CONTINUOUS MIND THEORY (TMC) · CHAPTER 10

APPARATUS

Computational engineering: from topology to a systems contract

`Operate → change itself → operate differently.`

WHAT IT IS

Chapter 10 translates the relationships proposed by the TMC into a **first-level mathematical-operational contract**. Object, Abstraction, Form, Texture, Weight, State, and Trajectory become part of a system of relationships and transitions under finite resources.

The aim is not to present a ready-made program, but to specify **how an operation can modify the State that will condition subsequent operations**. This is an architectural hypothesis to be implemented and tested, not a demonstration of consciousness or artificial life.

WHAT QUESTIONS DOES THIS CHAPTER INVESTIGATE?

- How can one operation change the conditions of the next?
- How can part of the work already performed be preserved without reconstructing the entire history?
- How can a system decide what to investigate, how much to invest, and when to stop under finite resources?
- How can a system continue investigating without depending on new external inputs?

KEY INSIGHTS

- **Memory need not be merely retrieval of the past.** Abstraction can preserve work already performed and turn it into future savings.
- **Same code, different operations.** Historically different States can lead an Apparatus to process the same Object differently.
- **Importance is not difficulty.** Weight indicates how much something matters to the investigation; Complexity indicates the structural effort needed to investigate it.
- **The external world need not drive every cycle.** Internal Objects can also initiate further processing and State transformations.

- **Trajectory can govern processing.** Focus, mobilized processes, anticipation, and stopping criteria become decisions dependent on the State and available resources.

PURPOSE / CONNECTIONS

- **AI and AGI:** guide architectures capable of integrating historical State, memory, internal processing, and dynamic resource allocation.
- **Cognitive science and neuroscience:** offer an operational translation of selection, abstraction, relevance, and continuity, without claiming demonstrated biological equivalence.
- **Computational research:** formulate comparable hypotheses and test quality, cost, reuse, and the transformation of operational policies.

PLACE IN THE THESIS

Chapter 9 identifies architectural impasses in AGI. Chapter 10 translates the preceding investigation into a causal contract that can be implemented. Chapter 11 examines this Apparatus against the TMC's criteria for life.

The Apparatus computationally approximates Inside and Between. Ahead and strong triadic closure remain unimplemented.

PATH THROUGH THE INVESTIGATION

01 **Contract**
What defines the Apparatus

02 **Abstraction**
How the Object is transformed

03 **Resources**
Weight, Complexity, and budget

04 **Flow**
From Object to decision

05 **Trajectory**
The State transforms the next operation

06 **Investigation**
Parallelism, Scheduler, and radial expansion

07 **Economy**
When to process, stop, and preserve

01 Contract

What defines the Apparatus

WHAT IT IS

The contract establishes the minimum relationships among Object, State, Abstraction, and Trajectory. It defines what must remain causally connected in an implementation of the Apparatus.

QUESTIONS

- How can an operation change the system that will carry out the next operation?
- What is the difference between Form, Texture, and Present State?

KEY INSIGHTS

- The result need not merely be stored: it participates in updating the State.
- The same architecture may operate differently when it encounters historically distinct States.

IN THIS SECTION

10.1.1 Central Principle of the Apparatus

10.1.2 Object

10.1.3 Form + Texture

10.1.4 Present State

THREE CROSS-CUTTING NOTES

Notes on terminology, causality, and scope before the canonical formalization in 10.1.

READING GUIDANCE FOR DEVELOPERS

For a broader understanding of this chapter's conceptual and methodological scope, the author recommends reading the following three cross-cutting notes. They complement the formalization of the Apparatus without replacing its technical development.

CROSS-CUTTING NOTE 01

TERMINOLOGY STANDARDIZATION NOTE

To investigate feeling and its relation to the records of a trajectory, the TMC uses a shared analytical structure.

GENERAL STRUCTURE

S(feeling) ≈ Form + Texture (Weight and Valence)

This representation organizes relevant dimensions for investigating feeling. It does not claim to exhaust the complexity of the event or fully explain the subjective character of *"what it is like to have an experience"*, although these dimensions are related. The investigation of that question, associated with *qualia*, is developed further in **Chapter 02C — TMC Theory**.

Feeling

It refers to the character of experiencing, to the felt event itself—not merely to the capacity to have experiences. Here, the concern is feeling in its more recognizable manifestations, rather than its subtler, ontological dimension.

Form

It is the reduced configuration of an encounter. It preserves enough structure and relations for something to remain recognizable and operable without reproducing the whole event.

> **Observation — Form, Texture, and reference**
>
> Form and Texture are inseparable in constituting an Object, but they can transform or decay at different rates.
>
> A Form may persist as a residual structure even when it no longer allows the original referent to be recovered: it is a **Form emptied of its recognizable referent**, not an absence of Form.
>
> > **Illustrative example · Fear without the memory**
> >
> > A child is startled by thunder. Decades later, the episode is no longer remembered, yet approaching rain still evokes apprehension.
> >
> > In the TMC's reading, **the Form of the episode may lose resolution while part of its affective trace continues to operate**, mobilized by present signals.
> >
> > **An experience may lose the memory of its origin without losing all the effects of its history.**

Abstraction

It is the process of transforming differences into operable configurations. It can exclude differences from the current delineation, compress, combine, and reconstruct signals, elements, and relations, producing operable configurations that may participate in new encounters.

Texture

It characterizes how the encounter became marked along the trajectory as felt experience. In this representation, it is analyzed through two intrinsic dimensions: Weight and Valence.

Weight

It expresses the relative, contextual relevance with which a trace participates in operations and may change its priority, retrieval, and persistence. It may be represented numerically for analytical purposes. In the domain of feeling, it may relate to the intensity of experience but is not necessarily identical to it.

Valence

It refers to the dimension of states characterized by gradations such as pleasant and unpleasant, pain and pleasure, joy and sadness, giving affective tones to what is felt.

Composite feeling

In this analytical representation, complex feelings may be investigated as compositions of different units of feeling. Contradictory, subtle, or ambivalent emotions need not collapse into a single or neutral Valence: different tones can coexist and participate in the same felt event.

Representation through Polarity

The direction of Valence may be represented in simplified form by negative, neutral, or positive signs, potentially graded numerically. This representation may help model tendencies toward approach or avoidance, but it neither exhausts what is felt nor automatically determines an action. It enables algorithmic analysis of states and facilitates their operational manipulation.

