Artificial Intelligence for Law and Finance

A Comprehensive Textbook for Practitioners and Researchers

Draft Version

November 1, 2025

Working Draft Textbook

Version 0.1 — In Active Development

This textbook is currently under active development. Individual chapters are being drafted, reviewed, and refined independently before integration into the complete book. Content, structure, and formatting are subject to change as the project evolves.

This is a collaborative scholarly work aimed at legal and financial practitioners, regulators, and researchers. We welcome feedback and contributions.

You can find the most current version of this project here:

https://github.com/mjbommar/ai-law-finance-book/

Contents

Part I

Agents and Agentic Systems

Working Draft Chapter

Version 0.1

This chapter is Part I of a three-part series from the textbook *Artificial Intelligence for Law and Finance*, currently under development. Individual chapters are being drafted and refined independently before integration into the complete book. Part II: *How to Build an Agent* and Part III: *How to Govern an Agent* are forthcoming.

You can find the most current copy of the textbook project here:

https://github.com/mjbommar/ai-law-finance-book/

Contents

How to Read This Chapter

This chapter answers a deceptively simple question: *What is an agent?*

The term "agent" appears everywhere in industry and academia today — from "AI agents" to "agentic workflows" — yet rarely with clear definition. This creates confusion, miscommunication, and

inevitably, disappointment.

We hope to help you avoid this disappointment, but **this chapter is not short and your time is valuable**. You don't need to read everything, especially on first reading.

If you're in a hurry, consider which of the three reading paths below best fits your goals.

Three Reading Paths:

Path 1: Definitions and Concepts

Read Sections 0.1–0.2 only. You'll get:

- The definition and three-level hierarchy

- A practical decision rubric with 10 questions

- Common misconceptions clarified

- Enough to use the term correctly and evaluate claims

Stop after Section 0.2—you'll have what you need.

Path 2: Historical and Disciplinary Context

Read Sections 0.1–0.4. You'll also learn:

- How the concept evolved from the 1950s to 2025 (Section 0.3)

- Why different disciplines define agents differently (Section 0.4)

- Historical context for current debates

- The intellectual foundations of modern definitions

Path 3: Understanding and Assessing

Read all sections for:

- Analytical framework: autonomy, entity frames, goal dynamics (Sections 0.5–0.6)

- Conclusion with key takeaways and reading paths (Section 0.7)

- Deep theoretical foundations across disciplines

What's in Each Section

The table of contents provides section titles and page numbers. Each section begins with a brief overview of its scope and purpose. Sections 0.1–0.2 provide definitions and practical tools; Sections 0.3–0.6 trace historical and disciplinary evolution; Section 0.7 concludes with key takeaways and forward references.

While this scope may seem broad, this chapter is not meant to be everything to everyone. We have purposefully focused on the conceptual and educational, not the technical or promotional.

Specifically, this chapter does not include:

- Marketing claims about "agentic AI" products

- Product reviews or vendor comparisons

- Implementation guides for specific frameworks

- Technical tutorials on building agent systems

Resources on these topics are widely available elsewhere.

0.1 Introduction

Agent. *Agentic.*

Few terms generate more confusion despite widespread use. While these words appear everywhere—from marketing copy to academic papers—their meanings remain contested and often unclear. Yet despite this definitional chaos, the underlying concepts are deeply intuitive and accessible.

At heart, agents are simply **"doers" with a to-do**. As we unpack this accessible starting point, we will discover more explicit conditions for identification. But this four-word formulation captures something essential: agency requires both goals and the capacity to act toward them.

0.1.1 Motivation and Approach

The proliferation of "agentic AI" makes definitional clarity urgent. Existing work remains fragmented across purpose and discipline: computer scientists cite Russell and Norvig (Russell and Norvig 2020), philosophers reference Bratman (Bratman 1987), legal scholars consult the Restatement of Agency (American Law Institute 2006), and commercial vendors seem untethered by anything other than sales.

Some of this fragmentation reflects genuinely different perspectives, such as whether we recognize agents by their *internal properties* (mental states, intentions) or *external manifestations* (observable behavior, delegated authority)—a spectrum we explore in Section 0.4. While theoretical considerations like these can be useful, it is now most critical that we **establish a practical framework** to guide communication and coordination.

To do so, we organize agency into three levels that correspond to answering the following questions:

Level 1: What makes *anything*—biological or computational—an agent?

Level 2: What makes computational systems agentic?

Level 3: How do traditional and AI-powered agentic systems differ?

Answering these progressive questions establishes a nested hierarchy with three levels, as illustrated in Figure 1.
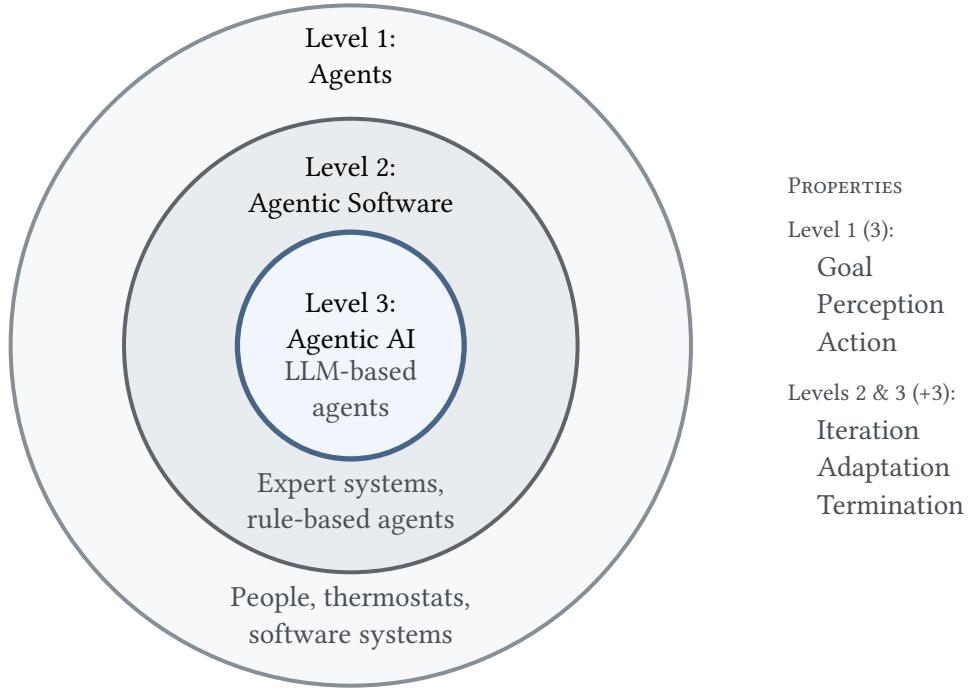
Figure 1: The three-level hierarchy of agency. Each level is a subset of the one above it: all agentic AI is agentic software, and all agentic software consists of agents.

0.1.2   Level 1: Minimal Agency

We begin with the conceptual foundation. What is the absolute minimum required for something to qualify as an agent—whether human, organizational, or computational? Level 1 establishes this baseline, applicable across all domains and technologies.

> ### Level 1: Agent
>
> An **agent** is any system that pursues goals through perception and action, with at least minimal discretion over which action to take in response to what it perceives.
>
> ---
>
> *Examples:* A paralegal researching case law. A thermostat maintaining temperature. An organization pursuing objectives.
>
> ---
>
> **Minimal properties (3):**
> - **Goal**
> - **Perception**
> - **Action**

What these properties mean:

Goal refers to a clear objective or task that the system pursues. These goals may be simple or complex, short-term or long-term, and may change over time. They may be provided by a human user, a software system, or even autonomously generated through "meta-goals." They provide direction and purpose for the system's actions.

Perception is awareness of an environment through sensing capabilities. Notably, this environment need not be physical, let alone real; it could be abstract like a contract bargaining space or simulated in an imaginary world.

Action is the ability to influence the environment, such as by exerting physical forces, modifying the value of variables, executing commands, or any other actuation mechanism, i.e., *actuators*. Minimal *discretion* is assumed: the system can select among at least two possible actions or policies contingent on its perceptions.

This trinity forms the conceptual bedrock of agency—equally applicable to humans navigating social contexts, organizations pursuing strategic objectives, biological organisms seeking survival, or computational systems executing tasks. While these three characteristics suffice for theoretical classification, practical deployment demands additional sophistication.

Distinguishing agents from non-agents reveals critical boundaries. Consider systems we encounter daily that, despite their complexity, fail to meet our criteria. A compiler, though it transforms code through sophisticated analysis, operates without autonomous goals—it executes predetermined transformations. Legal databases, while responsive to queries and rich with information, lack independent objectives or environmental adaptation. Even a single ChatGPT response, impressive as it may be, represents a one-shot generation rather than iterative goal pursuit.

This baseline framework illuminates the essence of agency, yet professional applications demand more. The gap between a simple thermostat cycling toward a temperature target and an AI system

conducting legal research spans more than technological sophistication—it requires architectural elements that ensure reliability, adaptability, and accountability. Legal research tools, portfolio management systems, and document review platforms operate in environments where stakes are high and errors costly. These operational realities shape our expanded framework for deployable agentic systems.

0.1.3   Operational Definition: Agentic Systems

While Level 1 establishes what makes anything an agent, computational systems in production require three additional properties beyond the minimal three. These six properties together define what we call **agentic systems**—the operational standard that bridges conceptual agency and real-world deployment. Both traditional software (Level 2) and AI-powered implementations (Level 3) can achieve this operational standard, though they differ fundamentally in how they realize each property.

Operational Definition: Agentic System

An **agentic system** is a goal-directed agent that repeatedly perceives and acts in its environment, adapting from observations until clear termination conditions are met (explicit or implicit).

Examples: Legal research assistants that iteratively refine queries and generate case summaries. Contract analysis systems that identify risks and produce issue reports. Portfolio rebalancing systems that monitor positions and execute trades. Fraud detection systems that analyze transactions and generate alerts.

Additional properties beyond Level 1 (+3):
- **Iteration** — Repeat perceive-act cycle
- **Adaptation** — Adjust based on feedback
- **Termination** — Stopping conditions (explicit or implicit)

Total: 6 operational properties (3 minimal + 3 additional)

These six properties emerged from decades of agent research as commonly recognized operational requirements for reliable computational deployment. While not formally proven as minimum necessary, they consistently appear across deployed systems in domains from robotics to enterprise software, reflecting lessons from fielding real-world implementations. Section 0.3 traces how each property became recognized as essential through both theoretical development and practical experience.

The three additional properties transform minimal agency into operational systems. **Iteration** sustains the perceive-act cycle, enabling the system to repeatedly gather information and take action rather than executing a single operation. **Adaptation** allows the system to modify its strategies based

on accumulated observations, learning which approaches succeed and which fail. **Termination** provides stopping conditions through either implicit goal satisfaction (reaching target state) or explicit control logic (resource limits, time constraints, escalation triggers), ensuring the system recognizes when to cease operation rather than cycling indefinitely.

The relationship between levels clarifies important boundaries. Every agentic system qualifies as an agent (possessing the minimal three properties), but the reverse doesn't hold—many agents lack the operational sophistication of agentic systems. A thermostat exemplifies the minimal baseline: it perceives temperature (sensor readings), acts (heating/cooling), pursues a goal (maintain target), iterates (continuous monitoring), adapts (responds to temperature changes), and terminates through implicit goal satisfaction (stops heating when target reached, stops cooling when target maintained). This implicit termination—where achieving the target state constitutes the stopping condition—differs from the explicit termination logic (resource budgets, escalation triggers, maximum iteration limits) typical of professional agentic systems. Both forms satisfy the termination requirement, though explicit mechanisms provide stronger operational guarantees for high-stakes deployments. Similarly, a human paralegal demonstrates all six properties behaviorally but operates through cognitive processes rather than discrete computational cycles.

This operational framework now raises the implementation question: *How* do computational systems realize these six properties? The answer reveals an architectural distinction. Some systems use traditional programming—rules, algorithms, control logic—to manage planning and orchestration. Others employ AI/ML, particularly large language models, for these functions. We distinguish these as Level 2 (traditional) and Level 3 (AI-powered). Critically, this distinction is architectural, not evaluative: neither approach is inherently superior, and the boundary between them remains fluid and context-dependent.

