution can actually constrain and observe.

Key concept
p. 101

Directed commitments. A commitment relates a debtor, a creditor, and content, potentially under a condition [Singh 1998]. A task message alone does not specify how that relationship is created, discharged, released, or violated. Preserve the parties and lifecycle independently of the conversation that led to the duty.

Key concept
p. 103

9. Reliability, evidence, and accountable closure

Verify results, not confident performance. Preserve source and action lineage, check acceptance independently of the worker, and observe the outcome over an appropriate horizon. A good rationale replaces none of these.

Core point
p. 108

Submission, acceptance, outcome. Submission records what the worker offers; acceptance records satisfaction of a review criterion; an outcome observation records what later happened. These are the monograph's distinct operational states. Conflating them lets a confident completion message masquerade as independent verification or business success.

Key concept
p. 110

Provenance is not proof. PROV represents entities, activities, agents, and their relationships [W3C 2013]. It can establish the recorded derivation of a claim without establishing that the claim is true, its inference is warranted, or its producing act was permitted. Keep lineage and evidential evaluation connected but distinct.

Key concept
p. 112

Warrants, qualifiers, and rebuttals. Evidence does not connect itself to a conclusion. Toulmin's warrant states the inference, backing supports it, the qualifier bounds the claim, and the rebuttal marks defeating conditions [Toulmin 1958]. Preserving only the claim and source loses precisely what a reviewer needs to challenge the reasoning.

Key concept
p. 113

Ignorance is not disbelief. In subjective logic, uncertain evidence and evidence against a proposition occupy different components [Jøsang 2001, 2016]. A supplier that has not been checked is not a supplier found invalid. The distinction matters even when the implementation uses explicit states rather than a numerical uncertainty calculus.

Key concept
p. 114

Task-specific reputation. A reliability judgment must state what activity and evidence it concerns [Sabater and Sierra 2001, 2005]. Accurate extraction does not establish sound legal judgment, and competence does not establish permission. A global trust score can conceal the very distinction needed to delegate safely.

Key concept
p. 115

Analysis of Competing Hypotheses. Compare plausible alternatives against evidence, especially evidence that distinguishes them [Heuer 1999]. Evidence consistent with every hypothesis provides little discrimination. Diagnostic inconsistencies deserve attention, but counting them without assessing quality and dependence is not a substitute for analysis.

Key concept
p. 117

10. Deliberation, dissent, and decisions

Preserve reasons that could change the decision. Deliberation earns its cost when independent evidence or competing values matter. End with a decision record, an owner, and authorized follow-through. Agreement among similar agents is not independent corroboration.

Core point
p. 123

Authentic dissent. A person or agent assigned to disagree does not necessarily provide the same challenge as a genuinely held contrary position. The human evidence reviewed here distinguishes those conditions. For an artificial organization, preserve the contrary evidence and test the resulting decisions; a role called "critic" is not evidence of independence.

Key concept
p. 124

Acceptance depends on semantics. An abstract argument framework specifies arguments and attacks; the semantics determines acceptable sets [Dung 1995]. Skeptical and credulous inference ask about all or at least one relevant extension. Neither formal acceptance nor a majority of agents establishes the factual truth of a premise.

Key concept
p. 127

Value-based argumentation. Defeat can depend on the audience's ordering of values [Bench-Capon 2003]. Two positions may differ because of priorities rather than evidence. Show that dependency explicitly; a formal calculation does not confer authority to choose the organization's values or override its non-negotiable constraints.

Key concept
p. 130

Agreement is not independence. Sycophancy, evaluator self-preference, and shared errors are different routes to misleading agreement [Sharma et al. 2023; Panickssery et al. 2024]. Different role names or model families do not prove independent judgment. Test error dependence and compare with strong single-worker and objective-check baselines.

Key concept
p. 131

11. Organizational memory and learning

Retrieve what is valid before what is similar. Policy, precedent, episodes, procedures, and transient context need different rules. Deliberate improvement needs evidence and review, not just a plausible retrospective story; explicit predictions make changes easier to evaluate.

