Agentic Design Patterns: A System-Theoretic Framework

Minh-Dung Dao [*] [†]
University College Cork
College Rd, Cork, Ireland T12 K8AF
123122658@umail.ucc.ie

Quy Minh Le [*] [†]
Vietnam National University
Xuan Thuy St, Hanoi, Vietnam 10000
22028190@vnu.edu.vn

Hoang Thanh Lam
IBM Research Ireland
182 Pearse Street, Dublin 2, Ireland D02 F6N2

Duc-Trong Le
Vietnam National University
Xuan Thuy St, Hanoi, Vietnam 10000

Quoc-Viet Pham
Trinity College Dublin
Dublin 2, Ireland D02 W272

Barry O'Sullivan
University College Cork
College Rd, Cork, Ireland T12 K8AF

Hoang D. Nguyen [†]
University College Cork
College Rd, Cork, Ireland T12 K8AF
hn@cs.ucc.ie

Abstract

With the development of foundation model (FM), agentic AI systems are getting more attention, yet their inherent issues like hallucination and poor reasoning, coupled with the frequent ad-hoc nature of system design, lead to unreliable and brittle applications. Existing efforts to characterise agentic design patterns often lack a rigorous systems-theoretic foundation, resulting in high-level or convenience-based taxonomies that are difficult to implement. This paper addresses this gap by introducing a principled methodology for engineering robust AI agents. We propose two primary contributions: first, a novel system-theoretic framework that deconstructs an agentic AI system into five core, interacting functional subsystems: Reasoning & World Model, Perception & Grounding, Action Execution, Learning & Adaptation, and Inter-Agent Communication. Second, derived from this architecture and directly mapped to a comprehensive taxonomy of agentic challenges, we present a collection of 12 agentic design patterns. These patterns — categorised as Foundational, Cognitive & Decisional, Execution & Interaction, and Adaptive & Learning — offer reusable, structural solutions to recurring problems in agent design. The utility of the framework is demonstrated by a case study on the ReAct framework, showing how the proposed patterns can rectify systemic architectural deficiencies. This work provides a foundational language and a structured methodology to standardise agentic design communication among researchers and engineers, leading to more modular, understandable, and reliable autonomous systems.

[*] Equal contribution.

[†] Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: LAW 2025: Bridging Language, Agent, and World Models for Reasoning and Planning.

1 Introduction

Foundation model (FM) creates a revolution in Artificial Intelligence (AI); AI systems can demonstrate behaviours reminiscent of natural entities with cognitive skills, such as remembering, reasoning, thinking and writing creatively Naveed et al. [2025]. They have enabled a wide variety of applications in different fields and changed the paradigm of research on intelligent systems. However, FMs face various problems that hinder their capabilities and usefulness for practical applications, such as catastrophic forgetting, hallucination, bias, and incapacity of slow thinking Kaddour et al. [2023]. Recently, there has been a growing interest from both industry and academia on agentic AI systems Acharya et al. [2025], with FMs at their cores and equipped with external tools (e.g., web searching and code execution) and abilities (e.g., memorising and planning). This approach allows systems to tackle difficult problems, interact with external environments, and make decisions with a certain level of autonomy. Moreover, a combination of many intelligent agents that interact with each other following certain structures, strategies, and coordination protocols creates a multi-agent system (MAS), in which agents can communicate, share information, orchestrate, and act toward a set of collective goals Tran et al. [2025]. These ideas about agentic systems and MAS originate from individual and collective human intelligence in society, enabled with equipments and collaborative mechanisms to perform different activities from daily routines to complex scientific thinking.

There have been several different attempts to formulate an encompassing definition of agentic AI systems, as well as establish common strategies to deal with the inherent aforementioned problems of the core FMs and problems arising from additional tools, capabilities, and interactions. Such strategies are often known as agentic design patterns (ADPs), and there have been different efforts to organise ADPs into structures Ng [2024], Liu et al. [2025a]. However, these attempts lack a systems-theoretic basis to facilitate a rigorous understanding of agentic AI and/or are mostly convenience-based taxonomies that originate mainly from observations of practical applications. Furthermore, the proposed ADPs are often high-level and their organisation is complicated, making them less useful for direct implementation. There is also little to no connection from existing ADPs to well-established software design patterns that are widely implemented in software systems Gamma [1995]. A systematic design approach is necessary to understand the purpose of different components, as well as create a collection of design patterns that allow solving different classes of problems and be able to apply straightforwardly in creating new agentic AI system or improving existing ones Miehling et al. [2025].

This paper introduces a principled engineering discipline for agentic AI systems to address the brittleness of current ad-hoc approaches. To achieve this, we embark on a structured inquiry to answer two fundamental research questions:

1. How can we formulate agentic AI with a rigorous, systems-theoretic foundation that moves beyond monolithic FM-centric designs?
2. What are the systemic classes of problems that undermine agent reliability, and what specific, reusable design patterns can provide structural solutions?

To answer these questions, the paper is organised as follows. We first establish the problem domain by reviewing foundational concepts and identifying the critical gap in current methodologies. We then systematically categorise the challenges plaguing FM-based agents into five classes, from World Modelling to Collaboration Mechanisms, providing a clear problem map.

In response to our first research question, we introduce our core contribution: a novel system-theoretic framework that conceptualises an agent as a layered organisation of five primary functional subsystems: Reasoning & World Model, Perception & Grounding, Action Execution, Learning & Adaptation, and Inter-Agent Communication. This architecture provides the theoretical foundation for the construction and analysis of agentic systems in a principled manner.