0.1.4   Level 2: Traditional Agentic Software

> **Level 2: Traditional Agentic Software**
>
> Traditional agentic software implements the six operational properties using rules, algorithms, or deterministic logic.
>
> ---
>
> *Examples:* Conflicts checking systems that scan firm databases and escalate matches. Regulatory compliance monitoring with rule-based alerts. Trading compliance systems that monitor positions and enforce limits. Invoice processing workflows with validation and retry logic.
>
> ---
>
> **Additional properties beyond agentic systems (+0):**
> - *No new properties — same 6 as agentic systems*

Traditional agentic software uses human-designed programming to implement all six properties.

Planning and orchestration logic is explicitly coded through rules, algorithms, state machines, or control systems. Level 2 systems can be extremely sophisticated: a conflict checking system might employ graph analysis for relationship detection, fuzzy matching for entity resolution, and adaptive threshold tuning—all implemented through traditional programming techniques. The key architectural characteristic is that decision logic is specified by programmers at design time.

A conflicts checking system exemplifies Level 2: it has goals (identify potential conflicts), perception (scans firm databases for client/matter relationships), action (flags matches and escalates to ethics committee), iteration (continuous monitoring as new matters are opened), adaptation (adjusts matching thresholds based on false positive rates), and termination (stops when matter is cleared or rejected). The system achieves these properties through explicitly programmed logic—rules for relationship detection, graph traversal algorithms for indirect conflicts, configurable thresholds for fuzzy name matching.

Level 2 systems achieve all six operational properties through traditional software engineering. Their performance depends on design quality, domain expertise, and implementation rigor—not on whether they employ AI/ML techniques. A well-engineered Level 2 system can substantially outperform a poorly designed Level 3 system in reliability, predictability, and effectiveness for its intended domain.

0.1.5 Level 3: AI-Powered Agentic Systems

Level 3 systems use AI/ML—particularly large language models—to manage planning, orchestration, and adaptation. The architectural distinction from Level 2 is straightforward: where Level 2 systems execute explicitly programmed decision logic, Level 3 systems employ neural networks, language models, or other learned components for these functions. In practice, modern Level 3 systems are typically hybrid: LLMs handle high-level planning and natural language interaction, while traditional code manages structured operations like database queries or API calls. This is what most practitioners mean by "AI agents."

Level 3: AI-Powered Agentic Systems

AI-powered agentic systems implement the six operational properties through strategic integration of artificial intelligence—typically large language models (LLMs), vision-language models (VLMs), or neural networks—with traditional computational components.

Examples: AI legal research assistants that iteratively search case law and synthesize findings. AI contract risk analyzers that identify problematic clauses and recommend revisions. AI-powered document review systems that classify content and adapt to feedback. AI trading assistants that analyze market data and adjust strategies.

Additional properties beyond agentic systems (+0):

> • *No new properties — same 6 as agentic systems*

The architectural choice to use AI/ML for planning and orchestration has practical implications. LLMs enable natural language interfaces—users can specify goals conversationally rather than through structured formats. Neural networks handle pattern recognition tasks that would require extensive rule engineering. However, this approach trades some predictability for flexibility: LLM outputs can vary across runs, and behavior may be harder to audit than explicit rule chains. The boundary between Level 2 and Level 3 can blur—is a system using gradient boosting for fraud detection Level 2 or Level 3? Today's practical dividing line focuses on whether LLMs manage high-level planning and orchestration.

Consider an AI contract risk analyzer as a Level 3 exemplar. It possesses all six operational properties: goals (identify and assess contract risks), perception (reads contract text and clause context), action (flags problematic provisions, generates risk assessments), iteration (reviews document section by section), adaptation (adjusts risk scoring based on clause combinations and jurisdiction), and termination (stops when complete review is done or high-severity risk triggers immediate escalation). The LLM manages high-level planning—deciding which clauses merit detailed analysis, how to interpret ambiguous language, when to flag issues versus when to request human review—while traditional code handles document parsing, clause extraction, jurisdiction lookup, and risk score calculation. This hybrid architecture is typical of Level 3 systems: AI handles interpretation and strategic decisions, traditional programming handles structured data operations.

Table 1 maps the progression from minimal agency (3 properties) through operational deployment (6 properties) to implementation approaches.

0.1.6   Key Distinctions

Having traced the progression from minimal agency (3 properties) through operational deployment (6 properties) to implementation approaches (traditional vs. AI), we can now synthesize what this hierarchy reveals. Level 1 establishes conceptual qualification, the operational definition adds production-readiness requirements, and Levels 2–3 distinguish implementation paradigms. This structure clarifies three critical distinctions that cut through definitional confusion:

The critical distinction lies between *properties* and *implementation*. The major jump in capability occurs between Level 1 (3 properties) and operational deployment (6 properties). Levels 2 and 3, by contrast, have *identical property requirements*—they differ only in how those properties are implemented: through explicit rules and logic (Level 2) or through AI/ML models (Level 3).

This hierarchy enables precise terminology:

Property	Agent Level 1 Conceptual	Agentic Systems Level 2 Traditional	Level 3 AI	Description
Goal	●	●	●	Clear objective to pursue
Perception	●	●	●	Sense environment & results
Action	●	●	●	Affect environment
Iteration	○	●	●	Repeat perceive-act cycle
Adaptation	○	●	●	Adjust based on feedback
Termination	○	●	●	Stopping condition (explicit or implicit)
AI-Powered	○	○	●	Uses AI/ML for implementation

Table 1: Property requirements by level. Filled circles indicate required properties: blue (filled) for minimal agency (Level 1), amber (filled) for additional operational properties (Levels 2 & 3). Empty circles indicate properties not required. Below the horizontal rule: green (filled) indicates AI-powered implementation (Level 3 only).

Terminology Precision

- **"Agent" (noun):** Any system meeting Level 1's three minimal properties
- **"Agentic" (adjective):** At the system level, describes systems meeting all six operational properties. We may also use it descriptively at the feature/behavior level (e.g., "agentic behavior/properties"); reserve "agentic system" for six-of-six conformance.
- **"AI agent":** An agentic system specifically powered by AI/ML (Level 3)

These definitions enable clear exclusions:

What Doesn't Qualify

- Compilers, databases → lack goals (not Level 1)
- Single chatbot responses → lack iteration (not operational)
- Traditional ML classifiers → lack iteration and goals (not agentic)
- Rule-based expert systems → agentic but not AI-powered (Level 2, not Level 3)

This framework provides scaffolding for the historical and theoretical analysis that follows. Tracing definitions across decades reveals how each level emerged from evolving technological capabilities and disciplinary concerns. But before diving into that scholarly context, we must address a pressing practical question: Why does definitional precision matter? In professional domains where autonomous systems make consequential decisions—legal research, medical diagnosis, financial advising—the distinction between marketing labels and architectural reality has high stakes.

0.1.7   Why This Matters: Stakes and Applications

With our three-level hierarchy and technical definition established, the practical implications become clear. The stakes extend far beyond academic precision—they shape how professionals evaluate systems, how researchers coordinate findings, how regulators craft policy, and how we hold vendors accountable for their claims.

Using the framework that we develop here, we can:

Clearer Evaluation

Distinguishing genuinely agentic systems from chatbots by assessing whether they meet all six operational properties or merely respond to single prompts (Table 1).

Better Research Coordination

Ensuring scholars reference compatible concepts by specifying which properties their systems exhibit rather than relying on ambiguous labels like "agents" or "autonomous systems."

Informed Regulation

Crafting policies that target measurable capabilities—the six operational properties—rather than vendor marketing claims about "AI agents."

Vendor Accountability

Holding companies to falsifiable claims by requiring demonstration of all six operational properties, not just conversational ability or AI-powered text generation.

The stakes are particularly high in professional domains like law, medicine, and finance, where autonomous action by AI systems raises questions of **liability**, **professional responsibility**, and public safety. For legal professionals, *Mata v. Avianca* sanctions ("US lawyer sanctioned for using ChatGPT to make up case citations" 2023) illustrate the consequences of deploying systems without understanding their architectural properties. Our framework provides the conceptual tools to avoid such failures—not by slowing adoption, but by enabling more precise evaluation of capabilities and limitations.

The remaining sections explain where this definition came from and why it takes this particular form. Section 0.2 provides a practical decision rubric for applying this definition.

0.2   How to Recognize an Agent

With the three-level hierarchy and complete definition established in Section 0.1, you now have the conceptual foundation. But recognizing agents in practice requires operational tools. This section provides practical approaches: a detailed evaluation rubric, concrete examples comparing agents to non-agents, and guidance for navigating common misconceptions.

0.2.1 The 6-Question Evaluation Rubric: Operational Properties

These six questions assess whether a system qualifies as an *agentic system*. They operationalize the six-property definition established in Section 0.1. The first three (Goal, Perception, Action) establish *Level 1 agency*; Questions 4–6 add the operational properties required for deployment.

Q1. Does it have goals?

Explicit objectives or implicit performance measures? Look for clear task specifications, desired outcomes, or optimization targets.

Q2. Does it perceive?

Access to environmental information through sensors, APIs, databases, or document access? Can it observe the results of its actions?

Q3. Does it act?

Capability to affect its environment—retrieve documents, generate text, invoke tools, move actuators, modify data?

Q4. Does it iterate?

Multiple cycles of observation and action, not single-shot processing? Does it loop: act → observe → act again?

Q5. Does it adapt?

Does behavior change based on observations or feedback? Can it adjust strategies when initial approaches fail?

Q6. Does it stop?

Clear termination conditions, either implicit (goal satisfaction, target state reached) or explicit (time limits, resource budgets, escalation triggers, maximum iterations)?

If Q1–Q3 are yes, the system qualifies as an agent. If all six answers are yes, it qualifies as an agentic system. Otherwise, it may be a useful tool but lacks full agentic character.

Note on Professional Governance: The six questions above establish whether something is an agent. Professional deployment in high-stakes domains (legal research, medical diagnosis, financial advising) requires additional governance safeguards—attribution, explanation, escalation, and confidentiality. These professional governance requirements are covered in detail in a companion chapter on governance and risk management for agentic systems.

0.2.2 Common Misconceptions

Understanding what agents are requires equally understanding what they are *not*. Five common misconceptions lead to over-attribution of agency:

---

1. Single-Shot Responses Are Not Agentic Systems

A single response from a language model is **not** an agentic system, even if sophisticated. Without iteration and adaptation, it's a one-time transformation.

Example: A single query to a research system returns results. No perception-action loop occurs.

---

2. Automation Alone Is Not an Agentic System

Automation doesn't imply an agentic system. A data extraction script that automatically runs nightly has goals (extract data) and acts (writes to database), but lacks perception, adaptation, and iteration. It's a scheduled task, not an agentic system.

---

3. Tool Use Alone Is Not Agency

Calling external tools doesn't automatically make a system agentic. The question is: does it iterate on tool results, adapting its strategy?

Example: A script that queries an API once is not an agent. A research system that queries the API, evaluates relevance, and decides what to search next may be.

---

4. Complexity Is Not Agency

Complex systems aren't necessarily agents. A document processing pipeline performs sophisticated transformations but follows a fixed sequence without goals, perception of results, or adaptation.

Conversely, simple systems can be agents—a thermostat has goals (maintain temperature), perceives (reads sensor), acts (turns heat on/off), and iterates.

---

5. AI-Powered Is Not Agency

Using AI/ML doesn't make something an agent. A neural network that classifies documents in a single forward pass is not an agent.

An AI system that iteratively reviews content, flags issues, and refines assessments can be.

0.2.3 Examples: Agents vs Non-Agents

Table 2 orders systems along the spectrum of agency, from non-agents through AI-powered agents. Each tier illustrates increasing qualification based on the six core properties. Detailed explanations follow.

System	Status	Key Properties
Not Agents (Missing Critical Properties)		
Form validation script	No	One-pass validation; lacks goals, perception, and adaptation
Agents with Incomplete Agentic System Properties (Missing 1–2)		
Single query to research system	Partially	Has goal, perception, and action; lacks iteration and adaptation
Rule-based pattern detection	Partially	Iterates continuously but lacks adaptive strategy
Content suggestion system	Partially	Updates per interaction but fixed learning approach
Full Agents: Traditional (Rule-Based)		
Thermostat	Yes	All 6 properties; minimal but complete baseline
Portfolio rebalancing system	Yes	Monitors, adapts to volatility, stops when balanced
Full Agents: AI-Powered (Flexible Reasoning)		
Document review system	Yes	Iterative classification with adaptive learning
AI research assistant	Yes	Tool loop with query refinement and escalation

Table 2: Spectrum of agency ordered by increasing qualification, from non-agents through AI-powered systems.