Core point
p. 134

Validity before similarity. A relevant-looking memory can contain a superseded policy or an inapplicable precedent. This monograph's recordkeeping rule is to establish authority, scope, and effective time before using similarity to select among eligible records. A retrieval score is not a certificate that a rule governed the act in question.

Key concept
p. 134

Organizational retention and transactive memory. Knowledge persists in people, routines, structures, and other retention locations, not only documents [Walsh and Ungson 1991]. Transactive memory concerns knowing who knows what [Wegner 1987]. A searchable archive and an expertise directory support different retrieval questions; neither replaces the other.

Key concept
p. 136

Monotonic evidence, revisable decisions. Adding an immutable observation need not invalidate the fact that an earlier observation was submitted. Declaring a policy current, a budget available, or a claim finally accepted can depend on facts not yet received. The CALM principle (consistency as logical monotonicity) connects monotonic program semantics with coordination-free distributed consistency under its assumptions [Hellerstein and Alvaro 2019]. It does not establish that every decision can avoid agreement or that communication becomes unnecessary.

Key concept
p. 139

Design rationale. Preserve the question, options, criteria, and reasons so a later decision can reconsider the tradeoff rather than imitate its outcome [MacLean et al. 1991]. A past success is not sufficient evidence that the same choice applies under different constraints.

Key concept
p. 140

12. Runtime architecture and protocols

Put institutional checks around the runtime. Protocols transport requests and artifacts; they do not decide who may commit the organization. Bind execution infrastructure to policy, authority, provenance, and acceptance.

Core point
p. 144

Interoperability is not authority. A protocol can make a request intelligible without making it authorized. Tool names, message IDs, and task states support exchange; institutional scope, decision rights, and evidence requirements must be enforced at the relevant action boundary. Inspect those layers separately when evaluating a runtime.

Key concept
p. 144

Acting, reflecting, and organizing are different mechanisms. ReAct couples reasoning with actions and observations; Reflexion retains linguistic feedback across attempts [Yao et al. 2023; Shinn et al. 2023]. Neither mechanism alone supplies institutional authority or independent acceptance. Inspect the mechanism beneath a team's role names.

Key concept
p. 152

Coordinate the invariant, not every observation. Concurrent evidence collection can often be decoupled, while exclusive ownership, revocation, and scarce-resource admission need agreement or correctly allocated rights [Bailis et al. 2014]. Choose the smallest coordination boundary that preserves the actual requirement. Do not build a bespoke agent queue when an established transactional or workflow substrate provides the required contract.

Key concept
p. 163

13. A construction method

Build one governed outcome end to end. Define acceptance and authority before dispatch, test failure and recovery, and only then expand. A collection of capable agents is not a substitute for one demonstrably correct organizational transaction.

Core point
p. 166

14. Putting the concepts to work

Connect the controls to the outcome. Roles, coordination, policy, evidence, memory, and review are useful together, at a particular boundary of work. An intelligible handoff is not necessarily authorized; a completed action is not necessarily an accepted result; an accepted result does not erase every downstream loss.

Core point
p. 172

Test the distinctions under pressure. Credentials are not permission; consensus is not independence; recency is not validity; a completed attempt is not an achieved outcome. An evaluation should expose the missing boundary and its consequences, not merely recognize the terminology.

Key concept
p. 177

15. What LLM-agent systems should inherit

Better workers do not replace organizational mechanisms. Decades of MAS research made role enactment, commitments, coordination, and institutional rules explicit. LLM systems make new kinds of work practical, but many familiar organizational labels still need those mechanisms beneath them. Recover the useful contract, not necessarily the old implementation.

Core point
p. 182

16. What remains open

Design around uncertainty, not promises. The reviewed literature does not establish reliable open-ended company autonomy, a universal supervision capacity, or one best memory architecture. Instrument the limits locally and state what evidence would change your conclusion.

Core point
p. 191

17. Synthesis

Coherent action needs structure and correction. Keep legitimate action inspectable, preserve evidence that summaries omit, invite correction at consequential decisions, and expand autonomy only where representative local results justify it.