Addressing our second research question, we derive from this framework a comprehensive collection of 12 agentic design patterns. Each pattern, such as Intergrator for data consistency or Controller for ethical oversight, is discussed with its intent and the specific problem it solves, offering a reusable solution to a recurring design challenge.

Finally, to demonstrate the framework's practical utility, we conduct qualitative case studies on a prominent agent system, ReAct, to diagnose its inherent weaknesses and prescribe targeted improvements using our patterns.

2 Design patterns in agentic AI

The idea of formulating patterns originated as soon as there was a growing trend to adopt FM systems in practice. An article, for instance, suggests seven key patterns (Evals, RAG, Fine-tuning, Caching, Guardrails, Defensive UX) arranged along the lines of enhancing performance versus cutting costs or risk, and getting closer to the data versus the user Yan [2023]. In addition, it connects these patterns to the principles of machine learning design such as data flywheel, cascade, and monitoring. The article takes into account software engineering design patterns, includes concrete and specific examples with code, and matches FM patterns with potential problems. Another master's thesis examined current FM applications, including MetaGPT, BabyAGI, and AutoGen as notable examples Ganesh and Sahlqvist [2024]. The six main architectural patterns — Retrieval-Augmented Generation (RAG), In-Context Learning, Ad-hoc, Multi-agent, Usage of Tools, and Chain-of-thought (CoT) prompting — were identified with varying degrees of granularity and presented in the Gang Of Four (GoF)'s format Gamma [1995], and their applicability to software application development was investigated. In addition, the problems that arise in FM and generative AI also require novel and unique design principles, as demonstrated in the set of six principles for generative AI applications in Weisz et al. [2024] and patterns evaluated from practical implementations in Koc [2024], Suresh [2025].

One of the first attempts to categorise design patterns in building AI agents was detailed in a series of blog posts by Andrew Ng Ng [2024]. These design patterns, namely Reflection, Tool Use, Planning, and Multi-Agent Collaboration, prove to be generalised and simple but effective approaches to enhance the performance and reliability of the system. Following that line, surveys have been conducted based on one or more of these patterns Masterman et al. [2024], Singh et al. [2024], and an evolving stack of commonly used tools and subsystems has been observed from AI startups and technology companies' solutions Andreessen Horowitz [2023], Gohel [2025]. An article takes a step further, designing FM-based agents with security in mind, in order to protect themselves from prompt injection attacks Beurer-Kellner et al. [2025]. Believing that reliable general-purpose agents are highly improbable, the authors suggest imposing agents with constraints that "explicitly limiting their ability to perform arbitrary tasks", and suggest six design patterns aimed at ensuring a certain degree of isolation between untrusted data and the agent's control flow.

Several notable efforts have proposed comprehensive reference architectures and pattern catalogues, such as the work by Lu et al. Liu et al. [2025a], Lu et al. [2024]. These approaches, often grounded in extensive literature reviews, provide valuable inventories of architectural components and design options. However, a closer analysis reveals a common characteristic: these architectures are primarily empirically-grounded aggregations of observed functionalities. While practical, this "bottom-up" approach can result in frameworks that lack a unifying theoretical foundation explaining why components interact in a certain way. Furthermore, the "patterns" identified often represent high-level architectural choices (e.g., selecting a plan generator type) rather than reusable, structural solutions to the recurring interaction problems between components, which is the essence of GoF-style patterns. Our work takes a different, principle-based route. Instead of aggregating existing features, our framework (Section 4) is derived from the first principles of system theory. This allows us to:

- Deconstruct an agent into a set of core, interacting subsystems with strong logical coherence.
- Define granular, interaction-centric design patterns that solve specific collaboration challenges between these subsystems.
- Emphasise the dynamic flows of information (e.g., context, feedback) that govern the agent's behaviour.

We recognise that a classification scheme should include the level of specificity as a key dimension, as being pointed out in Oluyomi et al. [2004] and Juziuk et al. [2014]. Besides, the relevance of these FM-based agentic AI design patterns to foundational research in design patterns for software, MAS, and AI is also important to be considered. Our classification based on the key literature identified in this section is summarised in Table 1 below.

The aforementioned publications are vital in shaping our understanding of agentic AI architectures and design patterns. Collectively, they reveal a clear trend towards more structured and reusable solutions. However, this review also highlights a significant gap in the current literature. Most existing approaches do not prioritise a cohesive theoretical foundation, such as system theory, to guide the design and analysis of agent systems. Consequently, the proposed patterns often fall into

Table 1: Comparison of literature

Literature	Specificity	GoF-related	System design	Approach
Ganesh and Sahlqvist [2024]	Specific	✓	×	Bottom-up
Ng [2024]	General	×	×	Top-down
Beurer-Kellner et al. [2025]	Specific	✓	×	Bottom-up
Liu et al. [2025a]	Specific	✓	✓	Bottom-up
Ours	Systematic	✓	✓	Integrated

two categories: either they are high-level strategic concepts (e.g., Ng's four strategies Ng [2024]) that lack detailed, implementable structure, or they are specific architectural choices (e.g., CSIRO's catalogue Liu et al. [2025a]) that, while useful, do not always capture the dynamic, collaborative essence of GoF-style patterns that solve recurring interaction problems.

This gap underscores the need for a framework that is both theoretically grounded and practically applicable through a set of well-defined, structural design patterns in the spirit of the original GoF. Our work aims to fill this gap by proposing:

- A system-theoretic agent architecture that explicitly delineates the core functional subsystems and their dynamic interactions.
- A collection of agentic design patterns that offer reusable, structural solutions to recurring problems in agent design, emphasising the 'why' and 'how' of inter-subsystem collaboration, not just the 'what' of individual components.