Not Agents. These systems lack the minimal properties (goals, perception, action). The **form validation script** checks inputs against rules in a single pass—it has no independent goals beyond validation, no perception of results, and no adaptation.

Agents with Incomplete Agentic System Properties. These systems qualify as agents (they satisfy goals, perception, and action) but lack one or more operational properties required for agentic systems. A **single query to a research system** produces one response and stops, lacking the iteration and adaptation required for agentic systems despite using AI.

Agents with Incomplete Agentic System Properties. These systems qualify as agents (they exceed the 3-property minimum for Level 1 agency) but lack one or two of the six operational properties required for full agentic systems. **Rule-based pattern detection** has a goal (flag patterns), perceives incoming data, acts (blocks or alerts), and iterates continuously with each input. However,

it typically doesn't adapt its detection strategy within a session—it applies fixed rules. **Content suggestion systems** similarly have a goal (suggest relevant content), perceive context, act (display suggestions), and iterate with each interaction. Yet they use a fixed learning strategy rather than adapting their suggestions based on user acceptance patterns within a session.

Full Agents: Traditional. **Thermostats** represent the minimal baseline—all 6 properties at their simplest: goal (maintain temperature), perception (sensor readings), action (heat on/off), iteration (continuous monitoring), adaptation (reacts to temperature changes), and termination through implicit goal satisfaction (stops heating when target reached). Basic thermostats rely on this implicit termination—the target state itself defines when to stop—while modern smart thermostats add explicit termination logic like time windows, energy budgets, or scheduled modes. **Portfolio rebalancing systems** demonstrate explicit termination through multiple stopping conditions: goal achievement (balanced portfolio), temporal constraints (market close), or resource limits (maximum trades per session). They monitor market data and portfolio drift, generate trade orders, adapt to volatility within predefined parameters, and stop when any termination condition is met.

Full Agents: AI-Powered. **Document review systems** use machine learning to iteratively classify documents based on relevance criteria. Unlike rule-based systems, they improve their categorization as they process more examples, adapting their classification strategy based on observed patterns. **AI research assistants** demonstrate the most sophisticated agency: they maintain goals (answer questions), observe search results from multiple sources, query iteratively, refine search strategies when initial approaches fail, and escalate to human oversight when encountering contradictory information or reaching confidence limits. This represents the convergence of all six properties enhanced by AI's flexible reasoning capabilities.

0.2.4   When to Call Something an Agent

Our taxonomy (Section 0.1) distinguishes "agent" (3 minimal properties) from "agentic system" (6 operational properties). This section provides practical guidance on applying these terms in professional contexts.

Use agent when: A system has the three minimal properties (goals, perception, action). This includes simple systems like thermostats alongside sophisticated AI systems. The term captures the conceptual essence but doesn't guarantee operational readiness.

Use agentic system when: All six operational properties are present (goals, perception, action, iteration, adaptation, termination). This signals production-ready systems suitable for professional deployment.

Use agentic AI when: The six properties are implemented through artificial intelligence rather than traditional programming.

For systems with partial properties:

- **3 properties (agent):** Use "agent" correctly—it meets the baseline. Clarify it's not an agentic system if operational properties are missing.

- **4-5 properties:** Specify which properties are present: "agent with iterative capability but fixed strategy" or "agent lacking explicit termination."

- **1-2 properties:** Avoid "agent"—describe what it actually does: "tool," "function," "transformation."

The term agentic is useful for:

- Describing properties: "exhibits agentic properties" or "agentic behavior"

- Architectural patterns: "agentic architecture" or "agentic workflow"

- Adjective usage when precision isn't critical

Context-dependent usage:

The level of precision required varies by context:

Regulatory filings or legal documents:
Use precise language. Specify whether the system qualifies as an agent (3 properties), agentic system (6 properties), or neither. List which properties are present.

Academic papers:
Be explicit about your framework. State which definition you're using and which properties your system exhibits. Distinguish agent from agentic system clearly.

Informal discussion or presentations:
Use "agent" for systems meeting the 3-property baseline, "agentic system" for the 6-property standard. When uncertain, describe specific properties: "has goals and perception but limited adaptation."

Marketing or vendor communications:
Be honest and specific. State which properties your system has. Don't claim "agentic AI" unless all six properties are implemented through AI/ML.

Quick decision guide:

- 3 minimal properties → "agent"

- 6 operational properties → "agentic system"

- 6 properties via AI → "agentic AI"

- 1-2 properties → Not an agent; describe actual function

- 4-5 properties → Specify: "agent with [properties]"

When in doubt, use the 6-question rubric (subsection 0.2.1). It provides clear, answerable criteria that cut through ambiguity. For professional deployments in high-stakes domains, additional governance safeguards are covered in our companion chapter on governance frameworks.

> **Stop Here If You're Busy**
>
> **You now know enough to:**
>
> - Define what an agent is (Section 0.1)
> - Recognize agents using the 6-question rubric (subsection 0.2.1)
> - Distinguish agents from non-agents (examples above)
> - Evaluate marketing claims about "agentic AI"
> - Use the terminology correctly in professional contexts
>
> **The remaining sections** provide optional context:
>
> - Section 0.3: How the concept evolved (1957–2025)
> - Section 0.4: Why disciplines define agents differently
> - Section 0.5: Analytical framework for understanding variation
> - Section 0.7: Key takeaways and reading paths

0.3   A Historical Journey: Defining Agents Across Seven Decades

You now have practical tools for recognizing agents. This section provides optional historical context—tracing how the concept evolved across seven decades and multiple disciplines. Understanding this evolution helps explain why definitions vary so widely today and why the six-property framework emerged as it did. If you're satisfied with the practical rubric, you can skip to Section 0.7.

The concept of **agency** has evolved dramatically since the mid-20th century, shaped by philosophical inquiry, economic theory, legal doctrine, and technological capability. This section traces agent definitions chronologically, revealing how each level of the hierarchy from Section 0.1 emerged historically: philosophical and legal foundations addressed general agency (Level 1), the computer science revolution instantiated these concepts in software (Level 2), and the LLM era added flexible

reasoning powered by AI (Level 3). Figure 2 provides a visual overview of key milestones.

> **On Etymology and Ancient Polysemy.** The semantic complexity surrounding "agent" has ancient roots. The Latin root *agō*, from which *agent* derives, had multiple meanings in classical Latin. With numerous distinct senses in Lewis & Short's classical dictionary, ranging from "driving cattle" to "pleading a case," contemporary definitional debates echo longstanding ambiguity (Lewis and Short 1879).

Evolution of Agent Definitions (1957–2025)

1950s–60s	1970s–80s	1990s	2000s–10s	2020s

1957 Anscombe
Intention

1958 Restatement
Legal Agency

1963 Davidson
Causal Theory

1974 Milgram
Agentic State

1976 Jensen
& Meckling
Principal-Agent

1986 Minsky
Society of Mind

1987 Dennett
Intentional Stance

1989 Bandura
Human Agency

1991 Shoham
AGENT0, AOP

1994 Maes
Software Agents

1995 Russell & Norvig
Perception-Action

1995 Wooldridge
Intelligent Agents

1996 Franklin
Autonomous Agents

1998 Sutton & Barto
RL Agents

2001 AOSE Roadmap
Software Agents

2002 Bonabeau
Agent-Based Models

2006 Restate-
ment (Third)
Updated Law

2022 Yao et al.
ReAct

2023 Park et al.
Generative Agents

2024 IBM, Google
Agentic AI

2025 Ope-
nAI, Anthropic
Tools-in-a-Loop

Legend (8 Disciplines):

Philosophy   Psychology   Law   Economics

Cognitive Sci   Complex Systems   Comp Sci   AI

Figure 2: Timeline of key milestones in agent definitions across eight disciplines (1957–2025). The figure traces the intellectual evolution of "agency" from philosophical accounts of intentionality (1950s–80s) through the computational revolution (1990s) and toward modern convergence around LLM-based agentic systems (2020s). Each discipline has a distinct visual marker combining border patterns (solid, dashed, dotted, double, thick) with color families: blue for theoretical foundations (philosophy, psychology, cognitive science, computer science) and orange for applied domains (law, economics, complex systems, AI engineering).

0.3.1 Philosophical and Legal Foundations (1957–1969)

The postwar period witnessed foundational philosophical work on action theory alongside the maturation of legal doctrine concerning agency relationships. These developments occurred independently yet addressed complementary questions: philosophy examined how intention relates to action and how we attribute actions to agents, while law formalized the conditions under which one party acts on behalf of another.

Three landmark works from this period established conceptual foundations that continue to shape agent definitions across disciplines. Anscombe provided a framework for understanding intentional action as distinct from mere behavior. Davidson connected mental states causally to actions, offering a naturalistic account of agency. The legal Restatement formalized the delegation relationship at the core of practical agency arrangements. Each addressed different aspects of the same underlying question: what makes an entity an agent?

These foundations set up core tensions that would animate later debates: the relationship between mental states and observable behavior, the balance between autonomy and control in delegated relationships, and the explanatory role of reasons versus causes. As computational systems began to exhibit action-like behavior, these philosophical and legal frameworks provided conceptual resources for interpreting artificial agency.

1957: Intention

G.E.M. Anscombe's *Intention* established that actions are **intentional** "under a description" and known through practical knowledge (Anscombe 1957). This introduced the idea that intention plays a basic explanatory role: we understand actions by understanding the reasons under which agents perform them.

1958: Legal Agency

The *Restatement (Second) of Agency* defined **agency relationship**: a principal manifests assent that an agent shall act on the principal's behalf and subject to control (American Law Institute 1958). This emphasizes delegation, fiduciary relation, and control—concepts relevant when discussing AI systems acting on behalf of humans.

1963: Causal Theory of Action

Donald Davidson's "Actions, Reasons, and Causes" defended the **causal theory of action**, arguing that intentional actions are explained by an agent's primary reason (belief–desire pair) that causally produces the action (Davidson 1963). This framework connected mental states to observable behavior.

These three works established complementary foundations for thinking about agency. Philosophy offered frameworks for intention and the causal structure of action, while law formalized delegation and control in principal–agent relationships. Together, they highlighted elements that would need to be addressed as the concept of agency expanded beyond human actors: intentionality, causal efficacy, goal-directedness, and the capacity to act on behalf of another. When researchers began implementing computational agents in subsequent decades, they inherited both the conceptual resources and the unresolved tensions from this period.

0.3.2 Economic and Social Theory (1970–1989)

The 1970s and 1980s witnessed agency concepts migrating from philosophy and law into the social sciences. Economics, sociology, and psychology each adapted the core ideas to their disciplinary concerns, generating new frameworks for understanding human behavior in organizations, social structures, and economic relationships. This period produced a remarkably diverse set of perspectives on agency, from Milgram's minimalist "agentic state" where individuals function as mere instruments, to Bandura's rich conception of human agency with intentionality, forethought, and self-regulation.

A key theme across these developments was the spectrum of autonomy. Economic models formalized principal-agent relationships as contracts with delegated decision-making but misaligned incentives, focusing on control mechanisms and information asymmetries. Sociological theories emphasized agents' capacity to act independently within structural constraints. Cognitive science explored how minds might be composed of simpler agents, while philosophy developed practical frameworks for planning and prediction. By the end of this period, researchers had articulated both the minimal conditions for agency (following another's directives) and the maximal human capacities (self-reflective goal pursuit).

These social science perspectives proved directly influential for computational implementations in the 1990s. Bratman's BDI framework became a dominant architecture for intelligent agents. Dennett's intentional stance provided a pragmatic criterion for when to treat systems as agents. Bandura's enumeration of human agency properties became a checklist for artificial systems. The diversity of definitions reflected different disciplinary needs, but collectively they mapped out the conceptual terrain that computer scientists would navigate when implementing computational agents.

1974: The Agentic State

Stanley Milgram's *Obedience to Authority* introduced the "**agentic state**"—individuals seeing themselves as instruments carrying out another's wishes (Milgram 1974). This anchors one end of the autonomy spectrum: agents as pure delegates without independent judgment.

1976: Principal-Agent Economics

Jensen and Meckling's "Theory of the Firm" formalized the **principal-agent relationship**, defining it as a contract where principals engage agents with delegated decision-making authority (Jensen and Meckling 1976). This established vocabulary of agency costs, information asymmetry, and incentive alignment.

