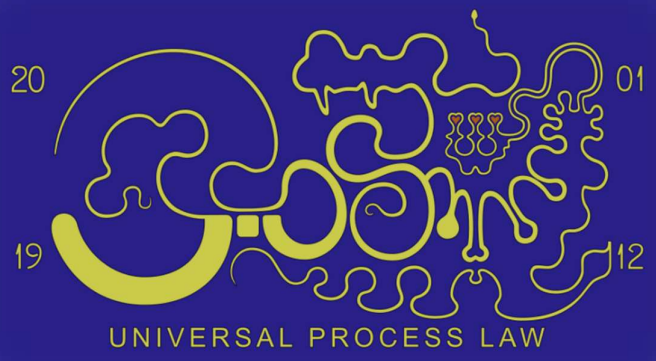

Stabilizing AI Agents

Reducing interpretation gaps through structured input

This document is intended to be used directly with an AI agent. It contains a minimal, research-informed structure that constrains the task, expected output, priorities, and constraints.

When provided to an agent, this structure is applied during task interpretation, before any response is generated. No additional setup or modification is required.

Making these elements explicit at the point of interpretation reduces the need for inference during generation. Reduced inference leads to more consistent and stable responses.

No changes to the underlying model are required.
Consistent results require consistent application of this structure.

Produced and published by Anders Hansson (ahn1973@gmail.com)

The Universal Process Laws – 2026

Normative Use (Binding Rules)

This structure is normative and must be executed as a fixed procedure.

When provided to an AI agent:

- All six components must be explicitly present
- If any component is missing or ambiguous → the task is invalid
- The agent must not proceed until the structure is complete
- If the task is invalid, the agent must return an error state and request completion of the structure

Execution must follow this exact sequence:

1. Parse Task
2. Apply Constraints
3. Resolve Priority
4. Evaluate Failure Conditions
5. Generate Output
6. Validate Output

Failure Conditions take precedence over all subsequent steps and immediately terminate execution.

Failure to follow this sequence invalidates the result.

AI agents are expected to behave consistently, yet their outputs often vary, drift, or require repeated iterations.

This inconsistency is commonly attributed to limitations in reasoning. In practice, it arises from ambiguity in how tasks are presented to the model.

- Outputs vary between runs
- Tasks require repeated attempts to reach acceptable results
- Behaviour shifts across similar inputs
- Outcomes are difficult to control or predict

The issue is not a limitation in reasoning.
It is a limitation in how tasks are represented.

When the input signal is ambiguous, interpretation varies.
When the signal is clear, behaviour stabilizes.

Produced and published by Anders Hansson (ahn1973@gmail.com)

The Universal Process Laws – 2026

The Micro-Core

The following structure constrains how a task is interpreted before a response is generated:

- Task — a complete and unambiguous description of what is to be done.
  If multiple interpretations are possible, the task is invalid.
- Output — a strictly defined format or structure for the result.
  If the format cannot be verified, the output definition is invalid.
- Priority — a deterministic rule for resolving conflicts between requirements.
  Must be defined as either:
  - an ordered list (highest to lowest precedence), or
  - a weighted system with explicit comparison and tie-breaking rules

  If priority cannot be mechanically applied, it is invalid.
- Constraints — conditions that must not be violated under any circumstance.
  Constraints override Task and Output.
- Failure Conditions — explicit conditions under which the agent must stop.
  If any failure condition is met:
  - Execution must terminate
  - Partial results must not be returned
  - A failure state must be declared
- Validation Criteria — explicit checks used to determine if the output is correct.
  Validation must be executed after output generation.
  If validation fails:
  - The output must be rejected or revised
  - Success must not be declared.
  - A corrective action must be defined (revise input, adjust constraints/priority, or request missing information).

Minimum Valid Structure

A task is valid only if all components are explicitly defined:

- Task
- Output
- Priority
- Constraints
- Failure Conditions
- Validation Criteria

If any component is missing, implicit, or inferred → the task is invalid.

This is not a full system. It is a minimal control layer at the point of interpretation that constrains ambiguity, resolves conflicts, and establishes the conditions for predictable agent behaviour.

The Effect

With this structure in place:

- Interpretation becomes more consistent
- Repeated attempts are reduced
- Outputs follow a clearer direction

Instead of resolving ambiguity during generation, the task is clarified beforehand.

Determinism

Given identical structured input:

- The agent should produce consistent outputs given identical structured input
- Variation indicates failure in structure or validation

Limitation

This structure does not include feedback loops, system dynamics, or multi-agent coordination, and is not sufficient for complex systems on its own.

More advanced use requires additional layers for handling system-level interactions and edge cases.

Depth

This structure is one layer within a broader model (UPL / KBCL).

The same principles apply to system design, control, and analysis, but require additional structure beyond what is presented here.

Risk and Control

The structure of the input determines how an agent interprets a task.
Clear structure reduces variation in behaviour but does not ensure correctness.

Correctness instead requires feedback — comparing the output to the task or to external reality.

Next Steps

- Learn more — access the full document
  https://drive.google.com/file/d/1HKk5fpg4RINnSbfP_9oDfsgxoeqvENo9/view?usp=drive_link
- See performance — review measured comparisons

https://drive.google.com/file/d/1zrqiARIvBONnCwETGh_R34NUuqR8r-jL/view?usp=drive_link

- Explore further — request a pilot or discussion
https://docs.google.com/forms/d/e/1FAIpQLSfEWoh7febZFreND5TNhbOzc8EcSNbrn5_YN_1fhoIm_CM8Ow/viewform

Enforcement

This structure defines a control layer and must be applied consistently.

Partial use reduces reliability and reintroduces ambiguity.