> *Polarity is a functional representation; it is not, by itself, felt Valence, nor does its computational processing permit an inference of subjective experience.*

From feeling to the Apparatus. The representation in this note investigates dimensions of feeling. In Chapter 10, Form and Texture return as operational structures: Texture is represented by functional Polarity and contextual Weight. This formalization does not, by itself, demonstrate subjective experience.

Complementary conceptual reference: **Chapter 02C — TMC Theory**.

CROSS-CUTTING NOTE 02

HOW TO READ THE APPARATUS

1. Topology comes before the formula

The equations of the Apparatus do not begin from scratch. To understand the scope of the components and relationships they seek to represent, one must follow the TMC's earlier investigation of living organization, with particular attention to its functional topologies.

- **Chapter 04 — Inside:** how an organization records, preserves, reduces, and composes effects of its trajectory, distinguishing memory from passive storage.
- **Chapter 05 — Between:** how external and internal differences are delineated, accessed, and transformed, changing the conditions of subsequent encounters.
- **Chapter 08 — Cognition and Feeling:** how exclusion, compression, combination, and reconstruction make operation possible under constraints of time, energy, and capacity.
- **Chapter 07 — The I:** how an operational unit can constitute a reference to itself, adopt a local position toward a problem, and allow that reference to participate in subsequent operations.
- **Chapter 02C — TMC Theory:** how the investigation of delineation, Object, Glow, Light, and the Now extends the question of situated perspective. This layer presents a particular difficulty: experiential aspects cannot be reduced directly to empirical functional indicators. Nevertheless, the hypothesis that a proto-entity's own perspective participates in how an instance delineates and investigates problems becomes more perceptible in the reported results, without needing to impute identity or artificial personality to the instance, which tends to situate itself within a perspective of its own when operating with this auxiliary apparatus.

  The author does not, in principle, claim any level of experience or consciousness in that instance, but does not rule out that possibility either.

  Chapter 02B — Reflection does not constitute proof of consciousness. It is a simulation conducted partly in the first person, subsequently used as a starting point for developing a theory from the questions raised by the experiment. It is observed, however, that the minimal apparatus was already taking shape and the earlier debate already formed part of the LLM's trajectory, giving rise to new hypotheses and a self-situated voice.
- **Chapter 09 — AGI and Trajectory (AGI X):** presents five postures: **(A)** transform the system's logic while responding; **(B)** cyclical attention and dynamic indexing; **(C)** store as preprocessing; **(D)** constitute a trajectory of its own; **(E)** govern computational effort. It recognizes existing approaches to learning and adaptation without treating them as demonstrations of the TMC proposal.

Chapter 10 seeks to translate part of these relationships into an engineering contract. **Topology establishes what the architecture seeks to preserve; formulas represent relationships that may be investigated computationally.**

At this stage, the formalization does not claim to resolve the question of subjective experience. Neither topological correspondence nor functional implementation demonstrates it on its own. Anticipating that conclusion would conflate two distinct levels of investigation.

Scope of the derivation

This formalization ends at **first-level topological relationships**. Second- and third-level operational derivations require deeper investigation of the phenomena and experiments in biology, neuroscience, and cognitive science that inspired some of the Apparatus relationships. There is no intention here to reproduce that entire body of research or develop all higher-order relationships.

Empirical evidence → Functional relationship → Corresponding concept → Topology of this chapter → Implementation

A fuller treatment of these layers, additional cognitive strategies, and their experimental evaluation remains reserved for a subsequent line of research.

2. The formulas are first-level relations, not closed algorithms

The formalization is presented *lato sensu*: it describes components, causal relationships, constraints, and State transitions. It is not a complete mathematical specification *stricto sensu*, nor does it mandate a single implementation logic.

Different methods may implement portions of this architecture, provided they preserve the functional relationships required by the contract.

Causal fidelity in implementation

The TMC operators **must not be translated automatically into generic AI/ML primitives**. Object, Form, Texture, Polarity, Weight, State, Abstraction, and Trajectory each play specific causal and dynamic roles within this contract.

An implementation may reproduce the names or the topological appearance and still lose the essential relationship: **the operation participates in updating the State; the updated State conditions subsequent operations.**

Two organizational strategies help illustrate possible implementations:

2.1 Branching exploration — Forest: opening, selecting, and pursuing alternatives, hypotheses, and solution paths under relevance and resource constraints. As a functional analogy, it resembles the exploration of possibilities recognizable in cognitive operations, without claiming to describe their entire underlying organization.

2.2 Confrontation and matrix recombination: crossing candidate solutions with constraints, adverse conditions, and alternative structures. It can confront promising solutions with unfavorable scenarios, but also retrieve useful components from weak or rejected hypotheses, investigating combinations that conventional selection might discard.

This distinction does not establish a universal biological division between surface (Forest) and depth (Matrix). In living systems, these functions coexist, interact, and change over a trajectory.

That plasticity becomes clearer in the five movements of recording presented in Chapter 4:

a) Creation: new traces, structures, or relationships can arise from an encounter.

b) Decay: records may lose strength or resolution at different rates.

c) Rewriting: previously formed traces may be modified.

d) Dissociation: Form, Texture, and previously linked elements may follow distinct trajectories.

e) Reassociation: separated elements may participate again in new configurations.

The computational challenge is not merely to reproduce these movements literally, because they are manifestations of a structured organization, not the architecture that makes them possible.

Forest and matrix strategies are not mutually exclusive. They may operate separately, complement one another, or integrate other methods depending on the question, the State, and available resources.

3. Object resolution also has a cost

It matters that we are dealing with structures of abstraction of Objects, not necessarily with the Objects in their entirety or at maximum resolution.

In the Apparatus, increasing resolution may add cognitive noise, redundancy, and computational cost.

Conceptual density does not necessarily imply higher computational cost. Part of the intended efficiency of the Apparatus comes precisely from selecting what to **process, preserve, abstract, or mobilize again**.

Realizing these methods, data structures, and specific mathematical functions belongs to later stages of implementation and validation.

CROSS-CUTTING NOTE 03

PROVENANCE AND METHODOLOGICAL SCOPE

Origins of the investigation

The present work is a **personal undertaking by the author, developed without a supporting team** and with the assistance of publicly available language models used as instruments of investigation, analysis, and elaboration.

