Words Made Real — An Epistemology of Entering Unfamiliar Disciplinary Domains

Building Pictorial — an AI-assisted infographic generation system developed for Tricontinental: Institute for Social Research — meant entering the domain of infographic design without formal training in it. The first attempts to specify what the system should produce ran into the same obstacle every time: there was no working vocabulary for what a correct design decision meant. Not an absence of examples — infographics were everywhere — but an absence of language: no framework for which properties of content determine which layout structure is appropriate, no terms for what distinguishes one visual register from another, no way to say why one design solution was better than another except by pointing. The breakthrough came not from acquiring design skill but from acquiring design language. Once there were names — for the difference between Layout and Style, for the seven primitives of compositional structure, for the gestalts of visual register — the rules could be written and the system could be built.

Austrian-British philosopher Ludwig Wittgenstein wrote in the *Tractatus Logico-Philosophicus* (1921): 'The limits of my language mean the limits of my world' (*Die Grenzen meiner Sprache bedeuten die Grenzen meiner Welt*).[1] Martinican psychiatrist Frantz Fanon, writing three decades later, arrived at a parallel judgement in *Black Skin, White Masks* (1952): 'to speak a language is to take on a world, a culture' (*parler une langue, c'est assumer un monde, une culture*).[2] For the researcher whose work crosses many unfamiliar disciplinary domains — often without time for formal apprenticeship in any one of them, and through literature whose authority and access are unevenly distributed — these propositions carry a direct epistemological implication: the most fundamental obstacle is not a deficit of data or skill, but the absence of the language through which to speak about it — the inability to articulate how the domain carves up its problems, names its key concepts, and organises its analysis. A domain that cannot be spoken of is a domain that does not exist within one's world.

Wittgenstein and Fanon describe the predicament from different traditions; neither account fully explains why the limit takes the form it does. The materialist account had already been given. In *The German Ideology* (1845–1846), Marx and Engels located language not as the membrane around an individual mind but as 'practical consciousness' — the accumulated, social form in which human practice exists for others and therefore for oneself.[3] To find a domain 'unspeakable', on this account, is to stand outside the practical consciousness in which that domain has been organised through the accumulated work of those who built it. The argument that

follows operates from this materialist position. The two opening formulations are retained because each names the predicament with exceptional compression from a tradition the other could not access — and because the later Wittgenstein himself moved decisively away from the *Tractatus*'s picture theory toward a use-based account broadly compatible with the position taken here.

Yet 'unspeakable' does not mean that the domain's knowledge is non-existent; it means only that the researcher has not yet mastered the language for speaking it. This distinction is crucial: it implies that the critical operation for entering an unfamiliar domain is neither the accumulation of facts nor the acquisition of skills, but rather the mastery of its language.

The emergence of large language models (LLMs) endows this epistemological judgement with practical significance. LLMs operate through language as their medium — they receive language, process language, and produce language. Consequently, any knowledge that can be sufficiently articulated in language can potentially be transformed, through LLMs, into practical action. 'Speakable' becomes 'actionable' — words made real.

Pictorial is the outcome of that entry. Mastering the language of infographic design — Layout primitives, Style gestalts, information structure classification — proved sufficient to build a system that generates professional-standard infographics with the aid of an LLM, without professional design training.

The central question this article seeks to answer is: what is the critical operation for entering an unfamiliar disciplinary domain? The answer is mastering its language — not as metaphor, but as a precise claim: vocabulary, patterns, and pattern language are not a taxonomy but a sequence of operations. Each level makes the next possible: without vocabulary, there are no patterns; without patterns, there is no pattern language; without pattern language, there is nothing precise enough for an LLM to execute.

What It Means to Speak a Domain's Language

Vocabulary: Carving Up the Problem Space

Building Pictorial made the first operative step visible. How does someone who cannot draw produce professional infographics? The answer is not aesthetic theory or colour technique — it is learning how to *carve up* the domain's problem space.