1984: Structuration Theory

Anthony Giddens framed agency as the capacity to make a difference and intervene in the world (Giddens 1984).

1986: Society of Mind

Marvin Minsky's *Society of Mind* popularized the idea of mind as a society of simple "agents" whose interactions generate intelligence, introducing the **multi-agent system** metaphor (Minsky 1986).

1987: Planning Theory

Michael Bratman's *Intention, Plans, and Practical Reason* developed a planning theory emphasizing partial plans structuring practical reasoning (Bratman 1987). This introduced the Belief-Desire-Intention (BDI) framework that became foundational for intelligent agent architectures.

1987: The Intentional Stance

Daniel Dennett's *The Intentional Stance* argued we can treat systems as agents when attributing beliefs and desires yields reliable predictions—providing a pragmatic criterion for agency (Dennett 1987).

1989: Human Agency

Albert Bandura synthesized psychological research, defining human agency through intentionality, forethought, self-regulation, and self-reflectiveness—properties increasingly sought in artificial systems (Bandura 1989).

These seven definitional works from 1974–1989 collectively established a spectrum of agency from minimal delegation to full autonomy. At one extreme, Milgram's agentic state described humans functioning as mere instruments with no independent judgment. At the other, Bandura enumerated

the rich capacities of human agency: forming intentions, planning ahead, monitoring and regulating behavior, and reflecting on one's own efficacy. Between these poles, economic models formalized the incentive problems in delegated decision-making, sociological theories examined how agents navigate structural constraints, and philosophical frameworks provided practical tools for prediction and planning.

This diversity reflected genuine differences in disciplinary focus—economics concerned with contracting problems, psychology with cognitive capacities, sociology with social structures. But the frameworks proved complementary rather than contradictory. The BDI architecture from Bratman's planning theory, the intentional stance from Dennett's pragmatic philosophy, and the property lists from Bandura's psychology would all become standard tools for designing and analyzing computational agents. The 1990s explosion of agent-oriented computing inherited this rich conceptual toolkit, translating social science insights into software architectures.

0.3.3   The Computer Science Revolution (1990–1999)

The 1990s saw a proliferation of work on computational agents, driven by the convergence of distributed systems, AI research, and the rapid growth of the internet. Previous decades had provided rich conceptual frameworks—intention, planning, autonomy, perception–action cycles—but these primarily described human or abstract theoretical entities. The 1990s marked a clear shift: researchers implemented these concepts directly in software, creating systems designed from the ground up as agents rather than as traditional programs.

This transition from human to computational agency required translating philosophical and social science concepts into executable architectures. How should beliefs and intentions be represented in code? What makes a software system autonomous rather than merely reactive? When do local interactions among simple agents produce emergent behavior? The decade produced multiple answers to these questions, each emphasizing different aspects of agency: mental states (BDI architectures), observable properties (autonomy, reactivity, proactivity), functional roles (perception–action), computational paradigms (reinforcement learning), and emergence from interaction (complex adaptive systems, agent-based modeling).

The proliferation of definitions reflected both vitality and fragmentation. Researchers drew selectively from the prior decades' conceptual toolkit, implementing what could be formalized and computed. The result was a rich ecosystem of agent types serving different purposes: software agents assisting users, intelligent agents in multi-agent systems, autonomous agents pursuing goals in environments, adaptive agents learning through experience. By decade's end, agent-oriented computing had established itself as a distinct paradigm with its own conferences, journals, and textbooks.

1991–1993: Agent-Oriented Programming

Yoav Shoham's AGENT0 and "Agent-Oriented Programming" implemented agents as computational entities characterized in mentalistic terms (beliefs, commitments, obligations) (Shoham 1993). This established **agent-oriented programming** as distinct from object-oriented programming and was closely related to the BDI (Belief-Desire-Intention) architecture.

1994–1995: Three Foundational Perspectives. Pattie Maes envisioned adaptive, long-lived software assistants that learn user preferences (Maes 1994). Three major 1995 works captured distinct facets:

1995: Intelligent Agents

Wooldridge & Jennings: **Intelligent agents** exhibit autonomy, reactivity, proactivity, and social ability (Wooldridge and Jennings 1995).

1995: Perception–Action Agents

Russell & Norvig: An **agent** perceives its environment through sensors and acts through actuators (Russell and Norvig 1995).

1995: Complex Adaptive Systems

Holland: **Complex adaptive system agents** exhibit local interactions under simple rules that produce emergent behavior (Holland 1995).

1996: Autonomous Agents

Franklin and Graesser distinguished **autonomous agents** as systems situated within an environment that sense and act over time in pursuit of their own agendas (Franklin and Graesser 1997).

1996: Agent-Based Modeling

Epstein and Axtell's *Growing Artificial Societies* established agent-based modeling (ABM), defining agents as autonomous heterogeneous individuals whose local interactions produce emergent macro patterns (Epstein and Axtell 1996).

> **1998: Reinforcement Learning**
>
> Sutton and Barto's *Reinforcement Learning* defined agents maximizing cumulative reward through environmental interaction, establishing the **RL framework** (Sutton and Barto 1998).

> **1999: Multi-Agent Systems**
>
> Edited volumes by Gerhard Weiss and Jacques Ferber consolidated **multi-agent systems (MAS)**, defining agents as autonomous problem-solving entities that cooperate, compete, and negotiate to achieve individual and collective goals (Weiss 1999; Ferber 1999).

The 1990s produced several major definitional frameworks, each capturing different facets of agency. Three complementary 1995 perspectives became particularly influential: Wooldridge and Jennings' property-based view (autonomy, reactivity, proactivity, social ability), Russell and Norvig's functional view (perception and action), and Holland's emergence-based view (simple rules producing complex behavior). These were not competing definitions but rather different levels of description—properties agents should exhibit, functional roles they should fill, and mechanisms by which agency emerges.

The decade also established two influential computational paradigms. Agent-based modeling demonstrated how simple local rules could generate macro-level social phenomena, providing a powerful tool for social science. Reinforcement learning formalized agents as reward maximizers learning through environmental interaction, a framework that proved durable and scalable. By 1999, multi-agent systems had matured into a distinct field with established textbooks, reference works, and architectural patterns. These 1990s frameworks would shape agent definitions for the next two decades, until large language models introduced qualitatively new capabilities.

0.3.4   Consolidation and Formalization (2000–2019)

The early 21st century marked a period of consolidation rather than conceptual revolution. Following the 1990s explosion of agent definitions and implementations, the field matured through textbooks that trained new generations of researchers, reference works that synthesized accumulated knowledge, and successful demonstrations that validated the frameworks at scale. The dot-com boom and bust, the rise of Web 2.0, mobile computing, and eventually the deep learning revolution all occurred during this period, yet they produced remarkably few new definitional frameworks for agency itself.

Two early definitional contributions framed the era's concerns. Kauffman raised fundamental questions about the relationship between physical and virtual agency, proposing thermodynamic criteria that most software agents clearly failed to meet. Jennings, Sycara, and Wooldridge formalized agent-

oriented software engineering, translating the 1990s research insights into practical development methodologies. Beyond these, the period's major developments were consolidative: textbooks like Wooldridge's *Introduction to MultiAgent Systems* and Epstein's *Generative Social Science* codified best practices, legal updates like the Restatement (Third) of Agency maintained doctrinal continuity, and reference works like the Stanford Encyclopedia entries provided authoritative syntheses.

The period's most dramatic developments came not from new definitions but from demonstrations of existing frameworks at unprecedented scale. Deep reinforcement learning combined neural networks with the RL framework established by Sutton and Barto in 1998, achieving superhuman performance first in Atari games and then notably with AlphaGo's victories over world champions. These successes validated that the 1990s conceptual frameworks could scale to complex domains, but they did not fundamentally redefine what agents are. By 2019, the field had mature definitions, established textbooks, and powerful demonstrations—but also growing awareness that large language models might catalyze the next definitional shift.

> **2000: Physical Agency**
>
> Stuart Kauffman's *Investigations* proposed thermodynamic criteria: natural autonomous agents must reproduce and perform thermodynamic work cycles, highlighting tensions between embodied and virtual agency (Kauffman 2000).

> **1998 Roadmap: Agent Definition Consensus**
>
> Jennings, Sycara, and Wooldridge's 1998 research roadmap synthesized the agent research community's consensus: an agent is "an encapsulated computer system that is situated in some environment, and that is capable of flexible, autonomous action in that environment in order to meet its design objectives" (Jennings et al. 1998; Wooldridge and Jennings 1995). This definition grounded agent-oriented software engineering (AOSE) work throughout the 2000s.

> **2006: Restatement (Third) of Agency**
>
> The *Restatement (Third) of Agency* updated legal doctrine while retaining core concepts of fiduciary duty and delegated authority (American Law Institute 2006).

Textbooks and Consolidation. Wooldridge's *Introduction to MultiAgent Systems* (Wooldridge 2009) and Bonabeau's agent-based modeling syntheses (Bonabeau 2002) trained the next generation. Epstein's *Generative Social Science* articulated ABM's explanatory program: explaining phenomena through simple rule-based actors whose local interactions generate macro patterns (Epstein 2006). Stanford Encyclopedia entries on "Agency" and "Action" provided philosophical syntheses (Schlosser 2019; Wilson and Shpall 2022). By 2020, Russell and Norvig's 4th edition of AIMA had refined agent

definitions incorporating ML progress (Russell and Norvig 2020).

Deep Reinforcement Learning Demonstrations. Deep RL combined neural networks with RL frameworks, achieving superhuman performance in complex domains. DeepMind's DQN mastered Atari games from raw pixels (Mnih et al. 2015), and AlphaGo defeated world champions at Go (Silver et al. 2016), demonstrating that learned agents could exceed human capabilities in domains previously considered intractable. These successes validated the RL agent framework at unprecedented scale and complexity.

The 2000–2019 consolidation period produced remarkably little definitional innovation compared to the 1990s explosion. Kauffman's thermodynamic criteria highlighted the still-unresolved tension between physical and virtual agency—a question that remains contentious as purely software-based LLM agents proliferate. Jennings and colleagues' software engineering formalization helped practitioners but didn't redefine agency itself. Otherwise, the period's contributions were consolidative: textbooks trained practitioners, demonstrations validated frameworks at scale, and reference works synthesized accumulated knowledge.

This relative stability reflected the 1990s frameworks' success—they fit the period's technological capabilities. Deep RL showed that agents learning through environmental interaction could achieve superhuman performance in complex domains, validating Sutton and Barto's 1998 framework rather than replacing it. The stage was set, however, for disruption. Large language models' capacity for flexible reasoning, few-shot learning, and natural language understanding would soon enable agent architectures qualitatively different from the hand-coded systems of previous decades.

0.3.5 The LLM Era (2020–2025)

Large language models brought capabilities that catalyzed a new wave of agent definitions. GPT-3's release in 2020 demonstrated few-shot learning and flexible reasoning at scale. ChatGPT's public launch in 2022 made conversational AI mainstream. But the definitional shift came not from language modeling itself, but from discovering that LLMs could orchestrate tool use, maintain working memory, and iteratively refine approaches to achieve complex goals. This combination—flexible reasoning plus tool use—enabled agent architectures qualitatively different from previous decades' hand-coded systems.

The key insight emerged in 2022: LLMs could implement the classical perception–action cycle not through programmer-specified rules but through learned parameters. The ReAct pattern (Reasoning and Acting) demonstrated LLMs interleaving chain-of-thought reasoning with tool use, iteratively planning, acting, and observing until goals were achieved. This simple pattern proved general, spurring implementations across research labs and commercial frameworks. Unlike previous decades'

diversity of agent architectures, the LLM era quickly converged on a dominant pattern: an LLM iteratively calling tools, observing results, updating state, and choosing next actions.

This architectural convergence reflects several factors: the flexibility of LLMs as general-purpose reasoners, the natural fit between language models and tool APIs, and the economic incentives driving rapid commercialization. Where 1990s agent definitions emerged from academic research programs over a decade, LLM agent definitions coalesced in months through practitioner experimentation and commercial deployment. The speed of convergence is unprecedented, though whether this pattern will prove as durable as the 1990s frameworks remains to be seen.