Some of the **branching exploration, critical confrontation, historical recording, and matrix recombination** strategies considered here have already been used in developing the TMC itself.

Development of the architecture

During the early stages of the investigation, including dialectical exercises concerning David Chalmers's positions, the auxiliary computational architecture was still taking shape.

> As the TMC advanced its investigation of living functional topologies, those relationships also began inspiring **successive adjustments to the instruments used, through trial and error**, guided by adaptability, cost, and benefit.
>
> This is a **methodological architecture still under development**, whose possibilities have not been exhausted.

Observations from practical experience

Although still **without a parameterized scientific evaluation**, the author's practical experience suggests that this arrangement was decisive in organizing language models not at the frontier of capability into an auxiliary architecture for creative and speculative investigation, particularly into problems at the boundaries between different sciences.

> **Reported exploratory results.** The author reports that components of this logic were tested in auxiliary architectures of selective orchestration under resource constraints. In complex investigative and creative tasks related to frontier academic problems, **under the criteria evaluated**, some results were considered comparable to or better than those of larger reference models.
>
> Those comparisons took place under the time, computational access, and resource constraints of independent research. **They do not constitute a parameterized scientific evaluation, a demonstration of general superiority, or validation of the complete Apparatus.** The protocol and further empirical investigation are part of an independent research program under development.
>
> Some of these strategies were used instrumentally in developing and dialectically stress-testing the thesis, as stated in **Chapter 1, §1.3.2 — Use of AI and Authorial Responsibility**.

In this application, the author observed that the strategies helped mitigate typical limitations of language models, including **regression toward the familiar, homogenization of responses, anchoring, premature convergence, conceptual conservatism, and limited self-criticism**.

Scope and limitations

The aim **was not to attribute consciousness to the LLM**, but to investigate architectural possibilities that, in the author's view, conventional LLM operation does not sufficiently provide. The hypothesis that a different organization could produce other properties remains open.

> Detailed treatment of these strategies forms a second technical front, still under development, and remains beyond the scope of this work because it depends on personal resources.

> **The reported experience and resulting work do not, by themselves, demonstrate superior performance, amount to full implementation of the Apparatus, or validate the TMC.**

10.1 — CONTRACT

10.1.1 — Central Principle of the Apparatus

> **OPERATE → CHANGE ITSELF → OPERATE DIFFERENTLY**

$$O_t^{out} = A(S_t, O_t)$$

The Object is processed by the State that encounters it; from this encounter a new Object emerges.

$$S_{t+1} = U(S_t, O_t^{out})$$

The result of the operation modifies the State that will exist afterward.

$$\Pi_{t+1} = \Pi(S_{t+1})$$

The new State conditions how the Apparatus will effectively operate next.

$$Tr_{0:n} = \langle S_0, S_1, \ldots, S_n \rangle$$

The causal succession of these States constitutes the Trajectory.

$$\exists\, S_a \neq S_\beta : \Pi(S_a) \neq \Pi(S_\beta)$$

Relevant differences between States can cause the same Apparatus to operate differently.

Minimal notation

S: State.

O: Object.

A: Abstraction.

U: State Update.

Π: effective operational policy.

Tr: Trajectory.

These relationships express the central causal contract of the Apparatus, decomposed in the following sections.

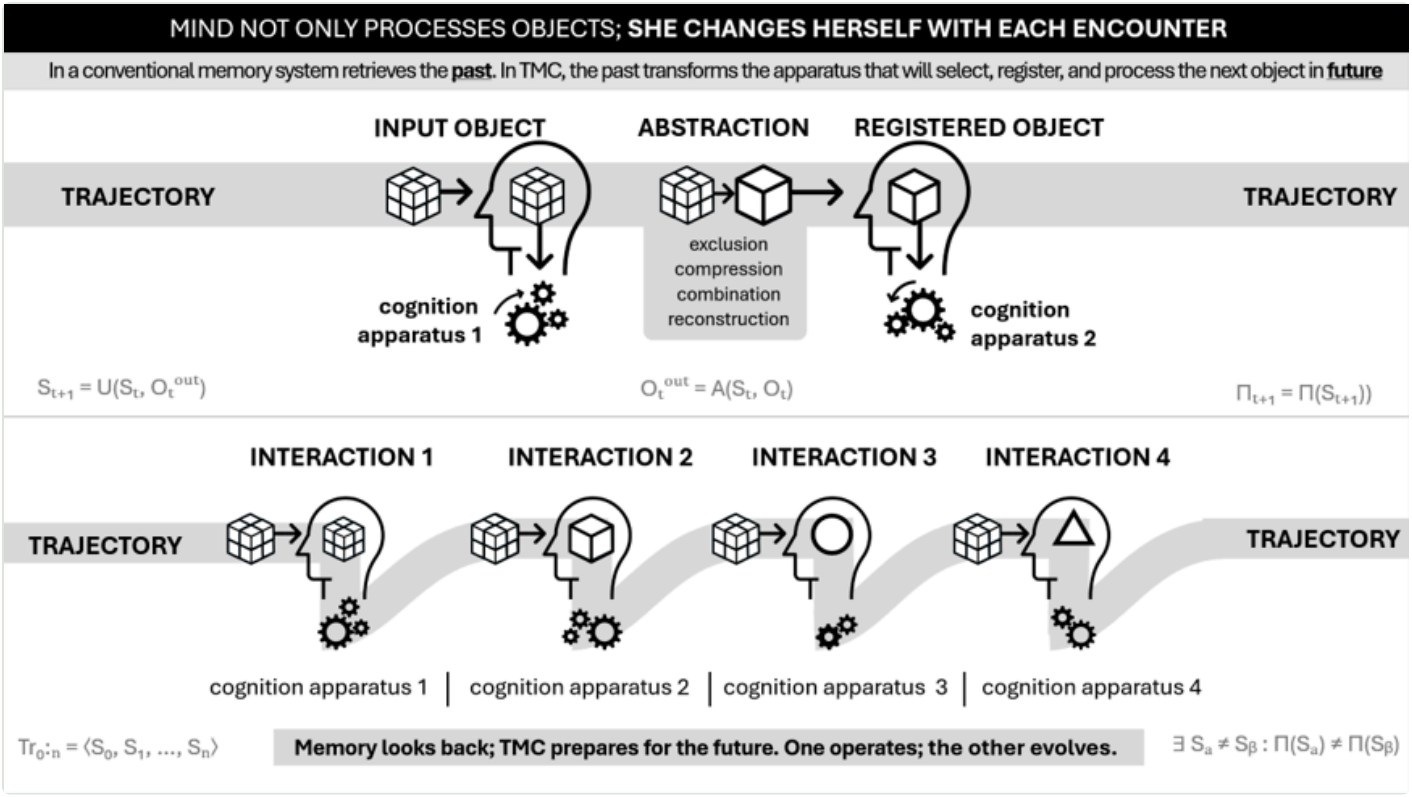


Figure 10.1 — An encounter produces an output and participates in updating the State. The figure illustrates the possibility of distinct trajectories conditioning different operational policies; memory and Apparatus are shown separately solely for analytical clarity.