In the domain of infographic design, the problem space was found to decompose along two orthogonal dimensions: Layout, which faces the content and answers how the relationships between pieces of information are organised in space; and Style, which faces the reader and answers what kind of visual impression the graphic should convey. The two possess fundamentally different internal structures. Layout is compositional — it can be decomposed into seven basic primitives (Axis, Slot, Connector, Anchor, Stack, Branch, Annotation) that are loosely coupled and freely combinable. Style is gestalt — its six dimensions (Medium, Palette, Line, Surface, Typography, Mood) are tightly coupled and cannot be separated; they must be anchored in an actual visual tradition to function.

The power of vocabulary lies not in naming but in carving. The orthogonal decomposition 'Layout × Style' segments the continuous, nebulous experience of 'design' into discrete, operable units. Saussure made the same observation from within linguistics: 'in language there are only differences without positive terms' (*dans la langue il n'y a que des différences sans termes positifs*).[4] When we name the basic elements of a domain — Axis, Slot, Connector, or Medium, Palette, Line — we are not merely affixing labels; we are creating cognitive handles that transform previously vague intuitions into operable concepts.

Stylistics provides a parallel illustration. At Tricontinental, the team specified the institution's own publication style across seven levels of language — from graphology and phonology through lexis to pragmatics and discourse. At the lexis level, for instance, the specification distinguishes between 'hegemony' (a structural concept) and 'dominance' (a looser near-synonym that the institution's publications consistently avoid): an accepted term and a rejected one, constituting a single operable rule where previously there was only inarticulate preference. These are equally operations of carving — equally the transformation of tacit 'stylistic sense' into operable dimensions.

Patterns: The Encoding of Contradictions

Vocabulary constitutes the foundational material; patterns are the instantiation of that material in specific contexts. Pattern theory was originally proposed by Christopher Alexander in the domain of architecture. In *Notes on the Synthesis of Form* (1964), he investigated the structured decomposition of design problems; in *A Pattern Language* (1977), he systematically described 253 architectural and urban design patterns; and in *The Nature of Order* (2002–2005), he developed the concept of 'centres' to explain how patterns generate 'living structure'. The core of a pattern is not invention but the identification and naming of contradictions and solutions that recur in practice. Each pattern describes a problem, the competing forces within that problem, and a solution that balances those forces.[5]

This method found extensive application in software engineering. From Gamma et al.'s design patterns (GoF, 1994) through Fowler's analysis patterns (1996), Buschmann et al.'s architectural patterns (POSA, 1996), Fowler's patterns of enterprise application architecture (PoEAA, 2002), Hohpe and Woolf's enterprise integration patterns (EIP, 2003), to Beck's implementation patterns (2007), the pattern method spans every level from code to system architecture. The PLoP (Pattern Languages of Programs) community continues to document and disseminate patterns.

In information visualisation, specific Layouts (such as sankey-flow) and Styles (such as clean-analytics) are themselves patterns — particular instantiations of the vocabulary that encode the domain's contradictions and their resolutions. Sankey-flow encodes the contradiction between 'information completeness and cognitive load' — when nodes become too numerous, they must be merged or degraded. Each prohibition in clean-analytics (no decorative illustrations, no 3D effects) encodes the contradiction between 'visual appeal and information clarity'. To fail to grasp these contradictions is to fall into metaphysics — selecting a layout or style in isolation without understanding why it is superior to another in a given context.

Pattern Language: The Organisational Structure of Patterns

Patterns do not exist in isolation; they constitute an organised 'language'. Alexander's 253 patterns descend from the scale of towns to architectural details, interconnected through a network of cross-references. POSA proceeds from architectural patterns through design patterns to idioms, forming a system of descending granularity. Cunningham's CHECKS represents a classic exemplar of pattern language in the software domain. The full title of PLoP — Pattern Languages of Programs — signals that the community has, from its inception, emphasised that patterns constitute a 'language' rather than merely a catalogue. What matters is not the quantity of patterns but the organisational rules governing their interrelation.

In information visualisation, this organisational relation manifests as: the mapping from information structures to layouts (with conditions and degradation paths), degradation semantics (how to settle for the next-best option when perfection is unattainable), document-level consistency (style is a document-level decision, not a per-graphic decision), and skeleton notation (the formalised expression of patterns). These rules determine how patterns relate to one another, how selections are made, and how degradation proceeds.

