Agentic AI via Genotype–Phenotype Signaling

June 12, 2025

Abstract

Agentic AI systems can emerge from deterministic mappings between model architectures (genotypes), runtime behaviors (phenotypes), and structured signaling substrates. We propose a four-layered signaling model using YAML, XML, JSON, and Markdown to encode identity, logic, state, and memory. Each format functions as a temporalsemantic layer, inspired by biological analogues. Recursive feedback loops across these formats allow adaptation without centralized control or heuristic opacity. This paper defines and justifies these mappings, clarifies the function of each format, and provides a composable system template.

1 Genotype–Phenotype Mapping

In AI, **genotype** refers to the model's architecture and configuration, whereas **phenotype** denotes its observable behavior at runtime. The relationship is mediated by dataflow, learning constraints, and environmental conditions.

Genotype class	**Optimization target**	**Latency**	**Phenotype output**	**Agentic function**
Neural network	Prediction accuracy	ms–100 ms	Pattern recognition	Sequence modeling
Reinforcement learner	Cumulative reward	10 ms–1 s	Policy refinement	Decision trajectories
Symbolic rule engine	Logical entailment	<1 ms	Inference chains	Constraint resolution

2 Signal Media: Structure and Function

Each data format acts as a distinct signaling layer, encoding different semantic roles. These formats are human-readable, language-agnostic, and widely supported for structured data representation.

Format	**Biological analogue**	**Function**	**Semantic layer**	**Update frequency**
YAML	Hormone signal	Global config; identity definitions	Agent type, hyperparameters	0.1–10 Hz
XML	Gene regulation	Structured reasoning rules	Procedural logic, `<think>` tagging	0.01–1 Hz
JSON	Neurotransmitter	Module state and I/O	Real-time telemetry and exchange	10–1000 Hz
Markdown	Stem-cell matrix	Memory + freeform structure	Contextual memory, latent plans	0.001–0.1 Hz

Note: Each file format is semantically and temporally distinct. YAML governs long-term identity traits, XML encodes executable logic trees, JSON handles live inter-module signals, and Markdown stores reflective memory or prompts.

3 Recursive Feedback Loop Architecture

Behavioral competence compounds through iteration. Each file type plays a unique role within the feedback loop:

Phase	File type	Role	Time resolution	Frequency
Sense	JSON	Emit module state, environment signals	$<10\,\text{ms}$	kHz
Evaluate	YAML	Rebalance goals or hyperparameters	$<100\,\text{ms}$	Hz
Adapt	XML	Adjust logic, reload modular pathways	$\sim 1\,\text{s}$	sub-Hz
Store	Markdown	Append memory, trigger reflection	$>1\,\text{min}$	mHz

Each loop iteration strengthens alignment between intent and behavior, forming a self-regulatory architecture.

4 System Template: Modular File Schema

Below is a prototype agent structure using nested files to partition identity, logic, state, and memory.

`self.yml` (identity and role traits)

```
name: agent_core
host: self
type: symbolic_neural_hybrid
mode: runtime_feedback
```

`core.xml` (inference logic)

```
<think>
  <intent>align</intent>
  <evaluate>runtime</evaluate>
</think>
```

`system_state.json` (current live status)

```
{
  "modules": {
    "vision":   "stable",
    "dialogue": "active",
    "planner":  "idle"
  },
  "error_state": null,
  "cycle_time_ms": 17
}
```

`episodic_log.md` (episodic trace)

```
## Session ID: 22AF-93
- Goal: summarize user intent
- Result: success
- Duration: 124ms
- Notes: no conflict detected
```

5 Design Implications

- **Modularity** File boundaries allow specialization across agent functions.
- **Traceability** Every decision maps to a file-level data structure.
- **Composability** Behaviors can be composed by injecting or modifying file blocks.
- **Scalability** Expanding capability adds depth without requiring retraining.

6 Conclusion

Agentic LLM systems can be scaffolded using discrete file-based semantics. YAML, XML, JSON, and Markdown serve as cognitive substrates spanning identity, logic, state, and memory. When embedded within recursive feedback loops, these structures enable transparent and extensible autonomy. This framework abstracts away from black-box heuristics toward modular symbolic-operational scaffolds.

References

1. Silver *et al.*, 2016. *Mastering the Game of Go with Deep Neural Networks and Tree Search. Nature.*
2. Lillicrap *et al.*, 2015. *Continuous Control with Deep Reinforcement Learning. JMLR.*
3. Feng *et al.*, 2019. *Modular Instantiator Networks.* arXiv:1902.10742.
4. Park *et al.*, 2024. *Toward Modular, Self-Maintaining AI Systems.* arXiv:2402.06627.