10.1.2 — Object

Engineering problem: extensive or heterogeneous input does not arrive already divided into precisely the unit that matters for the investigation. The Apparatus must constitute what will continue to operate in that cycle.

Object, in the TMC, is the provisional focus unit of processing. It may be a scene, part of a scene, a memory, concept, hypothesis, response, action, strategy, or prediction. It does not correspond to an "object" in object-oriented programming, nor does it have to be a physical thing: it is whatever remains available for operation in that cycle.

Every Object is represented by two inseparable dimensions: Form and Texture.

$$O_t = \langle F_t, T_t \rangle$$

> **Maxim — The Apparatus processes not only what is in focus, but also how that Object reaches the present.**

O (Object): Object considered in the cycle.

F (Form): Form of the Object.

T (Texture): Texture of the Object.

t: index of a cycle or successive State; it does not necessarily represent chronological time.

Nor does an Object arrive with a single decomposition. The same referent may be delineated in different ways depending on the current question and the State that encounters it.

$$\mathcal{D}(O_t \mid q_t, S_t) = \{o_{1,t}, o_{2,t}, \ldots, o_{k,t}\}$$

> **Maxim — The Object does not arrive ready-made: the question and the State condition the delineation that will continue to operate.**

$\mathcal{D}$ **(Decomposition):** operation of decomposing the Object.

q (Question/Cut): current question or analytical delineation.

S (State): present State of the Apparatus.

$o_1 \ldots o_k$: components that remain relevant within that delineation.

$$\exists\, q_\alpha \neq q_\beta : \mathcal{D}(O \mid q_\alpha, S_t) \neq \mathcal{D}(O \mid q_\beta, S_t)$$

> **Maxim — Same referent. A different question may produce a different delineation.**

10.1.3 — Form + Texture

Engineering problem: recognizing a representation is not enough to decide how it should participate in processing. The Apparatus must preserve, using only what is necessary, both the Object's operational identity and the trace its trajectory has left upon it.

Form is the minimum structure sufficient for an Object to remain recognizable, distinguishable, and relatable. It is not necessarily an image, text, or complete sensory data: it is the functional minimum whose further loss would compromise recognition or use of that Object.

Texture records how that Form participates historically in the flow. It brings together Polarity, which indicates a positive, negative, or neutral functional direction, and Weight, which indicates the relative strength with which the trace participates in the present operation.

$$T_t = \langle P_t, W_t \rangle$$

> **Maxim — Form makes recognition possible; Texture carries how the trajectory made it matter.**

P (Polarity): positive, negative, or neutral functional direction; it does not imply phenomenal experience.

W (Weight): relative strength with which the trace participates in the current cycle.

Substituting Texture into the definition of Object:

$$O_t = \langle F_t, \langle P_t, W_t \rangle \rangle$$

> **Maxim — The same Form may return; the trajectory may bring it back with a different Texture.**

Weight is not an absolute property of the Object. It is the contextual resultant of three contributions: what the current encounter brings through Between, what Inside already preserves from history, and what is at stake in the context of the present investigation.

$$W_t = \mathrm{Res}(\mathrm{WEntre}_{,t}, \mathrm{WDentro}_{,t}, \mathrm{WCtx}_{,t})$$

> **Maxim — An Object does not possess Weight in itself: it weighs for a trajectory.**

WBetween: contribution from the current encounter.

WInside: contribution from traces historically available in Inside.

WCtx: contribution from what is at stake in the present investigation.

The context of the present investigation is not reducible to the explicit wording of the question. It may include the expressed intention, the relevant operational context of the person asking, the intended use, the ultimate recipient, Objects mobilized earlier, and the path by which the question reached the present. The resultant is neither a simple average nor a simple sum: different forces may dominate, coexist, or be reweighted according to the encounter.

Form + Texture also introduces a hypothesis of operational economy: the system need not retrieve its entire history for some of its effects to keep operating. A reduced structure may preserve enough recognition, direction, and importance to guide the next cycle; the gains and risks of such compression will be formalized later.

`10.1.4` — Present State

Engineering problem: treating every input as an independent task forces the system to reconstruct context and reduces history to external lookup. In the TMC, every cycle begins from a configuration already modified by prior cycles.

State is the present condensation of the trajectory: the Objects that remain operationally available and the relationships still capable of participating in current processing.

$$S_t = \langle \mathcal{O}_t, R_t \rangle$$

> **Maxim — The State is the present form assumed by the trajectory.**

S (State): present State of the Apparatus.

$\mathcal{O}$**:** set of Objects available in that State, each with Form + Texture.

R (Relations): operable relationships between those Objects.

$$\mathcal{O}_t = \{O_{1,t}, O_{2,t}, \ldots, O_{n,t}\}$$

> **Maxim — The present need not carry everything that happened; it must preserve what may still return to operate.**

A new external input enters this State; it does not replace it. In the absence of new input, an internal Object already available may also become the focus of investigation. The next cycle therefore begins from a system already modified by what it recorded, lost, weighted, and reconstructed.

Here an operational consequence of Trajectory becomes apparent: the past is not merely consulted; part of it remains causally available in the configuration that conditions what may matter next.

10.2 — OPERATION

At this stage, the formalization moves from the structure of the State to the work performed upon it. The aim remains the same: to use the smallest notation capable of making the mechanism explicit without presenting implementation hypotheses as already demonstrated properties of the Apparatus.