To master the language of a domain is to master all three: vocabulary, which carves the problem space into discrete operable units; patterns, which encode the contradictions and solutions that recur in practice; and pattern language, which determines the organisational rules governing how patterns relate, how selections are made, and how degradation proceeds.

The Epistemological Path

These three levels are not learned from textbooks. The complete epistemological path is: beginning with a practical problem, discovering the domain's existing concepts and frameworks through dialogue with an LLM, tracing original sources to confirm their authority, extracting vocabulary, making patterns explicit, encoding them as AI-executable rules, having the AI execute and produce output, and refining vocabulary and patterns from the output. In building Pictorial, the path into infographic design began not with design theory but with a diagnostic question: what makes this infographic hard to read? The LLM's response did not return aesthetic preferences — it returned structural terms: information density, visual hierarchy, cognitive load, the distinction between layout as argument structure and style as reader address. These became the first vocabulary entries. From them, the first patterns became visible: the conditions under which a flow diagram degrades from a sankey to a simpler axis arrangement, the rule that style is a document-level commitment rather than a per-graphic decision.

Mao described the same movement in *On Practice*: from perceptual knowledge to rational knowledge, then using rational knowledge to guide new practice.[6]

This path contains a bootstrapping problem: how does one find the language before having mastered it? The common difficulty is not that 'practitioners cannot articulate tacit knowledge', but rather that newcomers 'do not know where to look' — they cannot find the right books, or they find a multitude of books without knowing which are authoritative. Here, the LLM serves as a bootstrapping instrument: having been trained on the domain's literature, it can, through dialogue, guide one towards the correct direction. The LLM thus functions not only as an execution engine but also as a discovery engine.

The resolution is iterative, not immediate. A researcher entering an unfamiliar domain begins with whatever language they already possess — everyday terms, adjacent concepts, rough analogies. The LLM, trained on the domain's literature, meets the researcher at that level and progressively introduces more precise vocabulary. Each exchange narrows the gap. The bootstrapping is not solved in a single move but dissolved through iteration: each cycle of dialogue produces enough language for the next cycle to go deeper.

This path works in practice. But why does it work? Why does mastering the language of a domain produce the effect of entering it — and why does that entry become actionable when the tool doing the executing is an LLM? The answer lies not in the technology but in the nature of language itself.

Why Language Has This Power

Language as the Medium of Thought

The later Wittgenstein, in the *Philosophical Investigations*, turned towards a considerably richer position. Section 43 proposes that 'the meaning of a word is its use in the language' (*Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache*) — Wittgenstein qualifies this as applying to 'a large class of cases' rather than all cases, but for our discussion this suffices.[7] Language acquires meaning within 'language games' (*Sprachspiel*, from section 7 onwards), which are embedded in 'forms of life' (*Lebensform*, section 19, section 23). Section 23 makes explicit: 'the word "language-game" is meant to bring into prominence the fact that speaking a language is part of an activity, or of a form of life' (*Das Wort 'Sprachspiel' soll hier hervorheben, dass das Sprechen der Sprache ein Teil ist einer Tätigkeit, oder einer Lebensform*).[8] Each disciplinary domain constitutes a form of life, with its own language games. What was accomplished in the domain of information visualisation was, in essence, learning the language game of that domain — and the same word can mean different things in different games within that domain. 'Flow', in the Pictorial system, names an information structure (a semantic relationship of directed movement: '62% flows to Y') and a layout family (sankey-flow, funnel) — two different games played with the same word.

Marx wrote in *The German Ideology*: 'Language is as old as consciousness — language *is* practical consciousness that exists also for other men and hence exists for me personally as well' (*Die Sprache ist so alt wie das Bewusstsein — die Sprache* ist *das praktische, auch für andre Menschen existierende, auch für mich selbst erst existierende wirkliche Bewusstsein*).[9] Language is not a private psychological phenomenon but a social product that encodes the experience accumulated by human beings through practice. When an LLM is trained on human language, it is being trained on humanity's accumulated 'practical consciousness'.