3 Contemporary issues in agentic AI

From the gap analysed above, we review challenges in FM-based agentic AI systems to facilitate the construction of a system design and design patterns. We categorise these problems into five classes with subproblems: World Modelling, Cognitive & Decision, Execution & Interaction, Learning & Governance, and Collaboration Mechanism. These align with human cognitive processes, such as mental modelling, reasoning, action execution, and ethical learning, providing a framework to understand the complexities of agent design. This view is also supported by LeCun's writing "Five Ways to Act Deluded, Stupid, Ineffective, or Evil" which details the five failure modes of agentic AI system based on a human behaviour model LeCun [2025], and the classification is corroborated by existing literature on this subject.

1. **World Modelling**: The challenge for FM-based agents is to create an accurate and dynamic representation of their environment. A primary issue is poor **cognitive data quality**, as models may favour pre-trained knowledge over retrieved information, leading to hallucinations and factual incorrect outputs Xi et al. [2025], Kambhampati [2024]. This is compounded by a lack of **world model consistency**, where an agent's linguistic competence is "patchy," resulting in logically inconsistent statements Mahowald et al. [2024]. Agents also struggle with **efficient context retrieval**, not just due to technical limitations on context length You et al. [2024], but more fundamentally in their inability to reliably integrate retrieved information into their reasoning Du et al. [2025], Team [2025]. Finally, long-term operation is hindered by challenges in **state saving and restoring**, where "misaligned experience replay" can cause the propagation of past errors, undermining the agent's performance over time Xiong et al. [2025].
2. **Cognitive & Decision**: The challenges in cognitive and decision making for agents stem from their probabilistic nature. Regarding **logical reasoning & uncertainty**, agents show heuristic aptitude but fail in rigorous and extended logical tasks, while their verbal confidence is an unreliable proxy for actual uncertainty Liu et al. [2025b], Han et al. [2024]. This inconsistency is due in part to the lack of a robust internal world model to simulate outcomes Hao et al. [2023]. This deficit also affects **goal-directed behaviour**, where agents struggle to adapt to dynamic environments for long-range goals, as strategies such as task decomposition can be brittle or prone to hallucinations Zheng et al. [2025], Zou et al. [2025], Huang et al. [2024]. Furthermore, agents are limited by poor **counterfactual reasoning**, as they often default to

pretrained knowledge instead of adapting to contradictory contextual information, restricting their ability to process hypothetical scenarios Yamin et al. [2025].

3. **Execution & Interaction**: A core challenge is translating plans into reliable real-world actions. Agents often lack **robustness to environmental changes**, struggling with multimodal perception and amplified hallucinations in chained actions within dynamic settings Tran et al. [2025]. Although they can be augmented with external tools, their **effective tool use** is hampered by the difficulty of integrating them into complex workflows, often resulting in non-deterministic "black-box" behaviours that are difficult to debug and control Plaat et al. [2025], Fournier et al. [2025]. This unreliability is exacerbated by inadequate **error recovery mechanisms**; agents can become trapped in unproductive cycles due to flawed reasoning, and existing reflection methods do not offer guaranteed convergence to a correct solution, especially against adversarial input Huang et al. [2024], Kumar et al. [2024].

4. **Learning & Governance**: A key technical hurdle is **catastrophic forgetting and adaptation to novel situations**, where agents forget previously learnt knowledge when acquiring new data, compromising their ability to adapt without performance degradation Li et al. [2024], Zheng et al. [2025]. Beyond learning, achieving **value alignment & transparency** is a significant challenge, Current alignment methods are costly and can become outdated, while the "black-box" nature of models obscures their reasoning and hinders public trust Padhi et al. [2024], Calderon and Reichart [2025]. This leads to complex issues in **ethical choices & moral development**, as agents lack a human-like understanding of concepts such as intention and can learn unethical behaviours, creating a significant accountability gap for their actions Zou et al. [2025], Reinecke et al. [2023], Wang et al. [2024].

5. **Collaboration Mechanism**: A fundamental obstacle is **communication and coordination breakdown**, where ambiguous language, asynchronous message sequencing, and security vulnerabilities frequently lead to misinterpretations and failures Gomez [2024], Tran et al. [2025], Zou et al. [2025], Kong et al. [2025]. This is compounded by weak **coordination and joint planning** capabilities; agents often fail to leverage collaboration effectively and exhibit poor joint planning, even in scenarios where cooperation is optimal Ni et al. [2025], Agashe et al. [2025]. Finally, navigating complex **trust and social dynamics** remains a significant hurdle. Building trust is essential for human-agent and inter-agent teams but is often undermined by undesirable emergent social behaviours and the difficulty of managing scenarios involving both cooperation and competition Tran et al. [2025], Ni et al. [2025].

4 A system-theoretic agent architecture

To address the systemic challenges outlined previously, we go beyond ad-hoc designs to propose a conceptual framework grounded in system theory. Our approach applies the principle of deconstruction Bass et al. [2003] to break down an agent into a set of core and extensible subsystems, providing a foundational language to design modular and reliable agents.