2022: The LLM-as-Agent Pattern

Yao et al.'s "ReAct: Synergizing Reasoning and Acting in Language Models" showed that LLMs can interleave chain-of-thought reasoning with tool use, iteratively planning, acting, and observing to achieve goals (Yao et al. 2022). This catalyzed the "**LLM-as-agent**" pattern in contemporary practice. As Simon Willison concisely puts it: "an LLM agent runs tools in a loop to achieve a goal" (Willison 2025).

Framework documentation adopted this pattern explicitly. LangChain defines agents as systems that use LLMs to iteratively call tools until a stop condition is met (LangChain Documentation 2024). OpenAI's Agents SDK similarly describes agents as LLMs configured with instructions and tools, operating in loops where the LLM decides which tools to call based on iterative feedback (OpenAI 2024). This convergence across academic research and commercial frameworks reflects rapid standardization around a single architectural pattern.

2023: Generative Agents

Park et al.'s "Generative Agents" showcased LLM-powered agents simulating believable human behaviors over long horizons via memory, planning, and reflection, popularizing sandboxed LLM agents in society simulations (Park et al. 2023).

2023: The Rise of LLM Agents

Xi et al.'s survey "The Rise and Potential of Large Language Model Based Agents" documented the emerging architectural consensus: systems using LLMs to iteratively call tools, observe results, and adapt until goals are achieved (Xi et al. 2023).

> **2024–2025: Commercial Frameworks**
>
> LangChain defines agents as systems using LLMs to iteratively call tools until a stop condition is met (LangChain Documentation 2024). OpenAI's Agents SDK describes agents as LLMs configured with instructions and tools, operating in loops where the LLM decides which tools to call based on iterative feedback (OpenAI 2024). This convergence represents an emerging architectural consensus across research and commercial frameworks.

The LLM era's definitional landscape differs from previous periods. Rather than multiple competing frameworks addressing different facets of agency, the field quickly converged on a single architectural pattern: LLMs orchestrating tools in iterative loops. ReAct established the template in 2022, and within months implementations proliferated across academia and industry. Park's generative agents demonstrated the pattern's applicability to social simulation, while commercial frameworks embedded it as default infrastructure.

What is genuinely new remains debated. Skeptics note that the perception–action cycle, goal-directed behavior, and tool use all appeared in 1990s definitions—the LLM era implements these through learned parameters rather than programmer-specified logic. Proponents argue that flexible reasoning over natural language representations enables qualitatively different agentic capabilities. A plausible resolution is that LLMs shifted where decision-making knowledge resides (from code to weights) without fundamentally changing the functional architecture of agency. Regardless, the speed of convergence—from research demonstration to industry standard in under three years—marks the LLM era as distinctive in the seven-decade history of agent definitions.

0.3.6 Historical Patterns

> **Five Patterns Across Seven Decades**
>
> Five patterns emerge:
>
> **Broadening entity frames.** Early definitions focused on humans or human–AI relationships. By the 1990s, purely computational agents became common. The LLM era introduced hybrid framings: humans provided goals, AI systems executed multi-step tasks.
>
> **Increasing autonomy.** Definitions migrated from delegated proxies (1958) through perception–action systems (1995) to self-directed tool orchestrators (2025). The locus of decision-making shifted progressively toward the agent.
>
> **From mental states to observable behavior.** Philosophical definitions emphasized intention and reasoning. Computer science implemented these as mentalistic constructs. Contemporary definitions focus on observable capabilities—tool use, iterative problem solving, task

completion.

"Tools-in-a-loop" convergence. The AI engineering community has converged toward iterative tool orchestration patterns, documented in practitioner accounts (Willison 2025) and academic surveys (Xi et al. 2023). This reflects technological maturity (LLMs capable of reliable tool use), research foundations (ReAct and related frameworks), and market pressures (customers evaluating empirical capabilities). While not yet a formal standard, this architectural pattern dominates contemporary implementations.

From hand-coded to learned behavior. Early agents used explicit rules and plans specified by programmers. Deep RL demonstrated agents learning behaviors through environmental interaction. LLMs shifted the architectural locus of planning and orchestration from programmer-specified logic to learned parameters, changing where decision-making knowledge resides rather than the fundamental agentic capabilities.

These patterns help explain why contemporary definitions vary: they reflect different points in this evolutionary trajectory and different disciplinary emphases.

The next section examines how different disciplines approached agency, revealing persistent tensions and complementary insights.

0.4   Disciplinary Perspectives on Agency

While Section 0.3 traced chronological evolution, this section examines how different disciplines approach agency from distinct theoretical foundations. Each field emphasizes different facets of agency, revealing complementary insights and persistent tensions.

0.4.1   Philosophy: Intentionality and Reasons

Philosophy's distinctive contribution centers on **intentional descriptions**, responsibility for actions, and predictive stances. Anscombe (1957) established that actions are intentional "under a description"—the same physical movement can be intentional under one description ("signing a contract") but not another ("smudging ink"). This insight matters for AI agents: when we attribute intentions to systems, we're choosing explanatory frameworks, not discovering intrinsic properties.

Bratman (1987)'s planning theory and Dennett (1987)'s **intentional stance** provide complementary perspectives: agents act according to their plans and commitments (Bratman) or can be treated as rational goal-pursuers regardless of internal mechanism (Dennett). Philosophy emphasizes *reasons* for

action over mechanical causation, focusing on responsibility for actions and conceptual foundations rather than implementation.

0.4.2   Psychology: Self-Regulation and Control

Psychology approaches agency through **self-regulation**, **perceived control**, and contextual variability. Bakan's agency-communion spectrum characterizes fundamental modes of human existence, while Milgram (1974)'s agentic state describes conditions under which people suppress personal judgment and defer to authority. Bandura (1989)'s social cognitive theory emphasizes human agency as exercised through intentionality, forethought, self-reactiveness, and self-reflectiveness.

Psychology's distinctive insight: agency is not merely a property of isolated actions but a stable characteristic of cognitive systems capable of self-regulation. Crucially, agency exists on a spectrum and can be diminished by context (Milgram 1974) or enhanced through **self-efficacy** beliefs. For AI systems, this highlights how humans *perceive* AI agency, affecting trust and delegation decisions.

0.4.3   Law: Delegation and Fiduciary Duty

Legal doctrine understands agency through **delegated authority**, **fiduciary duties**, and **institutional contexts**. The Restatement of Agency defines the relationship as arising when a principal manifests assent that an agent shall act on the principal's behalf and subject to the principal's control. Key elements include: (1) relational structure requiring a principal-agent pair, (2) delegated authority derived from the principal, (3) fiduciary obligations of loyalty and care, and (4) liability attribution within scope of authority.

Law's distinctive approach focuses on external authority structures and legally enforceable duties rather than internal intentions or motivations. Legal frameworks embed agency in formal mechanisms—contracts, corporate charters, powers of attorney—that create, bound, and terminate authority relationships. For AI systems, this raises immediate questions: Can an AI be an agent in the legal sense? Who serves as principal? As illustrated in Section 0.1, misunderstanding these relationships can have serious consequences when systems lack mechanisms for fulfilling fiduciary duties.

0.4.4   Economics: Incentives and Information Asymmetry

Economic analysis frames agency through **utility alignment**, **information asymmetry**, and **organizational design**. Jensen and Meckling (1976)'s principal-agent model identifies the core problem: principals and agents have divergent preferences and asymmetric information. Agents

may pursue self-interest at principals' expense (**moral hazard**), or principals may struggle to assess agent quality ex ante (**adverse selection**).

Economics' distinctive approach emphasizes cost-benefit analysis and divergence of preferences rather than rights or duties. The focus is pragmatic: given that perfect alignment is impossible, how do we minimize **agency costs** through compensation structures, performance metrics, and oversight mechanisms? For AI systems, economic frameworks highlight alignment challenges: How do we design "reward functions" that elicit desired behavior? What monitoring mechanisms detect when AI agents pursue proxy metrics rather than true objectives?

0.4.5   Cognitive Science: Mental Architecture and Plans

Cognitive science approaches agency through **modular mental societies**, **plan-based control**, and **representational structures**. Minsky (1986)'s Society of Mind framed intelligence as emerging from interactions among simple processes without central control. Bratman (1987)'s planning theory emphasized that intentions are elements of partial plans structuring practical reasoning over time. The **BDI (Belief-Desire-Intention)** architecture implements this: agents maintain beliefs about the world, desires representing goals, and intentions as committed plans.

Cognitive science's distinctive insight treats agency as a lens on *mental architecture*, not just overt action, focusing on information processing, symbolic manipulation, and hierarchical plan structures. The framework readily generalizes from human to machine minds via abstraction. For AI systems, cognitive science provides architectural blueprints—the BDI framework has directly influenced agent programming languages, and concepts like partial plans and means-end reasoning transfer naturally to computational implementations.

0.4.6   Complex Systems: Emergence and Local Rules

Complex systems science characterizes agency through **local rules generating global effects**, **adaptation**, and **environmental coupling**. Holland (1995)'s complex adaptive systems, Epstein and Axtell (1996)'s agent-based models, and Kauffman (2000)'s autonomous agents emphasize that interesting behavior emerges from interactions among rule-following entities. Individual agents need not be sophisticated—simple local rules, diversity, and environmental feedback suffice to produce **emergent macro patterns**.

Complex systems' distinctive insight treats agents as embedded components of adaptive networks rather than isolated decision-makers. Behaviors co-evolve through selection pressures, learning, and resource flows. For AI systems, this perspective highlights how collections of simple agents exhibit sophisticated collective behavior (multi-agent systems, swarm robotics), but also warns against over-attributing agency to individual components when behavior primarily reflects system-level

dynamics.

0.4.7 Computer Science: Protocols and Mentalistic Abstractions

Computer science, particularly agent-oriented software engineering, frames agency through **mentalistic programming abstractions**, **interaction protocols**, and **organizational modeling**. Shoham (1993)'s agent-oriented programming introduced beliefs, commitments, and choices as first-class programming constructs. Unlike object-oriented programming (encapsulation and message-passing), agent-oriented programming treats components as having mental states and engaging in speech acts. FIPA agent communication languages formalized performatives (inform, request, propose) modeled on human **speech acts**.

Computer science's distinctive approach uses mentalistic vocabulary as an *engineering tool* rather than philosophical claim. Attributing beliefs to software agents improves code clarity and reasoning about distributed systems, whether or not agents "really" have beliefs. For AI systems, computer science provides the middleware—communication standards, coordination mechanisms, organizational structures—that enable multi-agent systems to function. Agents negotiate rather than merely receiving method calls.

0.4.8 Cross-Disciplinary Synthesis

Table 3 summarizes key differences across disciplines.

Discipline	Autonomy Level	Entity Frame	Key Mechanism	Primary Concern
Philosophy	Low–moderate	Human, extensible	Intention & reasons	Rational causality, responsibility
Psychology	Varies by context	Human-centered	Self-regulation & control	Lived experience, efficacy
Law	Low–moderate	Hybrid (principal-agent)	Authority & fiduciary duty	Liability, professional responsibility
Economics	Moderate	Institutional/hybrid	Incentive alignment	Agency costs, information asymmetry
Cognitive Science	Moderate–high	Human, generalizable	Plans & representations	Mental architecture
Complex Systems	Moderate–high	Machine/hybrid	Local rules & emergence	Collective behavior, adaptation
Computer Science	Moderate–high	Machine/hybrid	Protocols & coordination	System organization, communication
AI	High	Machine (+ hybrid modern)	Perception-action loops, tools	Task completion, capabilities

Table 3: Cross-disciplinary comparison of agency perspectives. Disciplines vary systematically in assumed autonomy level, entity framing, core mechanisms, and primary concerns.

Several patterns emerge:

- **Autonomy gradient**: Disciplines focusing on human-AI relationships (law, economics) assume lower autonomy, while those focused on machine capabilities (AI, complex systems) assume higher autonomy

- **Entity frames**: Philosophy, psychology, and cognitive science start with humans; law and economics examine hybrid relationships; complex systems, computer science, and AI embrace machine entities

- **Mechanism diversity**: Each discipline emphasizes different causal mechanisms—intentions, self-regulation, incentives, plans, rules, protocols, actions

- **Normative vs descriptive**: Philosophy and law are more normative (concerned with responsibility and duty), while AI and complex systems are more descriptive (concerned with observable behavior)

> **Integration Framework**
>
> For understanding agentic AI in legal contexts, we require synthesis across disciplines:
>
> - **Philosophy** provides conceptual foundations for attributing agency
> - **Psychology** illuminates how humans perceive and interact with AI agents
> - **Law** establishes liability frameworks and professional responsibilities
> - **Economics** offers tools for analyzing alignment problems
> - **Cognitive science** suggests architectural patterns
> - **Complex systems** warns about emergent behavior in multi-agent contexts
> - **Computer science** provides implementation protocols
> - **AI** defines practical capabilities and evaluation criteria

The next section extracts analytical dimensions that cut across these disciplinary perspectives.