Core point
p. 195

Glossary and index

Definitions follow this monograph's usage. References lead to the principal explanations and selected applications, not every incidental mention. The PDF shows page numbers; the browser edition shows clickable section references. Subentries locate related distinctions without repeating the main definition.

A

A2A. Agent2Agent protocol for exchanging messages, tasks, and artifacts between independently implemented agents. 144, 154.

Task completion versus accepted obligation: 184.

Abstract argumentation. Representation of arguments and attack relations assessed under explicit acceptance semantics. 127.

Graph specimen: 128.

Acceptance. An attributable judgment that a submitted result meets a defined criterion. 110.

Independent evidence: 112.

Observed outcome: 176.

Actors. Addressed computational participants with local state and message handling. 71.

Agent. A computational actor that observes, selects actions, and acts toward goals. 8.

AGR. Agent–Group–Role model separating participants, groups, and enacted positions. 49, 53.

Air Canada. Public chatbot-advice dispute used to examine responsibility and inconsistent policy representation. 13.

Institutional consequences: 101.

Argument and evidence: 125.

Alert fatigue. Degraded attention or response under burdensome or unreliable warning conditions. 85.

Argument. A claim with reasons that can be supported, challenged, and assessed under a decision procedure. 123.

Attacks and defense: 127.

Qualifiers and rebuttals: 112.

Artificial organization. Agents embedded in durable purpose, roles, duties, authority, coordination, and accountability. 8.

Term and research traditions: 19.

Asset specificity. Investment whose value depends on a particular relationship. 42.

Attempt. One execution undertaken toward a work item, distinct from its obligation and accepted result. 110.

Authority. Governed rights to make decisions or take actions within specified boundaries. 84.

Acquisition, analysis, selection, execution: 84.

Access, power, permission: 96.

Authority lease. A bounded, time-sensitive delegation of action rights. 83, 84.

Autonomy. Scope of independent action allocated by task and stage, not a single global level. 84.

B

Benchmark. A defined evaluation setting with tasks, conditions, and an evaluator. 117.

Blackboard. Shared problem state with a policy for selecting contributions. 61.

Difference from tuple space: 69.

Bounded rationality. Decision making constrained by information, attention, time, and computational capacity. 23.

Brier score. Squared-error score for a probability forecast and resolved outcome. 114.

Business process. Activities, decisions and interactions directed toward an organizational outcome, often crossing departmental boundaries. 75.

Autonomous participants: 78.

C

Calibration. Agreement between stated probabilities and observed frequencies across comparable forecasts. 114.

Discrimination and baselines: 132.

CALM. Connection between logical monotonicity and coordination-free consistency under stated conditions. 137.

Channels. Explicit communication connections with specified synchronization or buffering behavior. 71.

Choreography. Expected exchanges among participants, without specifying all their private implementations. 75.

Chubby. Coordination service supplying small reliable shared state and ownership mechanisms. 162.

Commitment. A directed undertaking with parties, content, and lifecycle. 103.

Handoff versus acceptance: 184.

Violation, release, and repair: 103.

Compensation. A further action intended to address consequences that cannot simply be rolled back. 150, 162.

Constitutive rule. A rule specifying what an event counts as within an institution. 99.

Contract Net. Allocation through announcement, proposals, selection, and award. 59.

Cooptation. Incorporation of external interests that secures cooperation while creating institutional commitments and influence. 34.

Coordination. Management of dependencies among activities, resources, information, and outcomes. 59.

Work arrangements: 61.

Computational mechanisms: 71.

Coordination graph. A factorization of a shared payoff into terms involving subsets of decisions; not a guarantee that the factors can be optimized independently. 25.

Correlated errors. Failures whose dependence weakens the value of apparent independent agreement. 131.

Counts-as. Recognition of an observed act as an institutional fact under applicable rules. 99.

CRDT. Replicated data type designed to converge under specified merge or update conditions. 137.

CSP. Communicating Sequential Processes; a formal approach to synchronized interaction. 71.

Hoare's COPY process: 72.