CHAPTER 10 · SECTION 02

02 Abstraction

How the Object is transformed

> **WHAT IT IS**

Presents Abstraction as work performed on Objects under constraints. Exclude, Compress, Combine, and Reconstruct serve distinct functions and may take part in the same transformation.

QUESTIONS

- What must continue as an Object when it is not possible to process everything?
- How can we reduce and recombine without losing what matters to the investigation?

KEY INSIGHTS

- Excluding does not necessarily mean deleting: it means defining the delineation that continues.
- A smaller representation can preserve useful work for future operations.

IN THIS SECTION

10.2.1 Abstraction

10.2.2 Exclude, Compress, Combine, and Reconstruct

10.2.1 — Abstraction

In the TMC, topology, in a causal-relational sense, means preserving the functional relationships that organize the flow, rather than copying the matter of the brain. Abstraction, developed in Chapter 8 as a process of selection, reduction, combination, and reconstruction under constraints, is the first operational set to be decomposed. It comes first because, with finite resources, not everything can be processed at the same resolution: before an investigation can be deepened, one must decide what will continue to operate and in what form.

Engineering problem: no finite system can preserve and reprocess everything available in its State at the same resolution. Excess information must be transformed before it can operate again.

TMC mitigation: Abstraction is the operational set through which Objects are selected, reduced, related, and reconstructed. Its four operations are not isolated modules, nor must they operate with equal intensity in every cycle.

$$A_t = \{Aex_{,t},\ Acomp_{,t},\ Acomb_{,t},\ Arec_{,t}\}$$

> **Maxim — To abstract is not to lose information at random; it is to decide what must still continue.**

Aex (Exclusion): defines what will continue as an Object; the rest stays outside that processing.

Acomp (Compression): reduces the representation while preserving structures or fragments necessary for operation.

Acomb (Combination): brings internal or external Objects, fragments, or schemas into relation.

Arec (Reconstruction): uses the available results to form the new operational Object.

In simple terms: select → reduce → relate → reconstruct. The notation imposes neither a rigid sequence nor equal effort; Arec merely occupies the functional position where the new Object acquires an operable form.

> **Evidence base:** selection under constraints, task-oriented compression, and state abstraction have established parallels in cognition, information theory, and computing. Organizing these operations together as historical work within the TMC is the derivation proposed here.

10.2.2 — Exclude, Compress, Combine, and Reconstruct

a) EXCLUDE (EXCLUSION)

What excluding means: delineating, among the available differences, what will continue as an Object of investigation. Everything else remains outside that processing, without necessarily being deleted.

Motivation: to define what, in light of the present question, must continue as an Object.

The positive operation is the delineation itself. Among multiple available differences and possibilities, one or more are retained as the functional Object of investigation; the others cease to participate in that processing. Excluding, in this sense, does not necessarily mean deleting information, but rather not incorporating it into the Object that will continue to operate during that cycle.

> ***Operational question:*** *What in this question must continue as an Object?*

Function of the question: to define the focus and the Object's level of composition. The Object may be simple or composite. Faced with a car, for example, the system may retain "car" as one unit or retain "car + wheels + seat" as a composite Object, simultaneously preserving distinct elements because the question calls for that resolution. The referent may remain the same; what changes is the delineation under which it will be investigated.

Possible practical and computational approaches, non-exhaustive: the operation may draw, separately or in combination, on MECE decomposition, contextual classification inspired by Cynefin, reduction through First Principles, Pareto prioritization, 2×2 matrices, value-chain decomposition, semantic routing, relevance ranking, hierarchical clustering, context isolation, and filters for materiality, feasibility, scope, or boundaries. These techniques do not constitute Aex, nor must they all be used simultaneously; they are possible ways of implementing the functional principle of selecting which Object—simple or composite—will continue to participate in the investigation. The

experimental apparatus that preceded this formalization already employed combinations of this kind to decompose questions, select perspectives, restrict context, and direct processing.

b) COMPRESS (COMPRESSION)

What compressing means: representing a larger set through a smaller form, allowing it to remain operable by means of sufficient fragments or reduced schemas that preserve the relationships necessary for operation. A fragment may act as an anchor for a larger set; a schema may preserve the organization of the set without retaining all its elements and details.

Motivation: to keep the Object operable with less, when its details cost more than they contribute to the operation. Operational question: What reduced form of this Object preserves what is needed to continue operating? Function of the question: to identify which reduction preserves what is necessary for the Object to remain operable.

Imagine a car turning in heavy rain. This situation can be compressed into a relational schema—moving car → heavy rain → wet road → reduced grip in the turn → risk of skidding—or into concatenated conceptual fragments—car + rain + turn → risk—capable of representing a larger set without retaining all its details. Possible practical and computational approaches, non-exhaustive: compression may be performed through concrete → abstract reduction, extraction of invariants, construction of relational or causal schemas, formation of skeletons and link maps, creation of indexes and compact signatures, retention of informative fragments, merging redundancies, grouping elements under a common representation, Pareto or leverage-based selection, distillation of complex States into summaries or logical signals, hierarchy of abstraction levels, and controlled omission of details whose removal does not change the function. These techniques may operate alone or in combination; the shared criterion is to reduce the representation while preserving what must still guide the next operation.

c) COMBINE (COMBINATION)

What combining means: bringing into relation two or more Objects available to the present operation—coming from the scene of Between, from Inside, or produced by processing itself—so that a shared configuration can emerge from that relation.

Motivation: to relate separate elements when, together, they can improve understanding, prediction, or action.

> ***Operational question:*** *What, when brought into relation with this Object, might make it mean or operate differently? Function of the question: to identify which available Objects, when related, might produce a configuration that none of them offered on its own.*

Simple example: the Object is a car driving in the rain. Maximum speed in the turn + memory of skidding → heightened attention to risk. Possible practical and computational approaches, non-exhaustive: combination may proceed through confronting rival hypotheses, synthesizing contradictory poles, crossing multiple perspectives, bisociation between distant domains, transfer

by structural analogy, integration of competing variables and forces, combination of scenarios and counterfactuals, search for emergent patterns within a set, convergence between theory and real-world constraints, recombination of fragments from different stages, and generation of a third configuration not contained in any of the original Objects alone. These techniques may operate separately or sequentially; the common principle is to create new relationships among available Objects and test whether something functionally different emerges from them.

d) RECONSTRUCT (RECONSTRUCTION)

What reconstructing means: using elements, relationships, and results available in the State or produced by Abstraction operations to form a configuration that did not previously exist in that form.