Vygotsky, in the seventh chapter of *Thinking and Speech*, further argued that thought is not 'expressed' in words but rather 'completed' in them. 'The word is the microcosm of consciousness.'[10] Language is not merely an instrument of communication; it is an instrument of thought — thought acquires its form through language.

In the practice of building Pictorial, this was not a philosophical observation — it was a working constraint: without a name for the difference between Layout and Style, no rule could be written for the system to make that distinction.

The *Philosophical Investigations*, Marx, and Vygotsky point in the same direction: language is not a tool for describing thought; it is the medium of thought itself. If this judgement holds, the generality of LLMs ceases to be mysterious — what they operate upon is not symbols, but the very substance of thought.

The Isomorphism Between Knowledge Structure and Language Structure

The three-layered knowledge structure described in the previous section is isomorphic with language itself. Vocabulary corresponds to the paradigmatic axis — the set of elements available for selection at each position; patterns correspond to the syntagmatic axis — particular elements selected and arranged together to form meaningful combinations; and pattern language corresponds to *langue* — the system of rules governing these selections and combinations.[11] In Pictorial, the funnel layout makes this concrete: the paradigmatic axis is the set of seven primitives {Axis, Slot, Connector, Anchor, Stack, Branch, Annotation} from which any layout draws; the syntagmatic axis is the funnel's particular selection — a vertical Axis, a diminishing Stack of Slots, thin Connectors between stages; and *langue* is the rule that flow-type information structures should map to funnel or sankey-flow layouts, together with the degradation rule that too many stages must be merged.

Construction Grammar arrived independently, from within linguistics, at an analogous three-layered structure. Adele Goldberg argued in *Constructions: A Construction Grammar Approach to Argument Structure* (1995) and *Constructions at Work* (2006) that language is a structured constructicon — a repertoire of constructions (form-meaning pairings) that exist at every level from morpheme to discourse, organised through inheritance networks. 'Constructions all the way down' — there is no principled boundary between lexicon and grammar, just as in a pattern language there is no sharp boundary between individual patterns and organisational rules. Moreover, construction grammar is usage-based: knowledge emerges from patterns in practice.[12]

That a pattern theory developed from architecture and software engineering and a construction theory developed from within linguistics should share this three-layered structure is not coincidental. Returning to Marx's judgement — 'language is practical consciousness' — the knowledge accumulated by human beings through practice is necessarily organised in linguistic form, and therefore necessarily exhibits the structural characteristics of language itself.

The Generality of LLMs Follows from the Nature of Language

All explicit human knowledge — in medicine, law, engineering, architecture, information visualisation — is stored, transmitted, and practised through language. LLMs have learned the language system (*langue*) from massive quantities of speech acts (*parole*). Different domains have different *parole*, but they share the deep structure of *langue*. This argument also finds corroboration from the negative side: research has demonstrated that the training data of LLMs derives predominantly from the English-language internet, and what LLMs learn from *parole* includes not only linguistic structures but also the ideological biases embedded within them.[13]

This line of inquiry carries an important historical footnote. Lydia H. Liu's 2021 research revealed that Wittgenstein's philosophy, through Margaret Masterman and the Cambridge Language Research Unit, directly inspired the development of computational language technologies — machine translation, information

retrieval, knowledge representation — that were subsequently incorporated into the domains of AI and cognitive science.[14] From Wittgenstein to contemporary LLMs, this is a historical connection, not an analogy.

Searle argued in the 'Chinese Room' thought experiment (1980) that 'syntax is not sufficient for semantics' — that manipulating symbols does not constitute understanding.[15] But the claim here is epistemological, not cognitive-scientific. The question is not whether LLMs 'understand'; the question is whether knowledge is organised and transmitted in linguistic form, and whether LLMs can effectively operate upon that form. Within the framework of Wittgenstein's 'meaning as use' and Marx's 'language is practical consciousness', 'whether it truly understands' is not the right question; 'whether it can effectively participate in the language game' is.

If this philosophical argument is correct, it carries a historical prediction: at any point when language could be sufficiently articulated but could not be directly executed, there would be a community of practitioners producing excellent descriptions that produced nothing. The pattern community is exactly that community.