D

Decision record. Attributable choice with reasons, alternatives, dissent, and conditions for reopening. 125, 132.

Declarative process. Constraints delimit acceptable behavior while leaving choices to runtime. 75.

Dec-POMDP. A cooperative decision model in which actors choose from local observation histories; centralized policy design need not imply centralized execution. 44.

Delegation. Transfer of bounded responsibility and authority with conditions for review and recovery. 83.

Dissent. A challenge supplying reasons or evidence that can affect the decision. 124.

Human evidence: 126.

Model diversity: 131.

Durable workflow. Execution whose progress and recovery state survive interruptions under a defined contract. 144.

E

Enforcement. Monitoring and consequential treatment of violations that may occur. 96.

Escalation. Routing an exception to someone with relevant knowledge or authority. 85, 91.

Evidence. Retained observations and sources used to assess a claim. 108.

Provenance and rationale: 112.

Source quality: 116.

F

Factorization. Representation of an objective or model through smaller interacting components. Independence can permit separate optimization; coupling and intermediate factor width determine the remaining cost. 25.

Fan-out. Number of concurrent workers an overseer can effectively supervise under measured conditions. 90.

Fencing. Receiver-enforced rejection of actions from superseded owners or generations. 84, 162.

FIPA-ACL. Agent communication specifications defining communicative acts and message structure. 154.

G

Gaia. Agent-oriented methodology emphasizing roles, responsibilities, and interactions. 53.

GitLab recovery. Documented outage used to distinguish recoverability, restored service, and recovered data. 174.

Dependencies: 147.

Notification failure: 174.

Grounded extension. The least fixed-point set accepted under grounded argumentation semantics. 127.

H

Handoff. Transfer of control or information; not necessarily an accepted duty. 184.

Hierarchy. Arrangement concentrating assignment, integration, or exception routing through higher-level positions. 25, 59.

Hoare, Tony. Pioneer of program reasoning and Communicating Sequential Processes. 71.

Hold-up. Exploitation of relationship-specific dependence after investment. 42.

I

IBIS. Issue-Based Information Systems; issues, positions, and arguments made explicit. 123.

Idempotency. Repetition of an operation without repeating its intended effect, under a defined contract. 150.

Induced width. Largest number of remaining neighbors during variable elimination under a specified ordering; governs exponential intermediate costs in the tabular methods discussed here. 25.

InstAL. Institutional Action Language distinguishing observed events, institutional events, and normative state. 99.

Institutional power. Capacity for an act to create a recognized institutional effect. 96.

Interruption. Transfer of a person's attention through an immediate, negotiated, scheduled, or mediated channel. 87.

Invariant. A condition that permitted transitions must preserve. 71, 162.

J

JavaSpaces. Shared-entry coordination with separate lifetime, transaction, and waiting parameters. 163.

Joint intentions. Teamwork involving maintained shared commitment and communication about its status. 59, 185.

K

Knight Capital. Trading incident illustrating the difference between execution and effective risk controls. 13, 174.

KQML. Agent communication language making communicative acts, language, and ontology explicit. 154.

L

Landmark. Required state in an interaction, without necessarily fixing one complete execution script. 51.

Linda. Generative communication through associative tuple-space operations. 69.

Loss-accounted compression. This book's heuristic for retaining evidence, omissions, and objections behind a summary. 40.

M

Mars Climate Orbiter. Spacecraft-loss case illustrating failed cross-team interpretation and checking. 40.

MCP. Model Context Protocol; host-client-server exchange of tools and context. 144, 145.

Memory, organizational. Retention and use of knowledge across people, routines, structures, and records. 134.

Validity and recording time: 135.

Retention locations: 136.

Rationale reuse: 140.

Milner, Robin. Pioneer of proof tools, programming languages, and process calculi. 71.

Mission. A bundle of goals to which responsibility can be assigned. 51.

MOISE+. Organizational specification separating structure, goals and missions, and normative bindings. 51, 53.

Monotonicity. Accumulation that does not invalidate previously derived conclusions under the chosen logic. 137.