The proposed system-theoretic agent architecture, depicted in Figure 1, visualises this deconstruction. The model conceptualises an agent not as a monolithic entity but as a system of nested functional layers, where each layer represents a different level of abstraction and responsibility. The logic of this layered organisation, which is a direct result of our system-theoretic analysis, is as follows:

- **The cognitive core (innermost layer):** In the centre lies the `Reasoning & World Model (RWM)` subsystem. As the agent's decision-making nucleus, it is the most abstract and protected layer, responsible for maintaining the world model and directing all strategic behaviour.
- **The operational interfaces (middle layer):** Surrounding the core is a layer of three subsystems that act as the primary interfaces between the agent's internal reasoning and the external world. This layer includes two fundamental subsystems: the `Perception & Grounding (PG)`, which acts as the agent's senses to process and ground raw inputs into percepts, and the `Action Execution (AE)`, which serves as the agent's effectors to execute actions. For multi-agent capabilities, this layer can be extended with the optional `Inter-Agent Communication (IAC)` subsystem, the agent's social interface for structured peer-to-peer interaction.
- **The adaptive shell (outermost layer):** Encapsulating the entire system is the `Learning & Adaptation (LA)` subsystem. Its position signifies its overarching role: to observe the

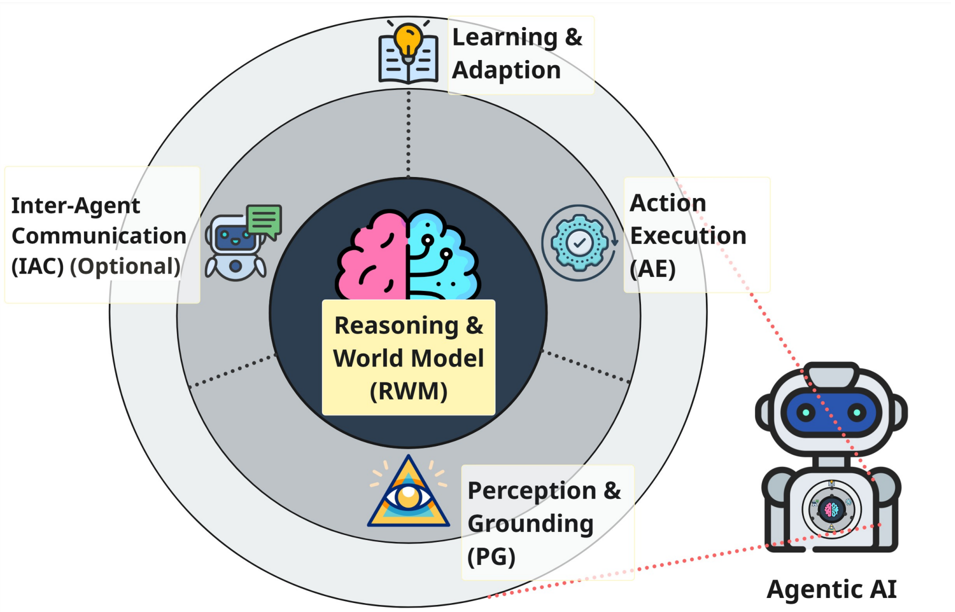

Figure 1: A system-theoretic agent architecture. The model illustrates the internal structure of an agent as nested functional layers, comprising four core subsystems and one extensible subsystem (IAC, highlighted as optional).

performance of all inner layers, learn from experience, and drive their continuous improvement through feedback.

Although the system-theoretic agent architecture illustrates the agent's static organisation into functional subsystems, its dynamic operation is best conceptualised as a continuous cognitive cycle. This cycle, a foundational concept in the design of rational agents Russell and Norvig [2010], is depicted in Figure 2 and details the key information flows that enable intelligent behaviour. The process begins with the `Perception & Grounding (PG)` subsystem processing `Raw Inputs` into `Structured Percepts`. These percepts are sent to the `Reasoning & World Model (RWM)` subsystem, which integrates them to maintain its internal world model. Based on this model, the `RWM` deliberates and generates either an `Action Plan` for the `AE` or a `Request` for the `IAC`. The results of these actions generate `Feedback`, which is processed by the `LA`. This crucial final step closes the loop: the `LA` synthesises insights into `Strategy Updates` and `Knowledge Updates`, both of which are sent back to the `RWM` to refine its future reasoning and enrich its world model, allowing true learning and adaptation Zheng et al. [2025].

This system-theoretic deconstruction into five core and extensible subsystems provides a stable yet flexible foundation for agent design. It strikes a deliberate trade-off, offering sufficient granularity for analysis while maintaining conceptual clarity. With this architectural blueprint established, we now turn to the specific and reusable solutions for its implementation: the agentic design patterns.

5 A catalogue of agentic design patterns

The design of agentic AI systems requires a structured and principled approach to address the inherent complexities of autonomy, reliability, and adaptability. Building upon the system-theoretic architecture established in the previous section, we introduce a catalogue of 12 *Agentic Design Patterns* (ADPs).

It is important to note that the concepts underlying many of these patterns are not entirely new; ideas such as reflection, skill acquisition, and tool use have long been explored across various subfields of AI. The primary contribution of this catalogue lies not in the invention of these individual concepts, but in their *systematisation* into a cohesive set of *architectural design patterns* for LLM-based agents.