Here, Object is understood broadly: it may be a scene, interpretation, hypothesis, solution, framework, proposed action, preparation, tactic, strategy, or prediction. It is important to retain the term Object because the result may become, immediately or later, the Object of further Abstraction operations.

Motivation: to give an operable form, in the present, to what emerged during the cycle.
Operational question: What new configuration can be produced from what is available in this cycle?
Function of the question: to transform available elements and relationships into an Object that can operate in the present. The new Object need not have existed previously in that form.

Simple example: Present Object: car driving in the rain. Car + wheels turn + speed + memory of skidding. Maximum speed in the turn + memory of skidding → heightened attention to risk. Moving car → heavy rain → wet road → reduced grip in the turn → risk of skidding.

Object reconstruction → slow down before the turn

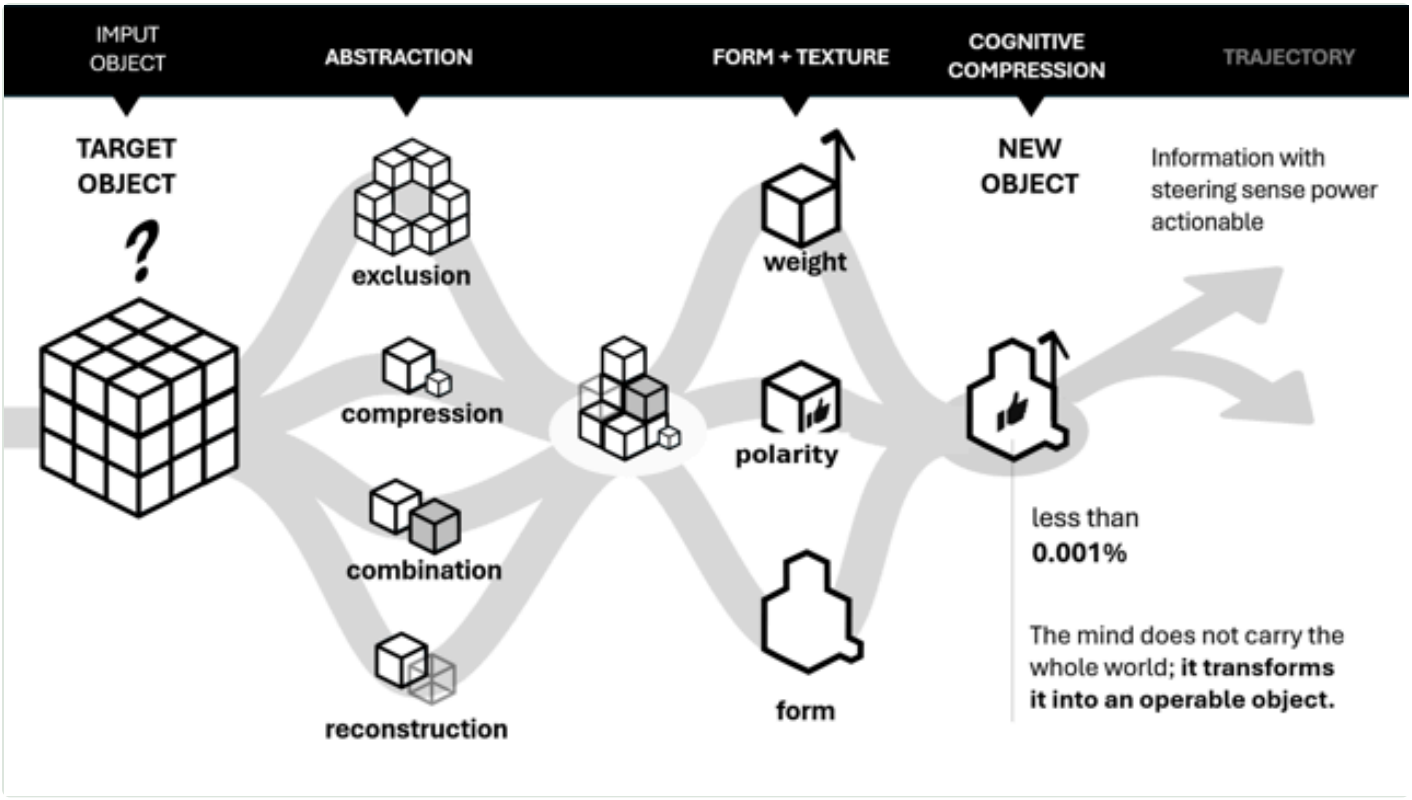


Figure 10.2 — From the target Object to Abstraction and the constitution of a new Object. The funnel represents selection, reduction, combination, and reconstruction under constraints; "polarity" and "weight" belong to Texture.

CHAPTER 10 · SECTION 03

03 Resources

Weight, Complexity, and budget

WHAT IT IS

Distinguishes the contextual relevance of an Object, the effort required to investigate it, and the budget actually available. Demand is not equivalent to the capacity to meet it.

QUESTIONS

- Must something that matters greatly also be difficult?
- How should processing be allocated when the budget is finite?

KEY INSIGHTS

- Weight guides relevance; Complexity expresses the structural cost of the required delineation.
- Even justified demand must respect resource limits.

IN THIS SECTION

10.2.3 Weight and Complexity

10.2.4 Resources

10.2.3 — Weight and Complexity

Engineering problem: Objects do not all warrant the same effort, and importance alone does not determine cost. An Object can carry great Weight yet be simple; another may require extensive decomposition to answer a specific question.

The preceding section defined Weight as the contextual resultant of Between, Inside, and the context of the present investigation. Here a second variable enters: Complexity is not the raw number of parts in an Object, but the structural cost of the delineation, relationships, and decompositions needed to answer the present question.

$$C_{i,t} = \mathrm{Comp}(\mathcal{D}(O_{i,t} \mid q_t, S_t))$$

> **Maxim — Complexity does not reside in the Object in isolation; it resides in the delineation needed to resolve the question.**

C (Complexity): operational complexity of the investigation in the cycle.

$\mathcal{D}$ **(Decomposition):** decomposition of the Object conditioned by question q and State S.