N

Near-decomposability. Relative separation of interaction strengths and timescales across parts of a system. 25, 25.

Norm. An institutional requirement or rule, with applicability and consequences needing explicit interpretation. 96.

O

OMNI. Framework coordinating organizational, normative, and ontological dimensions across refinement levels. 53.

OperA. Organizational-interaction model distinguishing requirements, participation, and interaction arrangements. 51.

Orchestrator. Component that decomposes work, directs execution, and integrates outputs. 61.

Ostrom, Elinor. Scholar of institutional diversity and collective governance of shared resources. 45.

Outcome. Observed effect in the world, including adverse or inconclusive results. 110.

P

Permission. Whether an act is allowed in its current context; distinct from technical access and power. 96.

Pi-calculus (π -calculus). Process calculus in which communicating names can change subsequent connectivity. 71, 73.

Policy lifecycle. Creation, revision, applicability, conflict resolution, retirement, and audit of rules. 97.

Positive predictive value (PPV). Fraction of raised alerts corresponding to actual target events. 85.

Process mining. Discovery, conformance checking and enhancement using event records; log-model agreement is not outcome verification. 80.

Provenance. Structured lineage connecting entities, activities, actors, and derivations. 112, 113.

Q

QOC. Questions–Options–Criteria; a representation of alternatives and explicit evaluative dimensions. 123.

R

Rationale. Reasons offered for a decision, distinct from a mechanically captured action trace. 140.

Regimentation. Prevention of prohibited transitions or recognition within the controlled institutional infrastructure. 96.

Reo. Exogenous coordination by composition of channels and connectors. 71, 74.

Residual control rights. Rights to determine asset use where a contract leaves decisions unspecified. 42.

Rinda. Ruby tuple-space library used for the retained local coordination checks. 70.

Role. Durable institutional position with duties and relationships, distinct from its occupant. 49.

Replacement: 51.

Rights and compatibility: 51.

S

Saga. A sequence of steps with compensating actions for effects that cannot be rolled back atomically. 150.

Satisficing. Searching until a sufficiently acceptable option is found under an aspiration criterion. 23.

Screening architecture. Arrangement of acceptance channels trading false admission against false rejection. 34.

Silence. Absence of a signal whose interpretation depends on monitoring state and evidence freshness. 109, 176.

Simon, Herbert A. Scholar connecting bounded decision making, organizations, and artificial intelligence. 23.

Situation awareness. Perception, comprehension, and projection of relevant conditions for action. 88.

STEAM. Teamwork framework addressing maintained commitments and selective communication. 85.

Subjective logic. Formalism separating belief, disbelief, uncertainty, and base rate. 114.

Sycophancy. Agreement with a user's beliefs that can undermine independent assessment. 131.

T

Toulmin model. Claims connected to data, warrants, backing, qualifiers, and rebuttals. 112.

Transaction costs. Costs of arranging, monitoring, adapting, and enforcing exchanges. 42.

Transactive memory. A group's knowledge of who knows what and how to access that expertise. 136.

Tropos. Agent-oriented methodology carrying stakeholder goals and dependencies through design. 53.

Trust. Task-specific reliance requiring relevant evidence and an account of uncertainty. 115.

Tuple space. Shared records published and selected by matching their contents. 69.

Read versus consume: 70.

Claim versus completion: 162.

TVA. Tennessee Valley Authority; Selznick's study illustrates institutional cooperation and influence. 34.

U

Uncertainty absorption. Conclusions travel while the observations behind them recede from view. 23.

V

Validity time. When a rule applies, distinguished from when a system records or learns it. 135.

Verification. Checking a specified claim against suitable independent evidence. 112.

W

Workflow. Explicit routing and execution rules supporting a process; these can leave choices to run-time rather than prescribe one fixed path. 75, 147.

Workflow soundness. A modeled execution property concerning possible completion, proper completion and absence of dead transitions; not a guarantee of business value. 75.

Z

ZooKeeper. Replicated coordination service whose primitives support ownership, membership, and synchronization recipes. 162.