From Silence to Action

The aspiration was always correct. What was missing was not better descriptions but a medium that could execute them.

Wittgenstein wrote in the final proposition of the *Tractatus Logico-Philosophicus*: 'Whereof one cannot speak, thereof one must be silent' (*Wovon man nicht sprechen kann, darüber muss man schweigen*).[16] The practice traced in this article points towards an inversion of this proposition: whereof one can speak, thereof one can act. The task of epistemology is to transform the unspeakable into the speakable — the preceding sections have demonstrated how this path unfolds.

The Impasse of the Pattern Community

Building Pictorial required solving a problem that had no existing solution: design knowledge existed in practice but not as articulable language. It had to be extracted from practice, named, and formalised before it could be specified to any executor, human or machine. The knowledge was latent in the domain; the vocabulary for specifying it had to be built. Once it was built, the LLM could execute it directly. This gap between articulable knowledge and executable output is not unique to infographic design — the pattern community in software engineering had encountered it decades earlier, at scale.

The pattern community harboured the aspiration of 'words made real' from its very inception. Beck and Cunningham, in their 1987 OOPSLA paper, took Alexander as their point of departure and proposed that computer users should be able to write their own programs — domain experts who understood nothing of Smalltalk's internal mechanisms could, using a pattern language, design a reasonable user interface within a single day.[17] When Gamma, Helm, Johnson, and Vlissides published *Design Patterns* (GoF) in 1994, their objective was to record the design experience of experts so that novices could 'get design decisions right the first time'.[18] The common kernel of these visions was: describe expert knowledge well enough, and non-experts will be able to act upon it.

Yet objective conditions constrained this aspiration — between description and implementation stood human skill. Alexander himself, in his 1996 OOPSLA keynote address, criticised the software pattern community: they had taken the 'format' of patterns without acquiring their 'generative' power — patterns were being used to describe existing design decisions rather than to generate new designs.[19] Brian Marick, in his 2017 Deconstruct conference talk, reviewed this history: GoF was supposed to be the 'first step' towards pattern languages, but 'all fell apart' — people ended up merely copying class diagrams rather than using patterns generatively.[20] Patterns constituted a 'correct but powerless' form of knowledge: experts already knew them and therefore did not need them; novices, having read the descriptions, still could not implement them.

The result is that the pattern movement has been in sustained decline over the past two decades. During its peak, PLoP sustained multiple active regional conferences — KoalaPLoP (Australia), VikingPLoP (Nordic countries), MensorePLoP (Japan), ChiliPLoP, and SugarLoafPLoP (Latin America). Today, KoalaPLoP (defunct since 2002), MensorePLoP (held only once), and VikingPLoP (defunct since 2017) no longer exist. The main conference PLoP accepts approximately 24–30 papers per edition, and in 2024 was renamed to Pattern Languages of Programs, People & Practices to broaden its scope.[21] The community has not vanished, but its influence has contracted significantly.

LLMs Eliminate the Rupture Between Description and Implementation

Before the emergence of LLMs, the chain from knowledge description to implementation ran through human skill: however excellent the description, a skilled practitioner was required to execute it. After the emergence of LLMs, the description passes through language directly to the LLM, and the output is produced without that intermediary. The description itself is the execution instruction.

The author verified this transformation in the practice of building Pictorial. A researcher who cannot draw, by mastering the pattern language of infographic design — information structure classification, Layout primitive composition, Style gestalt and cultural anchoring — enables the LLM to generate professional-standard infographics. The system's three-stage pipeline — the Extractor identifies the information structure, the Advisor matches Layout and Style, the Renderer produces the image — is AI-assisted: at each stage the researcher reviews the output, editing information structures, overriding layout and style recommendations, and approving images before they enter the publication. This constitutes a concrete realisation of 'description as execution': design knowledge was transformed from a disciplinary threshold requiring years of professional training into a methodology that could be systematically encoded and executed with AI assistance.

This transformation can be stated directly. The value of a pattern depends on two variables: description quality, and the implementation discount factor. Before the emergence of LLMs, that discount factor was very low — however well described, without the requisite skill nothing could be executed, and the practical value of patterns was severely diminished. LLMs raise this factor to approximately one. Description quality becomes the primary variable determining the value of patterns.