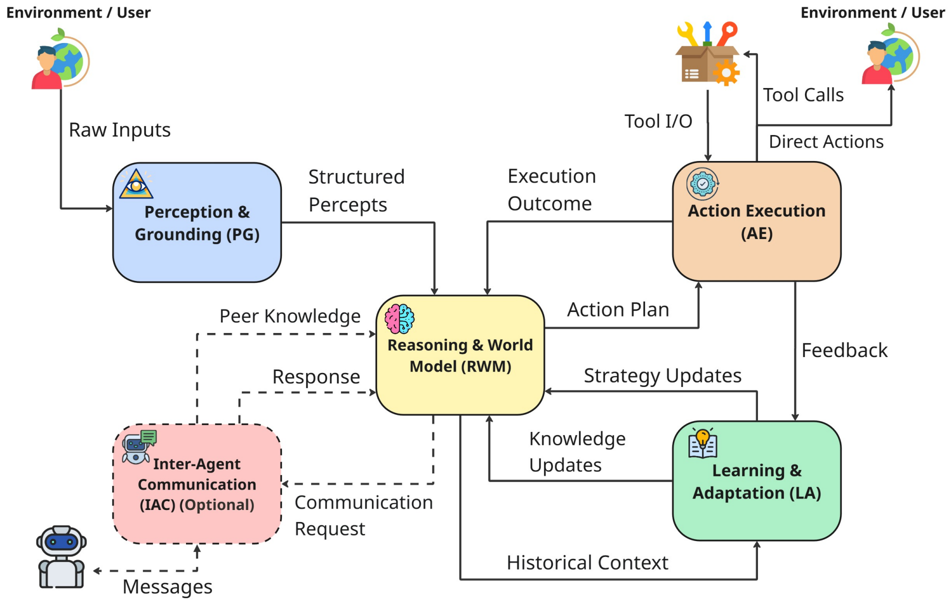

Figure 2: The agent's cognitive cycle. This diagram illustrates the dynamic interaction flows between the four core subsystems and the optional communication subsystem (`IAC`, shown with dashed lines).

Following our integrated methodology, these patterns are derived both from top-down architectural principles and from a bottom-up analysis of recurring solutions observed in the literature.

Each pattern provides a modular and reusable solution to a recurrent coordination problem among the subsystems of our framework. It establishes a standardised vocabulary and a consistent representational structure (e.g., *Intent*, *Problem*, *Solution*) that describe the involved components, their interactions and practical implications. These patterns are designed to systematically address the identified *Classes of Problems* (Section 3). Furthermore, they align with Miehling et al.'s Miehling et al. [2025] call for a systems perspective, offering a generative methodology to construct robust and reliable agentic architectures.

To present a holistic view of how these elements interconnect, Figure 3 illustrates the relationships between the major issues in Section 3, our framework's core subsystems in Section 4, and the ADPs proposed in this section. This Sankey diagram visualises the primary pathways from our identified problem classes to architectural components and finally to specific design solutions. It highlights how World Modelling issues predominantly impact the `Reasoning & World Model (RWM)` subsystem, which in turn is addressed by foundational patterns such as `Integrator` and `Retriever`.

The complete catalogue is summarised in Table 2. The patterns are organised into four groups that capture the core aspects of the operation of autonomous agents, reflecting the fundamental components of rational agents as described in the foundational AI literature Russell and Norvig [2010], Georgeff et al. [1999]. In addition, we briefly describe several representative patterns to illustrate their function and value as follows.

The `Integrator` pattern addresses cognitive data quality by defining a validation pipeline within the `PG` subsystem. For decision-making, the `Selector` pattern provides a solution for adaptive goal-directed behaviour by implementing the Mediator pattern Gamma [1995] within the `RWM` to dynamically manage and prioritise the agent's goals. For interaction, the `Tool Use` pattern ensures effective tool use by acting as a Proxy and Adapter Gamma [1995] for all external tool calls within the `AE`. Finally, the `Controller` pattern addresses value alignment by establishing a continuous monitoring loop, acting as an Observer Gamma [1995] of the agent's behaviour. A detailed description of all 12 patterns can be found in the full version of our work.

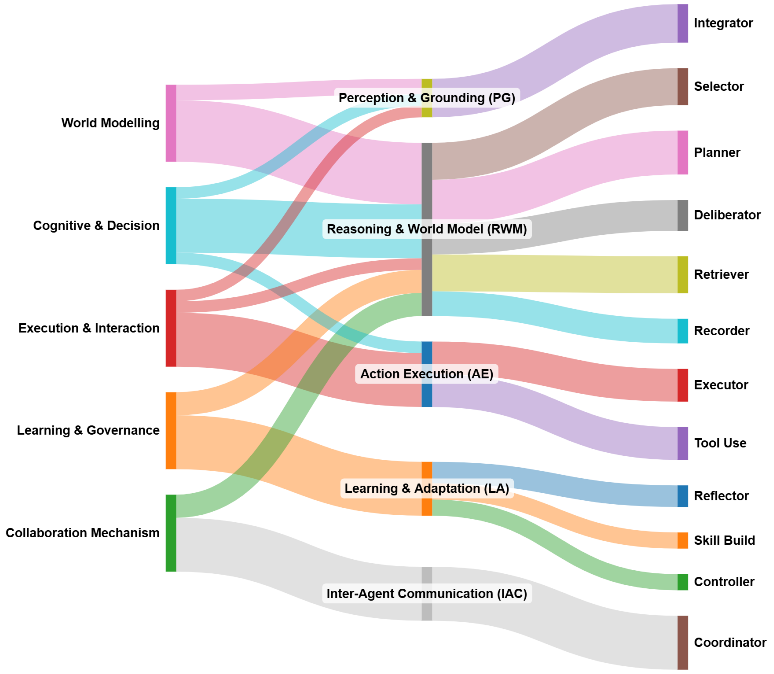

Figure 3: Conceptual Sankey diagram illustrating the relationships between identified classes of problems, core agent subsystems, and the 12 proposed agentic design patterns. Flow widths indicate qualitative relevance.

Table 2: Overview of the 12 proposed agentic design patterns (ADPs).