0.5   Analytical Dimensions of Agency

While different disciplines approach agency from distinct theoretical foundations (Section 0.4), certain analytical dimensions cut across disciplinary boundaries. This section identifies four key dimensions—**autonomy**, **entity frames**, **goal dynamics**, and **persistence**—that structure variation in how agency is understood.

0.5.1 The Autonomy Spectrum

Perhaps the most fundamental dimension along which agent definitions vary is **autonomy spectrum**: the degree to which an agent can set its own agenda, select tactics, and own accountability for outcomes.

Three plain-language questions help position any definition on the autonomy spectrum:

> **Who sets the goal?**
>
> Does the agent accept goals from an external principal, or does it generate its own objectives? Delegated proxies follow instructions; self-directed systems set their own agendas.

> **Who decides the next step?**
>
> Does the agent execute predefined plans step-by-step, or does it choose tactics autonomously? Contract-bound delegates have tactical discretion; perception-action planners select actions independently.

> **Who carries responsibility if things go wrong?**
>
> Does accountability remain with the principal, or does the agent own outcomes? Low-autonomy agents are instruments; high-autonomy agents bear responsibility for their choices.

Definitions range from delegated proxies at one extreme to self-directed systems at the other.

Delegated Proxies (Low Autonomy). The Restatement (Second) of Agency (1958) exemplifies this end: agents execute their principal's instructions within bounded authority. Goals, evaluation criteria, and ultimate responsibility remain with the principal. Milgram (1974)'s "agentic state" captures the psychological correlate—individuals who see themselves as mere instruments of authority, disclaiming personal responsibility.

Contract-Bound Delegates (Low-Moderate Autonomy). Jensen and Meckling (1976)'s principal-agent model grants agents tactical discretion within incentive contracts but evaluates them solely on principals' payoffs. Agents choose means but not ends, operating under monitoring and compensation schemes designed to align behavior.

Self-Regulating Actors (Moderate Autonomy). Bandura (1989)'s human agency emphasizes that individuals set sub-goals, monitor progress, and self-adjust within social structures. Maes (1994)'s software agents learn user preferences and initiate actions based on high-level intent rather than step-by-step commands.

Perception-Action Planners (Moderate-High Autonomy). Russell and Norvig (1995)'s AI agents sense environments and select actions via performance-driven policies without waiting for commands. Franklin and Graesser (1997)'s autonomous agents pursue their own agendas over time, acting to influence future perceptions.

Learning Loop Owners (High Autonomy). Sutton and Barto (1998)'s reinforcement learning agents experiment with actions and update policies from reward signals without human step-by-step guidance. The agent owns the exploration-exploitation tradeoff.

Tool Orchestrators (Very High Autonomy). Modern tool-use APIs (2025)—whether implemented through AI/ML or traditional logic—choose when and how to invoke tools, integrating outputs into their reasoning loops. Contemporary LLM frameworks provide one implementation approach for this architectural pattern, "independently accomplish[ing] tasks on behalf of users," planning, sequencing, and completing multi-step workflows before reporting back.

Table 4 summarizes this progression.

Position	Source	Description
Delegated proxy	American Law Institute (2006)	Follows instructions; principal retains liability
Obedience	Milgram (1974)	Suppresses personal goals under authority
Contract-bound	Jensen and Meckling (1976)	Chooses tactics within contracts
Self-regulating	Bandura (1989)	Sets goals, monitors, adjusts
Adaptive assistant	Maes (1994)	Learns preferences, initiates actions
Perception-action	Russell and Norvig (1995)	Senses environment, executes policy
Agenda-setting	Franklin and Graesser (1997)	Maintains independent agenda
Learning loop	Sutton and Barto (1998)	Experiments, updates from rewards
Tool orchestrator	Yao et al. (2022)	Chooses tools to invoke
Independent finisher	Xi et al. (2023)	Plans and completes tasks

Table 4: Autonomy spectrum: from delegated proxies to self-directed systems.

This dimension correlates with but differs from technological capability. Rule-based expert systems can exhibit moderate autonomy (goal-directed loops), while sophisticated neural networks operating in single-shot mode exhibit none. Autonomy is architectural, not merely a function of model sophistication.

0.5.2 Entity Frames

Agent definitions often talk past each other because they quietly assume different kinds of **entities**. We identify three clusters:

Human-Centered. These definitions privilege individual humans and their cognition or morality. Anscombe (1957)'s intentional action explores agency as practical knowledge held by persons. Milgram (1974)'s agentic state examines how people respond to authority. Bandura (1989)'s human agency emphasizes internal cognitive machinery—intentionality, forethought, self-reactiveness, self-reflectiveness. The hallmark questions: *What intentions animate the person? How is responsibility assigned?*

Institutional/Hybrid. These definitions situate agency in socio-technical collectives—humans embedded in contracts, firms, or organizational structures. The Restatement of Agency (1958) defines the principal-agent pair, not isolated individuals. Jensen and Meckling (1976)'s economic model examines "manager-with-contract" as the effective actor. Epstein and Axtell (1996)'s agent-based models simulate rule-following individuals shaped by local interactions. The hallmark questions: *What structure channels decisions? How are incentives and authority shared?*

Machine-Centered. These definitions assign agency to computational systems managing perception-action loops. Russell and Norvig (1995)'s AI agents can be any system perceiving through sensors and acting via actuators. Franklin and Graesser (1997)'s autonomous agents are persistent computational systems pursuing agendas. Contemporary LLM frameworks (2025) are explicitly software coordinating tools. The hallmark questions: *How does the system sense, plan, and act? What loop keeps it going?*

Table 5 summarizes representative examples.

Frame	Source	Description
Human	Anscombe (1957)	Human minds, intentions
	Milgram (1974)	
	Bandura (1989)	
Institutional	American Law Institute (2006)	Organizations, authority structures
	Jensen and Meckling (1976)	
	Epstein and Axtell (1996)	
Machine	Russell and Norvig (1995)	Computational systems, loops
	Franklin and Graesser (1997)	
	Xi et al. (2023)	

Table 5: Entity frames: human minds, institutional relationships, or computational systems.

Importantly, these frames are not mutually exclusive. Contemporary "agentic AI" often involves

hybrid entities: AI systems acting on behalf of human principals, with goals supplied by humans but tactics chosen by machines. The legal and ethical challenges arise precisely at these boundaries.

0.5.3   Goal Dynamics

Definitions also vary in how they understand the **relationship between agents and goals**. We identify three stances:

Goal Acceptance. The agent receives a mandate and optimizes for it without debate. The Restatement of Agency (1958) requires agents to follow principal objectives; divergence invites breach. Sutton and Barto (1998)'s RL agent maximizes a fixed reward function supplied externally. Contemporary LLM frameworks (2025) "independently accomplish tasks on behalf of users," but user goals remain the north star. Success equals compliance or reward maximization.

Goal Adaptation. The agent refines, reprioritizes, or balances goals within constraints. Maes (1994)'s software agents learn user preferences and decide when to intervene, reprioritizing information flows. Russell and Norvig (1995)'s agents balance performance measures, trading off actions based on context. Modern tool-use APIs (2025) interpret instructions, break them down, and choose which tools serve user intent. The agent has discretion in interpretation and tactics.

Goal Negotiation. The agent co-determines objectives with others, often through communication. Weiss (1999)'s multi-agent systems coordinate, cooperate, and negotiate when goals conflict. Jennings et al. (1998)'s AOSE roadmap treats agents as social entities that negotiate commitments and allocate tasks. Multi-agent frameworks (2025) enable multi-agent applications where agents message each other to propose plans, critique, and converge collaboratively.

Table 6 summarizes this dimension.

This dimension has direct implications for legal accountability. If an AI agent merely accepts goals, responsibility flows clearly to whoever set them. If it adapts goals through interpretation, questions arise about faithful execution. If it negotiates goals with other agents or humans, traditional principal-agent frameworks may not apply cleanly.

0.5.4   Persistence and Embodiment

Two additional dimensions merit brief mention:

Stance	Source	Description
Acceptance	American Law Institute (2006)	Follows supplied objectives
	Sutton and Barto (1998)	
	Xi et al. (2023)	
Adaptation	Maes (1994)	Refines goals, chooses tactics
	Russell and Norvig (1995)	
	Yao et al. (2022)	
Negotiation	Weiss (1999)	Co-determines objectives
	Jennings et al. (1998)	

Table 6: Goal dynamics: from acceptance through adaptation to negotiation.

Temporal Persistence. Franklin and Graesser (1997) distinguish programs from agents partly on **persistence**—agents maintain state and pursue objectives over extended periods. This contrasts with reactive systems that respond to inputs without memory. Modern LLM agents incorporate conversation history and tool-use results, creating persistence within sessions. As discussed in Section 0.1, single-shot LLM usage lacks the persistent goal pursuit that characterizes agentic systems.

Embodiment. Kauffman (2000)'s autonomous agents require physical **embodiment** capable of performing thermodynamic work cycles. This biological grounding contrasts sharply with purely virtual agents. Robotics and embodied AI represent one implementation; cloud-based LLM agents represent another. Whether virtual agents merit the term "agent" in the fullest sense remains philosophically contentious, though pragmatically settled in favor of inclusion.

0.5.5 Implications for Agentic AI

These four dimensions—autonomy, entity frames, goal dynamics, persistence—provide organizing principles for evaluating contemporary systems.

> **Critical Questions for Evaluating "Agentic" Systems**
>
> When someone claims a system is "agentic," we should ask:
> - **Autonomy:** How much discretion does it have in selecting goals and tactics? Where does accountability reside?
> - **Entity:** Is this a pure machine system, a human-AI hybrid, or an institutional arrangement?
> - **Goals:** Does it accept goals passively, adapt them through interpretation, or negotiate them with other agents?

> • **Persistence:** Does it maintain state and pursue objectives over time, or operate in single-shot mode?

These questions cut across the marketing hype and disciplinary jargon, providing a shared vocabulary for more precise discourse. The next section synthesizes these insights into a working definition grounded in theoretical foundations.

0.6   Analytical Framework

Having traced the historical evolution (Section 0.3), examined disciplinary perspectives (Section 0.4), and identified analytical dimensions (Section 0.5), we now synthesize these insights into a theoretical framework for analyzing agentic systems in legal contexts.

0.6.1   Cross-Cutting Patterns

Despite disciplinary diversity, several patterns emerge consistently:

> **Cross-Cutting Patterns**
>
> **Goal-Directedness.** Nearly every definition involves goals, objectives, or performance measures. Whether expressed as intentions (philosophy), reward functions (AI), mandates (law), or incentive structures (economics), agents are distinguished by oriented behavior toward outcomes.
>
> **Perception-Action Coupling.** From Russell and Norvig (2020)'s sensors and actuators to Bratman (1987)'s practical reasoning to Bandura (1989)'s self-reactiveness, agents observe their environments and take actions. Single-shot transformations (compilers, converters) fall outside this pattern; iterative loops characterize agentic systems.
>
> **Selective Autonomy.** While autonomy levels vary (Section 0.5), agency requires some discretion. Pure puppets executing scripted commands without choice do not count. The threshold varies—legal agents have minimal discretion, AI agents have substantial discretion—but the requirement persists.
>
> **Stopping Conditions.** Implicit or explicit, operational definitions assume systems eventually complete tasks or reach terminal states. Unbounded processes without convergence criteria—infinite loops, perpetual monitoring without decision points—lack the bounded purposefulness that characterizes agentic systems.

0.6.2   Theoretical Foundations

The cross-cutting patterns observed across disciplines build on rich theoretical traditions:

Classical Planning (AI). Goal-directedness and means-end reasoning connect to STRIPS-style planning, where agents decompose goals into subgoals and select actions to achieve preconditions. Legal tasks like due diligence map naturally to hierarchical goal structures.