A Paradigm Shift in Knowledge Production

This means that the quality of pattern descriptions — the work the pattern community has been pursuing for three decades — has become high-value work. At PLoP 2024, an Imagination Run Wild session titled 'The New Emperor's Old, Old Clothes: Patterns & Programming in the Era of AI' began to address the impact of AI on the pattern community, and AsianPLoP 2025 adopted 'AI & Patterns' as its conference theme.[22] Michael Weiss presented 'An Exploration of Pattern Mining with ChatGPT' at EuroPLoP 2024.[23] The community is perceiving the change, but no one has yet explicitly advanced the core thesis of this article: that LLMs render patterns executable.

Pattern languages consequently become the critical interface between human knowledge and AI execution capability. Extracting patterns from practice — identifying contradictions, naming the competing forces within problems, describing solutions that balance those forces — becomes one of the most valuable forms of knowledge work. This is not merely a revival of software design patterns; rather, all domain knowledge amenable to pattern formalisation can potentially be transformed, through LLMs, into executable capability.

The methodological specifications developed in the practice of knowledge engineering — from issue analysis workflows to the complete steps for constructing steelman arguments — are themselves instances of pattern language in knowledge engineering.

New demands are placed upon pattern writing as well: the reader is no longer exclusively the human practitioner, but also the LLM. Pattern descriptions must be sufficiently precise to serve as the basis for action.

A prompt specification developed in the practice of knowledge engineering — 'implicitly apply the epistemological framework' — exemplifies this: 'The ideological framework should: Inform analysis of power, interests, and structures... The ideological framework should NOT: Appear as explicit rhetorical terms; Replace evidence with assumption.'

The specification exemplifies pattern writing that is 'precise enough for an LLM to act upon'.

The Human Use of Human Beings

Norbert Wiener proposed in *The Human Use of Human Beings* (1950) that machines should liberate human beings from repetitive labour, freeing them for more creative intellectual work.[24] Wiener's insight acquires a more precise signification in the age of LLMs: what LLMs automate is the chain from description to implementation; human irreplaceability resides in the description itself — in the judgement of which decomposition is correct, which forces are primary, and what constitutes a good solution.

To take information visualisation as an example: the most fundamental decomposition — the orthogonal decomposition 'Layout × Style' — was not invented from thin air but extracted from the most accomplished projects after extensive survey of existing infographic practice and comparison of different decomposition schemes. The judgement of 'most accomplished' derives from the perception of core contradictions — it is the contradiction between 'information density and cognitive load' that determines layout selection, and the contradiction between 'visual appeal and information clarity' that determines style trade-offs. This

consciousness of contradiction is not something that a verification protocol can encode; this is what Wiener meant by the human use of human beings — the capacity for abstract thought and value judgement.

In the knowledge engineering practice that produced these specifications, this was established as a hard prerequisite: 'What is the fundamental contradiction driving the current situation?' If the researcher cannot identify the principal contradiction, the research does not commence.

'Human-in-the-loop' is a concept widely deployed in the AI domain, yet its meaning is frequently degraded to approval or oversight — the human as a final checkpoint, verifying whether the AI's output meets standards. This reduces the human to a mechanical checkpoint. The 'human use of human beings' in Wiener's sense refers to a deeper form of intervention: providing consciousness of contradiction and aesthetic judgement, declaring the epistemological framework — determining from what standpoint to view the world — these can only come from human beings. LLMs can operate effectively within existing language games, but the rules of the language game must be established by humans.

Conclusion

Returning to the question posed in the introduction: what is the critical operation for entering an unfamiliar disciplinary domain? This article's answer is: mastering its language — unfolded across three levels: carving vocabulary (decomposing continuous, nebulous experience into discrete, operable units), identifying patterns (encoding contradictions and solutions that recur in practice), and establishing a pattern language (determining the organisational rules governing patterns). In the age of LLMs, mastering the language is sufficient for practice — the description itself is the execution instruction.