Pattern name	Intent	Core problem addressed
Foundational patterns: building the agent's understanding and state		
Integrator	Ensure `PG` consistency by validating all incoming information.	Cognitive data quality
Retriever	Provide a simplified, context-aware interface to the `RWM`'s memory.	Inefficient context retrieval
Recorder	Capture and externalise `RWM` state for later restoration.	State saving & restoring
Cognitive & decisional patterns: shaping agent thought and action		
Selector	Select, prioritise & adapt primary goal objectives based on dynamic contexts.	Goal-directed behavior (Tactical step selection)
Planner	Decompose high-level goals into manageable, actionable steps.	Goal-directed behavior (Strategic decomposition)
Deliberator	Select the optimal concrete action at each step of the plan.	Goal-directed behavior (Dynamic adaptation)
Execution & interaction patterns: enabling action and engagement		
Executor	Reliably execute the dispatched actions and collect systematic feedback.	Error recovery mechanism
Tool Use	Provide a secure, standardised interface for all external tool invocations.	Effective tool use
Coordinator	Manage and facilitate structured multi-agent communication.	Communication and coordination breakdown
Adaptive & learning patterns: enabling improvement and evolution		
Reflector	Analyse outcomes to infer causality and generate actionable insights.	Adaptation (Causal learning)
Skill Build	Discover and refine reusable procedural skills from experience.	Adaptation (Procedural learning)
Controller	Continuously monitor and align agent behaviour with ethical principles.	Value alignment & transparency

6 Application: a qualitative analysis of an existing system

This section uses a bottom-up approach to validate our framework, qualitatively analysing the ReAct system to demonstrate enhancements from our system-theoretic architecture through a three-step methodology.

1. **Deconstruct:** We deconstruct the architecture to map its functionalities to the five core subsystems of the framework.
2. **Diagnose:** We then diagnose architectural weaknesses by analysing which of our five problem classes manifest most prominently in the system.
3. **Prescribe:** We propose specific agentic design patterns (ADPs) as solutions to these problems.

Deconstruct: In the ReAct paradigm, the functionalities of our subsystems are implicitly and monolithically implemented within the central LLM and its interaction loop.

- The *Reasoning & World Model (RWM)* is similar to the LLM's 'Thought' generation process. Its world model is an implicit and transient state held within the LLM's limited context window.
- The *Perception & Grounding (PG)* is rudimentary; the agent perceives the world solely through unstructured 'Observations' from the environment.
- The *Action Execution (AE)* is the 'Act' step, where the LLM's generated action is passed to an external environment or tool.
- The *Learning & Adaptation (LA)* and *Inter-Agent Communication (IAC)* are absent. ReAct has no mechanism for long-term learning and is designed as a single-agent framework.

Diagnose: The original ReAct framework exhibits systemic fragilities. Its monolithic design leads to significant world-modelling challenges, as it lacks mechanisms for validating observations or managing context efficiently. The unstructured 'Thought' process results in suboptimal planning, hindering goal-directed behaviour. Furthermore, the framework lacks robust error recovery mechanisms for tool use and a dedicated process for adaptation, which prevents it from learning from failures.

Prescribe: We propose enhancing the ReAct loop by integrating specific ADPs, transforming its simple cycle into a structured workflow as illustrated in Figure 4. To address World Modelling challenges, the Integrator pattern first validates the incoming observations. In case of a critical `Inconsistency`, it can trigger the Recorder to save the problematic state and the Reflector to initiate a learning cycle. For valid data, the Retriever and Recorder patterns provide robust mechanisms for context retrieval and state management within the core `RWM`. The `RWM`'s `Thought` is then passed to the Executor and Tool Use patterns for reliable execution. Finally, the feedback from this execution is also processed by the Reflector, allowing the agent to perform causal analysis of failures and adjust future strategies, creating an adaptive and resilient agent.

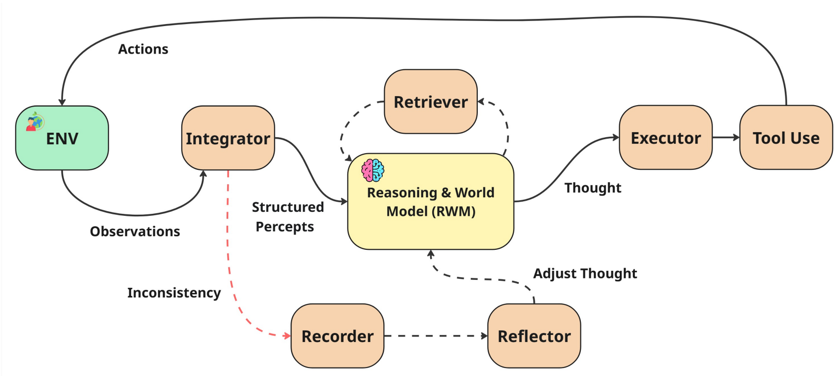

Figure 4: A conceptual diagram showing how ReAct can be enhanced by integrating our proposed agentic design patterns.

7 Limitations and future work

We acknowledge several limitations that also highlight promising directions for future research. Our framework is primarily conceptual; a critical next step is to conduct quantitative benchmarking to empirically measure the performance improvements (e.g., reliability, efficiency) offered by our patterns against baselines. Secondly, the implementation of sophisticated patterns such as Reflector and Controller introduces architectural complexity and potential computational overhead, whose trade-offs require further investigation. Finally, while our work promotes reliable agent design, it does not fully address the broader societal impacts of large-scale autonomous systems, such as accountability and emergent behaviours, which remain open critical problems.