Reactive Architectures (Robotics). The **perception-action coupling** loop reflects Brooks' subsumption architecture (layered reactive control) and behavior-based robotics. Agents need not maintain complex world models—tight coupling between sensing and acting suffices for many tasks.

BDI Models (Cognitive Science, Computer Science). Beliefs (about the world), Desires (goals to achieve), and Intentions (committed plans) provide one influential formal model. BDI architectures persist in agent-oriented programming and explain how agents manage competing objectives.

Reinforcement Learning (AI). The agent-environment interaction framework, where agents select actions, receive observations and rewards, and update policies, implements the iterative loop. RL formalizes learning from experience.

Intentional Stance (Philosophy). Dennett (1987)'s pragmatic criterion—treat systems as agents when attributing mental states yields reliable predictions—provides a philosophical justification. We adopt the intentional stance toward systems that exhibit goal-directed behavior.

0.6.3 Application to Professional Contexts

The analytical framework establishes what makes something an agent in general. Professional deployment in legal, medical, or financial contexts requires additional governance safeguards beyond these core properties—attribution mechanisms for tracing claims to sources, provenance tracking for auditable reasoning chains, escalation protocols for recognizing limitations, and confidentiality enforcement for maintaining information barriers. These capability-centric refinements are addressed in a companion chapter on governance and risk management for agentic systems.

0.6.4 Limitations and Boundary Cases

The analytical framework inevitably confronts boundary cases:

Minimal Autonomy. Thermostats qualify as agents under our definition, yet have trivial autonomy. We accept this inclusion—thermostats exhibit goal-directedness, sensing, acting, and looping. Modern smart thermostats include explicit termination logic (time windows, energy budgets), while basic thermostats rely on implicit goal satisfaction through continuous cycling. We acknowledge that interesting agency questions arise at higher autonomy levels.

Emergent vs Individual Agency. In multi-agent systems, is agency located in individuals or collectives? Both, potentially. Individual agents exhibit goal-directed behavior; the collective may exhibit emergent properties not reducible to individuals. Legal frameworks struggle with this

(corporate vs individual liability), and AI systems inherit these tensions.

Virtual vs Embodied. Kauffman (2000)'s thermodynamic criteria exclude purely virtual agents. We side with the computational traditions that accept virtual agents, but acknowledge philosophical objections remain valid.

Single-Shot LLM Responses. As discussed in Section 0.2, single-shot responses lack the iteration and adaptation required for agentic systems. However, if a response includes internal tool calls, observations, and updated reasoning within a single interaction, it may exhibit agentic properties architecturally even if perceived as one response by the user.

0.6.5   Looking Forward

The analytical framework and theoretical foundations provide stable conceptual tools during rapid technological change. As LLMs grow more capable, as multi-agent systems proliferate, and as AI-human collaboration evolves, these cross-disciplinary patterns help us:

- Distinguish genuine architectural innovations from repackaged capabilities

- Maintain cross-disciplinary communication through shared vocabulary

- Evaluate new systems against consistent criteria

- Anticipate legal and ethical challenges based on autonomy levels and entity frames

A companion chapter examines how these theoretical foundations manifest in contemporary AI systems and frameworks.

0.6.6   Consistency Checks (Informal)

To keep the definitions precise and testable, we list simple assumptions and what follows from them. These are lightweight and engineering-oriented.

> **Definition Consistency**
>
> **Basics we assume.** - A1 (Minimal agency). An **agent** has *Goal*, *Perception*, and *Action*. - A2 (Operational addition). An **agentic system** is an agent with *Iteration*, *Adaptation*, and *Termination* (explicit or implicit). - A3 (Implementation is orthogonal). **AI-powered** is an implementation choice (e.g., LLMs), not a property definition.
>
> **Immediate consequences.** - C1. Every agentic system is an agent (A2 → A1). - C2. Not every agent is an agentic system (e.g., one-shot systems that perceive and act once). - C3. An **AI agent** is an *agentic system* implemented with AI/ML (A3); therefore AI agent → agentic system → agent. - C4. We may describe features as "agentic" (behavior/properties)

without implying full six-of-six conformance; reserve "agentic system" for six-of-six. - C5. Termination includes implicit goal satisfaction (e.g., thermostat at target) or explicit controls (budgets, time windows, escalation). Continuing tasks qualify when they use bounded episodes or quiescent checks rather than unbounded loops.

Avoiding confusion. - P1. Properties (what a system *is*) are distinct from implementation (how it is *built*). - P2. Single-shot LLM outputs can count as *agents* only when they instantiate Goal, Perception, and Action within the episode; they are *not* agentic systems without iteration/adaptation/termination. - P3. Tool use alone does not imply agency; iterative, adaptive use toward goals does.

0.6.7 Common Questions

Is "Iteration" just repeated Action?

No. Level 1 allows a single perceive–act cycle. Iteration requires multiple cycles with state carried forward, enabling error correction and pursuit across steps.

Do continuous control tasks fail Termination?

No. We accept implicit termination (goal satisfaction) or explicit governance (budgets, episodes, escalation). Engineered deployments typically bound horizons even for continuing tasks.

If thermostats are agents, doesn't that make everything an agent?

No. Many systems lack at least one of Goal, Perception, Action (e.g., compilers lack goals; pipelines often lack perception of results). Minimal inclusion does not trivialize the category.

Is "AI agent" just marketing?

Here it has a falsifiable meaning: an agentic system (six properties) whose planning/orchestration is implemented with AI/ML. Single-shot chat completion is excluded by construction.

0.7 Conclusion

This chapter traced how eight disciplines—philosophy, psychology, law, economics, cognitive science, complex systems, computer science, and AI—approach agency from different angles. Rather than forcing consensus, we extracted practical frameworks that respect this diversity while enabling clear evaluation.

0.7.1 Key Takeaways

The three-level hierarchy. An **agent** requires three minimal properties (goals, perception, action). An **agentic system** adds three more for operational deployment (iteration, adaptation, termination). **Agentic AI** implements these six properties through artificial intelligence rather than traditional programming. This hierarchy provides clear criteria for evaluating any system.

The six-question test. Does it have goals? Does it perceive? Does it act? Does it iterate? Does it adapt? Does it stop? Systems meeting all six criteria qualify as agents in professional contexts. This rubric translates theory into practical evaluation.

Four analytical dimensions. The autonomy spectrum (delegated to self-directed), entity frames (human to institutional to machine), goal dynamics (acceptance to negotiation), and persistence requirements structure how disciplines define agency differently. These dimensions enable precise discussion of where definitions agree and diverge.

Disciplinary diversity matters. Eight disciplines—philosophy, psychology, law, economics, cognitive science, complex systems, computer science, and AI—each emphasize different aspects. No single perspective captures the whole. Understanding this diversity prevents talking past each other and enables productive synthesis.

Historical convergence on tools-in-a-loop. Early definitions (1950s-1980s) focused on humans. The 1990s broadened to pure machines. The 2020s LLM era converged on "tools-in-a-loop" architectures: Willison (2025)'s synthesis captures the emerging consensus across research and industry.

Professional applications require additional safeguards. Legal and financial contexts demand attribution to sources, auditable provenance, escalation protocols, and confidentiality mechanisms. These don't change what makes something an agent—they define what makes an agent suitable for high-stakes work.

0.7.2 The Framework

From this disciplinary diversity, we extracted a practical three-level framework. An **agent** requires three minimal properties: goals, perception, and action. This baseline applies to humans, organizations, thermostats, and computational systems alike. An **agentic system** adds iteration, adaptation, and termination—the six properties needed for reliable operational deployment in professional contexts. **Agentic AI** refers to systems implementing these six properties through artificial intelligence rather than traditional programming. This hierarchy provides scaffolding for evaluating real systems against clear criteria.

The six-question evaluation rubric translates theory into practice: Does it have goals? Does it perceive? Does it act? Does it iterate? Does it adapt? Does it stop? Systems meeting all six criteria

qualify as agents in professional contexts. Four analytical dimensions—the autonomy spectrum (delegated to self-directed), entity frames (human to machine), goal dynamics (acceptance to negotiation), and persistence requirements—provide organizing principles for comparing definitions and understanding where disagreements arise.

0.7.3 Historical Perspective

The historical evolution we traced reveals both convergence and persistent tensions. Early definitions (1950s-1980s) focused on humans or human-AI relationships, rooted in philosophy, psychology, law, and economics. The 1990s computational revolution broadened entity frames to include pure machines, driven by distributed systems, the internet, and advances in AI. The LLM era (2020s onward) has simultaneously sharpened around "tools-in-a-loop" architectures while introducing hybrid framings where humans provide goals and oversight while AI executes multi-step tasks.

Willison (2025)'s practitioner synthesis—"LLM agent runs tools in a loop to achieve a goal"—captures the emerging consensus that spans research frameworks (Yao et al. 2022), commercial implementations (LangChain, LlamaIndex, vendor APIs), and operational definitions across the AI engineering community. While not fully standardized, this architectural convergence provides stable foundations for current practice.

0.7.4 Professional Application

For professional applications in law and finance, general agency definitions must be augmented with domain-specific requirements: attribution to authoritative sources, auditable provenance, escalation protocols recognizing human judgment needs, and confidentiality mechanisms protecting privilege. These requirements don't change what makes something an agent; they define what makes an agent suitable for high-stakes professional work. Subsequent chapters address these professional safeguards in detail.

As agentic AI systems become more capable and widely deployed in professional domains, definitional clarity matters. When a legal AI makes an error, who is responsible? When an agent escalates inappropriately, what went wrong? When capabilities exceed oversight capacity, how do we ensure accountability? These questions require shared vocabulary for discussing autonomy levels, entity frames, goal dynamics, and architectural properties.

0.7.5 Connections and Further Reading

This foundational chapter connects to several companion chapters in this textbook. The governance chapter explores professional safeguards—attribution, provenance, escalation, and confidentiality—in depth. The architectures chapter details contemporary patterns like ReAct (reasoning and acting) and Reflexion (self-reflection). The evaluation chapter addresses capability levels and comprehensive benchmarks. The regulation chapter analyzes frameworks like the EU AI Act and NIST guidance.

Implementation chapters provide practical guidance for deployment in legal practice and financial services.

For readers seeking deeper engagement with specific disciplinary traditions, we recommend: Bratman (1987) for the philosophical BDI foundation; Bandura (1989) for psychological perspectives on human agency; American Law Institute (2006) for legal foundations; Jensen and Meckling (1976) for economic principal-agent theory; Russell and Norvig (2010) for comprehensive AI coverage; Wooldridge (2009) for multi-agent systems; and Xi et al. (2023) for contemporary LLM-based agents.

0.7.6 Final Thoughts

The intellectual history traced in this chapter—from Anscombe (1957)'s practical knowledge through Russell and Norvig (2020)'s perception-action loops to Willison (2025)'s tools-in-a-loop consensus—provides stable conceptual foundations for navigating rapid technological change. The concepts are old; the technologies are new; the challenge is synthesis.

The journey from "driving cattle" to "pleading a case" (both senses of Latin *agō*) to "running tools in a loop" spans millennia of human intellectual development and decades of computational innovation. These conceptual foundations continue through the remaining chapters of this textbook and into your professional practice.

Bibliography

American Law Institute (1958). *Restatement of the Law, Second, Agency*. St Paul, MN: American Law Institute Publishers.

American Law Institute (2006). *Restatement of the Law Third, Agency*. St Paul, MN: American Law Institute Publishers.

Anscombe, G. E. M. (1957). *Intention*. Oxford: Basil Blackwell.

Bandura, Albert (1989). "Human Agency in Social Cognitive Theory". In: *American Psychologist* 44.9, pp. 1175–1184.

Bonabeau, Eric (2002). "Agent-Based Modeling: Methods and Techniques for Simulating Human Systems". In: *Proceedings of the National Academy of Sciences* 99.suppl 3, pp. 7280–7287. DOI: 10.1073/pnas.082080899.

Bratman, Michael E. (1987). *Intention, Plans, and Practical Reason*. Foundational work establishing the BDI (Belief-Desire-Intention) model of practical reasoning. Cambridge, MA: Harvard University Press. ISBN: 978-0674003989.

Davidson, Donald (1963). "Actions, Reasons, and Causes". In: *Journal of Philosophy* 60, pp. 685–700. DOI: 10.2307/2023177.

Dennett, Daniel C. (1987). *The Intentional Stance*. Cambridge, MA: MIT Press.

Epstein, Joshua M. (2006). *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton University Press.