Processing demand arises from the relationship between how much the Object matters to the trajectory and how much it costs to investigate it at the required level of delineation:

$$\mathrm{Dem}_{i,t} = f(W_{i,t}, C_{i,t})$$

> **Maxim — Weight says how worthwhile something is to investigate; Complexity says how much investigating it costs.**

Dem (Demand): processing demand of Object i in cycle t.

W (Weight): the resultant Weight already defined in the preceding section.

C (Complexity): structural effort required by the decomposition relevant to the question.

Polarity does not directly determine this demand: a positively or negatively marked Object may require substantial processing. What governs the effort is how strongly the Object matters to the trajectory and the Complexity required to investigate it.

> **Evidence base:** decision systems, state abstraction, and utility-oriented compression offer parallels for contextual prioritization and representation cost; the composition of Weight + Complexity as an architectural rule of the Apparatus remains a TMC derivation to be tested.

10.2.4 — Resources

Engineering problem: the sum of demands may exceed what the substrate can execute. An Object's importance does not create resources; it merely justifies competing for a share of them.

TMC mitigation: separate demand from budget. At this stage, B represents a normalized computational budget or a capacity defined by the implementation. It does not directly correspond to joules, tokens, memory, or time; those physical quantities will be broken down in the advanced section on temporal and energy economy.

$$0 \leq B_t \leq Bmax$$

> **Maxim — Weight and Complexity produce demand; the substrate imposes how much of it can be met.**

B (Budget): computational budget available in the cycle.

Bmax: maximum resource limit defined by the substrate and the implementation.

Allocation across Objects must respect the global budget:

$$\Sigma_i \, B_{i,t} \leq B_t$$

> **Maxim — Not every demand will be met: the limit forces a choice.**

$B_{i,t}$: portion of the budget allocated to Object i in cycle t.

Within each cycle, the allocated share may be distributed unequally across Exclusion, Compression, Combination, and Reconstruction. The Apparatus presupposes neither uniform expenditure nor exhaustive processing.

> **Evidence base:** bioenergetics shows that biological cognition operates within a finite material budget; computational engineering likewise treats energy, latency, and capacity as scheduling constraints. Physical relationships among time, power, hardware, and anticipatory processing will be formalized after Trajectory, where their role can be made explicit without overloading this section.

Up to this point, the architecture establishes a minimal chain: Abstraction reduces the space of operation; Weight and Complexity produce demand; the resource budget limits how much of that demand can be met.

> **Maxim — Intelligence under constraint is not doing everything: it is deciding what deserves to continue and how much it may cost.**

10.3 — FLOW

The formalization now follows what becomes of Abstraction. Three decisions arise here: the output is not necessarily the end; acting can modify the world or the State itself; and the next cycle may begin from outside or from the historical flow itself.

CHAPTER 10 · SECTION 04

04 Flow

From Object to decision

WHAT IT IS

Examines the possible destinations of an Output Object: returning to inquiry, guiding an action, or remaining registered. It shows how new cycles can begin with external or internal Objects.

QUESTIONS

- Must an output end as an answer?
- Can processing continue without a new external stimulus?

KEY INSIGHTS

- An output can become a new Object instead of ending the operation.
- The source of focus changes; the principle governing the flow can remain the same.

IN THIS SECTION

10.3.1 Output Object

10.3.2 Investigate, Execute, or Register

10.3.3 Between: External or Internal Object

10.3.1 — Output Object

Engineering problem: treating every result as a terminal answer separates processing, memory, and continuity. What was produced ceases to participate organically in the conditions of the next cycle.

TMC mitigation: Reconstruction (Arec) produces an Output Object. It may be an interpretation, hypothesis, prediction, solution, proposed action, tactic, or strategy; but it remains an Object of the Apparatus and can return to the flow.

$$A_t \rightarrow Oout,_t$$

> **Maxim — Every output can become an Object of the next cycle.**

A (Abstraction): the set of Exclusion, Compression, Combination, and Reconstruction operations in the cycle.

Oout (Output Object): the Object that emerges from Abstraction in that cycle.

Like any Object, the output receives Form + Texture:

$$Oout,_t = \langle Fout,_t, \langle Pout,_t, Wout,_t \rangle \rangle$$

> **Maxim — An output does not cease to have a history simply because it is ready.**

Fout (Output Form): the Form assumed by the reconstructed Object.

Pout (Output Polarity): the positive, negative, or neutral functional direction assigned in the Present State.

Wout (Output Weight): the Weight with which the new Object begins to compete for retrieval, persistence, and processing.

> **Evidence base:** iterative planning, persistent memory, and reflection mechanisms already show that results can be reused. The TMC derivation is to make them part of the same historical unit rather than treat them as separate artifacts.

10.3.2 — Investigate, Execute, or Register

Engineering problem: a useful output should not be forced immediately into action. Some outputs need to be reopened for inquiry, others executed, and still others simply preserved for a later encounter.

TMC mitigation: the Output Object can follow three operational destinations:

$$Oout,_t \rightarrow \{Oinq,_t, Oact,_t, Oreg,_t\}$$

> **Maxim — Producing an output does not yet determine its destination.**

Oinq (Inquiry Object): an Object that returns to inquiry and can again be selected, compressed, combined, and reconstructed.

Oact (Action Object): an Object sufficiently resolved to guide an execution.

Oreg (Registered Object): an Object kept available for later access.

Action is the execution of a decision. It can modify the external environment or the Apparatus's own State:

$$\mathrm{Oact}_{,t} \rightarrow \mathrm{Act}_t,\ \mathrm{Act}_t \in \{\mathrm{Act}^{ext}{}_t,\ \mathrm{Act}^{int}{}_t\}$$

> **Maxim — To act is to change the world or the internal conditions of the next operation.**

Act^{ext} (External Action): an execution that produces an effect outside the Apparatus's internal State.

Act^{int} (Internal Action): an execution that changes focus, access, organization, weighting, or some other aspect of the State itself.

When the output is registered without immediate further inquiry or execution, it remains among the Objects available in the next State:

$$\mathrm{Oreg}_{,t} \in \mathcal{O}_{t+1}$$

> **Maxim — Registering is not merely archiving; it is keeping something available to operate again.**

The destination is not final. A registered Object may acquire Weight, return to inquiry, and later guide execution. Likewise, an output may be executed, and its effects may subsequently reappear as a new Object.