This judgement points towards a new division of intellectual labour. Humans are responsible for the language itself — carving the problem space, naming basic elements, identifying core contradictions, declaring the epistemological framework. AI is responsible for execution — transforming sufficiently articulated knowledge into practical action. This is neither 'AI replaces humans' nor 'AI is merely a tool', but rather a division of labour grounded in the respective capabilities of each — the 'human use of human beings' in Wiener's sense.

The argument of this article carries different implications for three communities. For the pattern community, the pattern descriptions accumulated over three decades acquire new value in the age of LLMs — description quality directly determines execution quality, and pattern writing deserves revival and refinement. For the linguistic community, the fact that pattern theory and construction grammar, proceeding from entirely different traditions — architecture and software engineering on the one hand, linguistics on the other — independently arrived at the same three-layered structure constitutes a convergence worthy of further theoretical investigation.

For researchers seeking to enter unfamiliar domains with the aid of AI, the path from vocabulary to patterns to pattern language to LLM execution constitutes an actionable epistemological path.

[1] Wittgenstein, *Tractatus Logico-Philosophicus* (1921), 5.6.

[2] Frantz Fanon, *Peau noire, masques blancs* (Paris: Éditions du Seuil, 1952), Chapter 1, 'Le Noir et le langage'.

[3] Karl Marx and Friedrich Engels, *Die deutsche Ideologie* (1845–1846); the full passage is quoted at note 9 below.

[4] Ferdinand de Saussure, *Cours de linguistique générale* (1916), Chapter IV, section 4.

[5] Christopher Alexander, *Notes on the Synthesis of Form* (1964); *A Pattern Language* (1977); *The Nature of Order*, 4 vols. (2002–2005).

[6] Mao Zedong, *On Practice* (1937). Translation from the *Selected Works of Mao Tse-tung*, Vol. I (Foreign Languages Press, Peking, 1965); wording may vary across English editions.

[7] Wittgenstein, *Philosophische Untersuchungen* (1953), section 43.

[8] Ibid., section 23.

[9] Marx & Engels, *Die deutsche Ideologie* (1845–1846), MEW Bd. 3, S. 30.

[10] Vygotsky, *Thinking and Speech* (1934), Chapter 7. 'Thought is not expressed but completed in the word.'

[11] Terminological note: Saussure himself used 'associative relations' (*rapports associatifs*) rather than 'paradigmatic axis'; the latter was introduced by Hjelmslev in *Prolegomena to a Theory of Language* (1943).

[12] Adele E. Goldberg, *Constructions: A Construction Grammar Approach to Argument Structure* (1995); *Constructions at Work: The Nature of Generalization in Language* (2006).

[13] On ideological bias in LLM training data, see Emily Bender et al., 'On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?' FAccT (2021).

[14] Lydia H. Liu, 'Wittgenstein in the Machine', *Critical Inquiry* 47, no. 2 (2021).

[15] John Searle, 'Minds, Brains and Programs', *Behavioral and Brain Sciences* 3, no. 3 (1980); Margaret Boden, 'Escaping from the Chinese Room', in Boden (ed.), *The Philosophy of Artificial Intelligence* (1990).

[16] Wittgenstein, *Tractatus Logico-Philosophicus* (1921), 7.

[17] Kent Beck & Ward Cunningham, 'Using Pattern Languages for Object-Oriented Programs', OOPSLA 1987 Workshop on Specification and Design for Object-Oriented Programming.

[18] Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software* (1994); 'A Look Back', InformIT (2009).

[19] Christopher Alexander, Keynote address, OOPSLA 1996.

[20] Brian Marick, 'Patterns Failed. Why? Should We Care?' Deconstruct 2017.

[21] PLoP community data from plopcon.org and Hillside Group official records.

[22] PLoP 2024, 'The New Emperor's Old, Old Clothes: Patterns & Programming in the Era of AI', Imagination Run Wild session by Antonio Mana and James Noble; AsianPLoP 2025 conference theme 'AI & Patterns'.

[23] Michael Weiss, 'An Exploration of Pattern Mining with ChatGPT', EuroPLoP 2024.

[24] Norbert Wiener, *The Human Use of Human Beings: Cybernetics and Society* (1950; rev. ed. 1954).