8 Conclusion

The rapid development of agentic AI has largely relied on ad-hoc methods, resulting in systems that are powerful but often brittle and unreliable. This paper addresses this fundamental issue by introducing a principled engineering discipline grounded in system theory. We proposed a novel framework that deconstructs an agent into five core subsystems and presented a catalogue of 12 agentic design patterns that offer structural solutions to recurring problems. The practical utility of this approach was demonstrated through our qualitative analysis of the ReAct framework, where we diagnosed its systemic weaknesses and prescribed specific patterns to enhance robustness and adaptability. By providing a shared vocabulary and a structured methodology, this work aims to shift the development of agentic systems from informal experimentation to a principled engineering practice, paving the way for more modular, reliable and trustworthy autonomous agents.

Acknowledgments and Disclosure of Funding

This publication has emanated from research supported in part by grants from Research Ireland under Grant 12-RC-2289-P2 and 18/CRT/6223 which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY-NC 4.0 public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72, 2025.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.

Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B Divya. Agentic ai: Autonomous intelligence for complex goals–a comprehensive survey. *IEEE Access*, 2025.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*, 2025.

Andrew Ng. Agentic design patterns part 1: Four ai agent strategies that improve gpt-4 and gpt-3.5 performance, mar 2024. URL https://www.deeplearning.ai/the-batch/how-agents-can-improve-llm-performance/. The Batch (blog).

Yue Liu, Sin Kit Lo, Qinghua Lu, Liming Zhu, Dehai Zhao, Xiwei Xu, Stefan Harrer, and Jon Whittle. Agent design pattern catalogue: A collection of architectural patterns for foundation model based agents. *Journal of Systems and Software*, 220:112278, 2025a.

Erich Gamma. *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.

Erik Miehling, Karthikeyan Natesan Ramamurthy, Kush R. Varshney, Matthew Riemer, Djallel Bouneffouf, John T. Richards, Amit Dhurandhar, Elizabeth M. Daly, Michael Hind, Prasanna Sattigeri, Dennis Wei, Ambrish Rawat, Jasmina Gajcin, and Werner Geyer. Agentic ai needs a systems theory, 2025. URL https://arxiv.org/abs/2503.00237.

Ziyou Yan. Patterns for building llm-based systems & products. *eugeneyan.com*, Jul 2023. URL https://eugeneyan.com/writing/llm-patterns/.

Sundarakrishnan Ganesh and Robert Sahlqvist. Exploring patterns in llm integration-a study on architectural considerations and design patterns in llm dependent applications. Master's thesis, CHALMERS UNIVERSITY OF TECHNOLOGY, Gothenburg, Sweden, 2024.

Justin D. Weisz, Jessica He, Michael Muller, Gabriela Hoefer, Rachel Miles, and Werner Geyer. Design principles for generative ai applications. *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024. URL https://api.semanticscholar.org/CorpusID:267301068.

Vincent Koc. Generative ai design patterns: A comprehensive guide. *Towards Data Science*, 2 2024. URL https://medium.com/data-science/generative-ai-design-patterns-a-comprehensive-guide-41425a40d7d0.

Rahul Suresh. Beyond the gang of four: Practical design patterns for modern ai systems. *InfoQ*, 2025. URL https://www.infoq.com/articles/practical-design-patterns-modern-ai-systems/.

Tula Masterman, Sandi Besen, Mason Sawtell, and Alex Chao. The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. *arXiv preprint arXiv:2404.11584*, 2024.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Enhancing ai systems with agentic workflows patterns in large language model. In *2024 IEEE World AI IoT Congress (AIIoT)*, pages 527–532. IEEE, 2024.

Andreessen Horowitz. Emerging architectures for llm applications, 2023. URL https://a16z.com/emerging-architectures-for-llm-applications/.

Rakesh Gohel. Don't waste every day reinventing your ai agent architecture, 2025. URL https://www.linkedin.com/posts/rakeshgohel01_dont-waste-every-day-reinventing-your-ai-activity-7331296814974808065-co8M/.

Luca Beurer-Kellner, Beat Buesser Ana-Maria Creţu, Edoardo Debenedetti, Daniel Dobos, Daniel Fabian, Marc Fischer, David Froelicher, Kathrin Grosse, Daniel Naeff, Ezinwanne Ozoani, et al. Design patterns for securing llm agents against prompt injections. *arXiv preprint arXiv:2506.08837*, 2025.

Qinghua Lu, Liming Zhu, Xiwei Xu, Zhenchang Xing, Stefan Harrer, and Jon Whittle. Towards responsible generative ai: A reference architecture for designing foundation model based agents. In *2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)*, pages 119–126. IEEE, 2024.

Ayodele Oluyomi, Shanika Karunasekera, and Leon Sterling. An agent design pattern classification scheme: Capturing the notions of agency in agent design patterns. In *11th Asia-Pacific Software Engineering Conference*, pages 456–463. IEEE, 2004.

Joanna Juziuk, Danny Weyns, and Tom Holvoet. Design patterns for multi-agent systems: A systematic literature review. *Agent-oriented software engineering: reflections on architectures, methodologies, languages, and frameworks*, pages 79–99, 2014.

Yann LeCun. Five ways to act deluded, stupid, ineffective, or evil, may 2025. URL https://www.linkedin.com/posts/yann-lecun_five-ways-to-act-deluded-stupid-ineffective-activity-7327058733967052800-TsUK/.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2025.

Subbarao Kambhampati. Can large language models reason and plan? *Annals of the New York Academy of Sciences*, 1534(1):15–18, 2024.