Epstein, Joshua M. and Robert Axtell (1996). *Growing Artificial Societies: Social Science from the Bottom Up*. Cambridge, MA: MIT Press.

Ferber, Jacques (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley.

Franklin, Stan and Art Graesser (1997). "Is It an Agent, or Just a Program? A Taxonomy for Autonomous Agents". In: *Intelligent Agents III (ATAL'96)*. Vol. 1193. Lecture Notes in Computer Science. Springer, pp. 21–35. DOI: 10.1007/BFb0013570.

Giddens, Anthony (1984). *The Constitution of Society: Outline of the Theory of Structuration*. University of California Press.

Holland, John H. (1995). *Hidden Order: How Adaptation Builds Complexity*. Addison-Wesley.

Jennings, Nicholas R., Katia Sycara, and Michael Wooldridge (1998). "A Roadmap of Agent Research and Development". In: *Autonomous Agents and Multi-Agent Systems* 1.1, pp. 7–38. DOI: 10.1023/A:1010090405266.

Jensen, Michael C. and William H. Meckling (1976). "Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure". In: *Journal of Financial Economics* 3.4, pp. 305–360.

Kauffman, Stuart A. (2000). *Investigations*. Oxford University Press.

LangChain Documentation (2024). *Agents*. URL: https://python.langchain.com/docs/modules/agents/ (visited on 10/26/2025).

Lewis, Charlton T. and Charles Short, eds. (1879). *A Latin Dictionary. Founded on Andrews' Edition of Freund's Latin Dictionary*. Perseus Digital Library. Oxford: Clarendon Press. URL: https://www.perseus.tufts.edu/hopper/text?doc=Perseus:text:1999.04.0059 (visited on 10/26/2025).

Maes, Pattie (1994). "Agents that Reduce Work and Information Overload". In: *Communications of the ACM* 37.7, pp. 30–40. DOI: 10.1145/176789.176792.

Milgram, Stanley (1974). *Obedience to Authority: An Experimental View*. Harper & Row.

Minsky, Marvin (1986). *The Society of Mind*. Simon & Schuster.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, et al. (2015). "Human-level control through deep reinforcement learning". In: *Nature* 518, pp. 529–533. DOI: 10.1038/nature14236.

OpenAI (2024). *OpenAI Agents SDK*. URL: https://openai.github.io/openai-agents-python/ (visited on 10/31/2025).

Park, Joon Sung, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein (2023). "Generative Agents: Interactive Simulacra of Human Behavior". In: arXiv: 2304.03442 [cs.HC]. URL: https://arxiv.org/abs/2304.03442 (visited on 10/27/2025).

Russell, Stuart J. and Peter Norvig (1995). *Artificial Intelligence: A Modern Approach*. 1st ed. Prentice Hall.

Russell, Stuart J. and Peter Norvig (2010). *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall.

Russell, Stuart J. and Peter Norvig (2020). *Artificial Intelligence: A Modern Approach*. 4th ed. Pearson. URL: http://aima.cs.berkeley.edu/ (visited on 10/27/2025).

Schlosser, Markus (2019). "Agency". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Metaphysics Research Lab, Stanford University. URL: https://plato.stanford.edu/archives/fall2019/entries/agency/ (visited on 10/27/2025).

Shoham, Yoav (1993). "Agent-Oriented Programming". In: *Artificial Intelligence* 60.1, pp. 51–92. DOI: 10.1016/0004-3702(93)90034-9.

Silver, David, Aja Huang, Chris J. Maddison, et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529, pp. 484–489. DOI: 10.1038/nature16961.

Sutton, Richard S. and Andrew G. Barto (1998). *Reinforcement Learning: An Introduction*. MIT Press.

US lawyer sanctioned for using ChatGPT to make up case citations (2023). Reuters coverage of Mata v. Avianca sanctions. URL: https://www.reuters.com/world/us/us-lawyer-who-used-chatgpt-made-up-cases-sanctioned-2023-06-23/ (visited on 10/25/2025).

Weiss, Gerhard (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* MIT Press.

Willison, Simon (Sept. 2025). *I think "agent" may finally have a widely enough agreed upon definition to be useful jargon now.* Practitioner definition: tools-in-a-loop to achieve a goal. URL: https://simonwillison.net/2025/Sep/18/agents/ (visited on 10/26/2025).

Wilson, George and Samuel Shpall (2022). "Action". In: *The Stanford Encyclopedia of Philosophy.* Ed. by Edward N. Zalta and Uri Nodelman. Metaphysics Research Lab, Stanford University. URL: https://plato.stanford.edu/archives/spr2022/entries/action/ (visited on 10/27/2025).

Wooldridge, Michael (2009). *An Introduction to MultiAgent Systems.* 2nd ed. John Wiley & Sons.

Wooldridge, Michael and Nicholas R. Jennings (1995). "Intelligent Agents: Theory and Practice". In: *The Knowledge Engineering Review* 10.2, pp. 115–152. DOI: 10.1017/S0269888900008122. URL: https://www.cambridge.org/core/journals/knowledge-engineering-review/article/intelligent-agents-theory-and-practice/9C5FD9D5B6B93AFC0E7E5A0C9F6A4E82 (visited on 10/27/2025).

Xi, Zhiheng et al. (2023). "The Rise and Potential of Large Language Model Based Agents: A Survey". In: *arXiv preprint arXiv:2309.07864.*

Yao, Shunyu et al. (2022). "ReAct: Synergizing Reasoning and Acting in Language Models". In: arXiv: 2210.03629 [cs.AI]. URL: https://arxiv.org/abs/2210.03629 (visited on 10/25/2025).

Bibliography

American Law Institute (1958). *Restatement of the Law, Second, Agency*. St Paul, MN: American Law Institute Publishers.

American Law Institute (2006). *Restatement of the Law Third, Agency*. St Paul, MN: American Law Institute Publishers.

Anscombe, G. E. M. (1957). *Intention*. Oxford: Basil Blackwell.

Bandura, Albert (1989). "Human Agency in Social Cognitive Theory". In: *American Psychologist* 44.9, pp. 1175–1184.

Bonabeau, Eric (2002). "Agent-Based Modeling: Methods and Techniques for Simulating Human Systems". In: *Proceedings of the National Academy of Sciences* 99.suppl 3, pp. 7280–7287. DOI: 10.1073/pnas.082080899.

Bratman, Michael E. (1987). *Intention, Plans, and Practical Reason*. Foundational work establishing the BDI (Belief-Desire-Intention) model of practical reasoning. Cambridge, MA: Harvard University Press. ISBN: 978-0674003989.

Davidson, Donald (1963). "Actions, Reasons, and Causes". In: *Journal of Philosophy* 60, pp. 685–700. DOI: 10.2307/2023177.

Dennett, Daniel C. (1987). *The Intentional Stance*. Cambridge, MA: MIT Press.

Epstein, Joshua M. (2006). *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton University Press.

Epstein, Joshua M. and Robert Axtell (1996). *Growing Artificial Societies: Social Science from the Bottom Up*. Cambridge, MA: MIT Press.

Ferber, Jacques (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley.

Franklin, Stan and Art Graesser (1997). "Is It an Agent, or Just a Program? A Taxonomy for Autonomous Agents". In: *Intelligent Agents III (ATAL'96)*. Vol. 1193. Lecture Notes in Computer Science. Springer, pp. 21–35. DOI: 10.1007/BFb0013570.

Giddens, Anthony (1984). *The Constitution of Society: Outline of the Theory of Structuration*. University of California Press.

Holland, John H. (1995). *Hidden Order: How Adaptation Builds Complexity*. Addison-Wesley.

Jennings, Nicholas R., Katia Sycara, and Michael Wooldridge (1998). "A Roadmap of Agent Research and Development". In: *Autonomous Agents and Multi-Agent Systems* 1.1, pp. 7–38. DOI: 10.1023/A:1010090405266.

Jensen, Michael C. and William H. Meckling (1976). "Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure". In: *Journal of Financial Economics* 3.4, pp. 305–360.

Kauffman, Stuart A. (2000). *Investigations*. Oxford University Press.

LangChain Documentation (2024). *Agents*. URL: https://python.langchain.com/docs/modules/agents/ (visited on 10/26/2025).

Lewis, Charlton T. and Charles Short, eds. (1879). *A Latin Dictionary. Founded on Andrews' Edition of Freund's Latin Dictionary*. Perseus Digital Library. Oxford: Clarendon Press. URL: https://www.perseus.tufts.edu/hopper/text?doc=Perseus:text:1999.04.0059 (visited on 10/26/2025).

Maes, Pattie (1994). "Agents that Reduce Work and Information Overload". In: *Communications of the ACM* 37.7, pp. 30–40. DOI: 10.1145/176789.176792.

Milgram, Stanley (1974). *Obedience to Authority: An Experimental View*. Harper & Row.

Minsky, Marvin (1986). *The Society of Mind*. Simon & Schuster.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, et al. (2015). "Human-level control through deep reinforcement learning". In: *Nature* 518, pp. 529–533. DOI: 10.1038/nature14236.

OpenAI (2024). *OpenAI Agents SDK*. URL: https://openai.github.io/openai-agents-python/ (visited on 10/31/2025).

Park, Joon Sung, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein (2023). "Generative Agents: Interactive Simulacra of Human Behavior". In: arXiv: 2304.03442 [cs.HC]. URL: https://arxiv.org/abs/2304.03442 (visited on 10/27/2025).

Russell, Stuart J. and Peter Norvig (1995). *Artificial Intelligence: A Modern Approach*. 1st ed. Prentice Hall.

Russell, Stuart J. and Peter Norvig (2010). *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall.

Russell, Stuart J. and Peter Norvig (2020). *Artificial Intelligence: A Modern Approach*. 4th ed. Pearson. URL: http://aima.cs.berkeley.edu/ (visited on 10/27/2025).

Schlosser, Markus (2019). "Agency". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Metaphysics Research Lab, Stanford University. URL: https://plato.stanford.edu/archives/fall2019/entries/agency/ (visited on 10/27/2025).

Shoham, Yoav (1993). "Agent-Oriented Programming". In: *Artificial Intelligence* 60.1, pp. 51–92. DOI: 10.1016/0004-3702(93)90034-9.

Silver, David, Aja Huang, Chris J. Maddison, et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529, pp. 484–489. DOI: 10.1038/nature16961.

Sutton, Richard S. and Andrew G. Barto (1998). *Reinforcement Learning: An Introduction*. MIT Press.

US lawyer sanctioned for using ChatGPT to make up case citations (2023). Reuters coverage of Mata v. Avianca sanctions. URL: https://www.reuters.com/world/us/us-lawyer-who-used-chatgpt-made-up-cases-sanctioned-2023-06-23/ (visited on 10/25/2025).

Weiss, Gerhard (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* MIT Press.

Willison, Simon (Sept. 2025). *I think "agent" may finally have a widely enough agreed upon definition to be useful jargon now.* Practitioner definition: tools-in-a-loop to achieve a goal. URL: https://simonwillison.net/2025/Sep/18/agents/ (visited on 10/26/2025).

Wilson, George and Samuel Shpall (2022). "Action". In: *The Stanford Encyclopedia of Philosophy.* Ed. by Edward N. Zalta and Uri Nodelman. Metaphysics Research Lab, Stanford University. URL: https://plato.stanford.edu/archives/spr2022/entries/action/ (visited on 10/27/2025).

Wooldridge, Michael (2009). *An Introduction to MultiAgent Systems.* 2nd ed. John Wiley & Sons.

Wooldridge, Michael and Nicholas R. Jennings (1995). "Intelligent Agents: Theory and Practice". In: *The Knowledge Engineering Review* 10.2, pp. 115–152. DOI: 10.1017/S0269888900008122. URL: https://www.cambridge.org/core/journals/knowledge-engineering-review/article/intelligent-agents-theory-and-practice/9C5FD9D5B6B93AFC0E7E5A0C9F6A4E82 (visited on 10/27/2025).

Xi, Zhiheng et al. (2023). "The Rise and Potential of Large Language Model Based Agents: A Survey". In: *arXiv preprint arXiv:2309.07864.*

Yao, Shunyu et al. (2022). "ReAct: Synergizing Reasoning and Acting in Language Models". In: arXiv: 2210.03629 [cs.AI]. URL: https://arxiv.org/abs/2210.03629 (visited on 10/25/2025).