Kyle Mahowald, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. Dissociating language and thought in large language models. *Trends in cognitive sciences*, 2024.

Haoran You, Yichao Fu, Zheng Wang, Amir Yazdanbakhsh, and Yingyan (Celine) Lin. When linear attention meets autoregressive decoding: towards more effective and efficient linearized large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z Pan. Rethinking memory in ai: Taxonomy, operations, topics, and future directions. *arXiv preprint arXiv:2505.00675*, 2025.

Contextual AI Team. Introducing the most grounded language model in the world. https://contextual.ai/blog/introducing-grounded-language-model/, 2025. Accessed: 2025-06-30.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts llm agents: An empirical study of experience-following behavior. *arXiv preprint arXiv:2505.16067*, 2025.

Hanmeng Liu, Zhizhang Fu, Mengru Ding, Ruoxi Ning, Chaoli Zhang, Xiaozhang Liu, and Yue Zhang. Logical reasoning in large language models: A survey. *arXiv preprint arXiv:2502.09100*, 2025b.

Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. Towards uncertainty-aware language agent. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6662–6685, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.398. URL https://aclanthology.org/2024.findings-acl.398/.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.507. URL https://aclanthology.org/2023.emnlp-main.507/.

Junhao Zheng, Chengming Shi, Xidi Cai, Qiuke Li, Duzhen Zhang, Chenxing Li, Dong Yu, and Qianli Ma. Lifelong learning of large language model based agents: A roadmap. *arXiv preprint arXiv:2501.07278*, 2025.

Henry Peng Zou, Wei-Chieh Huang, Yaozu Wu, Yankai Chen, Chunyu Miao, Hoang Nguyen, Yue Zhou, Weizhi Zhang, Liancheng Fang, Langzhou He, et al. A survey on large language model based human-agent systems. *arXiv preprint arXiv:2505.00753*, 2025.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. Understanding the planning of llm agents: A survey. *arXiv preprint arXiv:2402.02716*, 2024.

Khurram Yamin, Gaurav Ghosal, and Bryan Wilder. Llms struggle to perform counterfactual reasoning with parametric knowledge. *arXiv preprint arXiv:2506.15732*, 2025.

Aske Plaat, Max van Duijn, Niki van Stein, Mike Preuss, Peter van der Putten, and Kees Joost Batenburg. Agentic large language models, a survey. *arXiv preprint arXiv:2503.23037*, 2025.

Fabiana Fournier, Lior Limonad, and Yuval David. Agentic ai process observability: Discovering behavioral variability. *arXiv preprint arXiv:2505.20127*, 2025.

Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Aaron Jiaxun Li, Soheil Feizi, and Himabindu Lakkaraju. Certifying LLM safety against adversarial prompting. *Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/pdf?id=9Ik05cycLq.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4297–4308, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.249. URL https://aclanthology.org/2024.findings-emnlp.249/.

Inkit Padhi, Karthikeyan Natesan Ramamurthy, Prasanna Sattigeri, Manish Nagireddy, Pierre Dognin, and Kush R. Varshney. Value alignment from unstructured text. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1083–1095, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.81. URL https://aclanthology.org/2024.emnlp-industry.81/.

Nitay Calderon and Roi Reichart. On behalf of the stakeholders: Trends in NLP model interpretability in the era of LLMs. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 656–693, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.29. URL https://aclanthology.org/2025.naacl-long.29/.

Madeline G Reinecke, Yiran Mao, Markus Kunesch, Edgar A Duéñez-Guzmán, Julia Haas, and Joel Z Leibo. The puzzle of evaluating moral cognition in artificial agents. *Cognitive Science*, 47 (8):e13315, 2023.

Han Wang, An Zhang, Nguyen Duy Tai, Jun Sun, Tat-Seng Chua, et al. Ali-agent: Assessing llms' alignment with human values via agent-based evaluation. *Advances in Neural Information Processing Systems*, 37:99040–99088, 2024.

Kye Gomez. The hidden challenges of multi-llm agent collaboration. *Medium*, sep 2024. URL https://medium.com/@kyeg/the-hidden-challenges-of-multi-llm-agent-collaboration-59c83f347503.

Dezhang Kong, Shi Lin, Zhenhua Xu, Zhebo Wang, Minghao Li, Yufeng Li, Yilun Zhang, Zeyang Sha, Yuyuan Li, Changting Lin, et al. A survey of llm-driven ai agent communication: Protocols, security risks, and defense countermeasures. *arXiv preprint arXiv:2506.19676*, 2025.

Ansong Ni, Ruta Desai, Yang Li, Xinjie Lei, Dong Wang, Ramya Raghavendra, Gargi Ghosh, Daniel Li, and Asli Celikyilmaz. Collaborative reasoner: Self-improving social agents with synthetic conversations. https://ai.meta.com/research/publications/collaborative-reasoner-self-improving-social-agents-with-synthetic-conversations/, 2025. Accessed: 2025-06-30.

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. LLM-coordination: Evaluating and analyzing multi-agent coordination abilities in large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 8038–8057, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.448. URL https://aclanthology.org/2025.findings-naacl.448/.

Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison-Wesley Longman Publishing Co., Inc., USA, 2 edition, 2003. ISBN 0321154959.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, Upper Saddle River, NJ, USA, 3 edition, 2010. ISBN 978-0136042594.

Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. The belief-desire-intention model of agency. In *Intelligent Agents V: Agents Theories, Architectures, and Languages: 5th International Workshop, ATAL'98 Paris, France, July 4–7, 1998 Proceedings 5*, pages 1–10. Springer, 1999.