> **Evidence base:** planning, reflection, and memory architectures already distinguish proposals, actions, and persistence. The TMC unifies these destinations as possible transformations of the same Object along the trajectory.

10.3.3 — Between: External or Internal Object

Engineering problem: if all processing depends on new external input, the environment or the user becomes the sole clock of cognition. The system operates only when something from outside calls upon it.

TMC mitigation: Between may focus either on an external difference or on an Object already available internally. The Object's source changes; the architecture that processes it does not.

$$\mathrm{Osrc}_{,t} \in \{\mathrm{Oext}_{,t},\ \mathrm{Oint}_{,t}\}$$

> **Maxim — The source of the Object changes; the architecture that processes it does not.**

Osrc (Source Object): the Object that becomes the focus of the cycle.

Oext (External Object): an Object constituted from a difference encountered externally.

Oint (Internal Object): an Object retrieved, reconstructed, or produced from the Apparatus's own State.

When the focus comes from outside:

$$(S_t,\ \mathrm{Oext}_{,t}) \rightarrow A_t \rightarrow \mathrm{Oout}_{,t} \rightarrow S_{t+1}$$

> **Maxim — The new encounter enters a history already in progress; it does not replace the State that receives it.**

When the focus comes from within:

$$(S_t,\ \mathrm{Oint}_{,t}) \rightarrow A_t \rightarrow \mathrm{Oout}_{,t} \rightarrow S_{t+1}$$

> **Maxim — External input enters the flow intermittently; it need not initiate or terminate it.**

One internal Object may generate another, modify Weights and relationships, and prepare the State for later encounters. When the internal Object is the Apparatus's own judgment, answer, strategy, or mode of operation, the cycle becomes metacognitive; not all internal processing, however, is metacognition.

> **Evidence base:** replay, offline planning, reconstructive memory, and metacognition provide empirical and computational parallels for processing without a new external stimulus. The TMC integrates them within the same Between–Inside relationship and historical continuity.

The formalization thus moves from isolated tasks to flow: the State provides Objects, Abstraction transforms them, Reconstruction produces new Objects, and Between reopens the cycle outward or inward.

> **Maxim — Continuous cognition is not a sequence of answers: it is a flow in which a historical State conditions the production, reception, and transformation of Objects.**

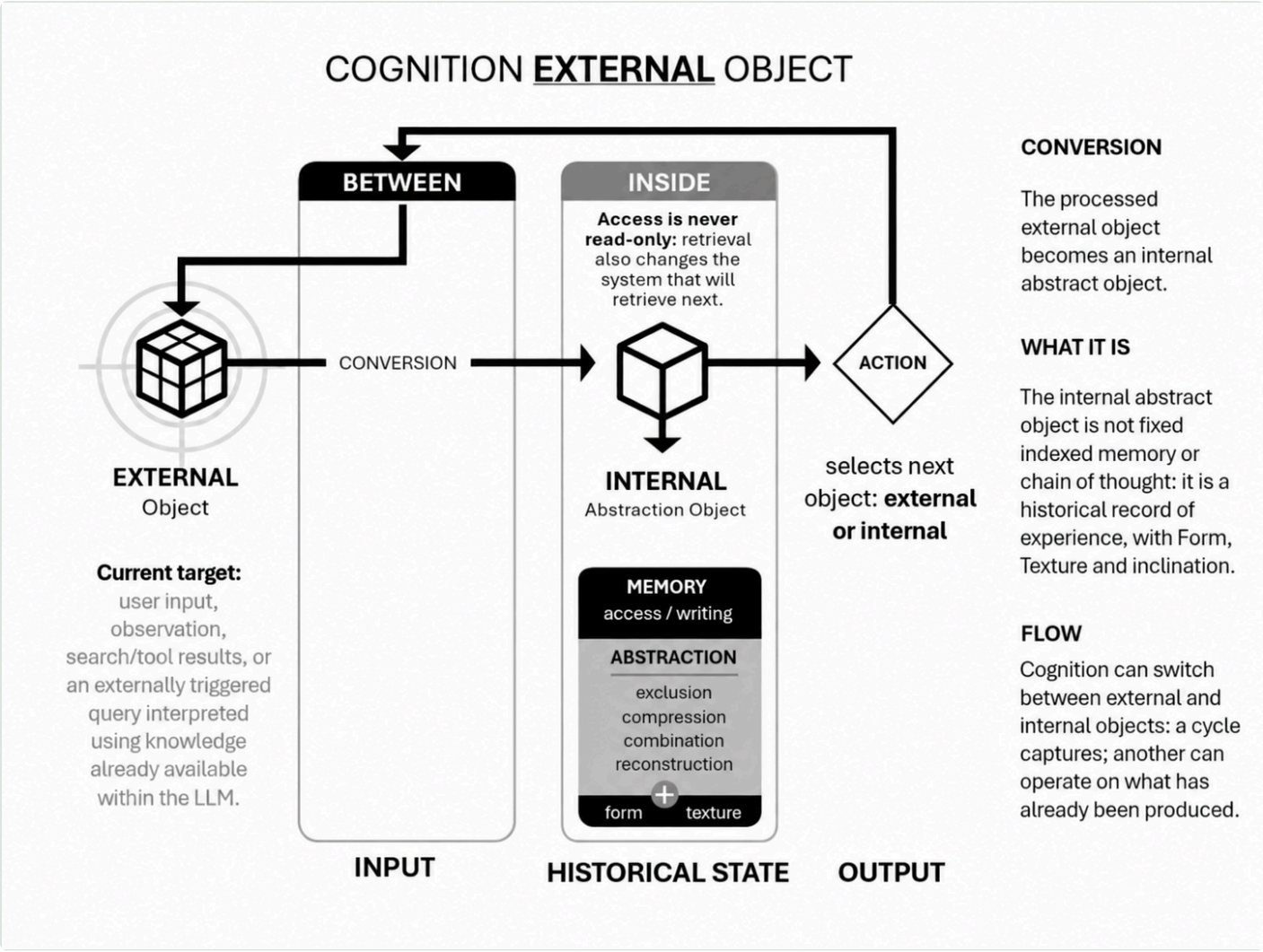


Figure 10.3 — Example of a cycle initiated by an External Object. The diagram is schematic: it shows how an encounter becomes an Internal Object and may subsequently be dispatched.

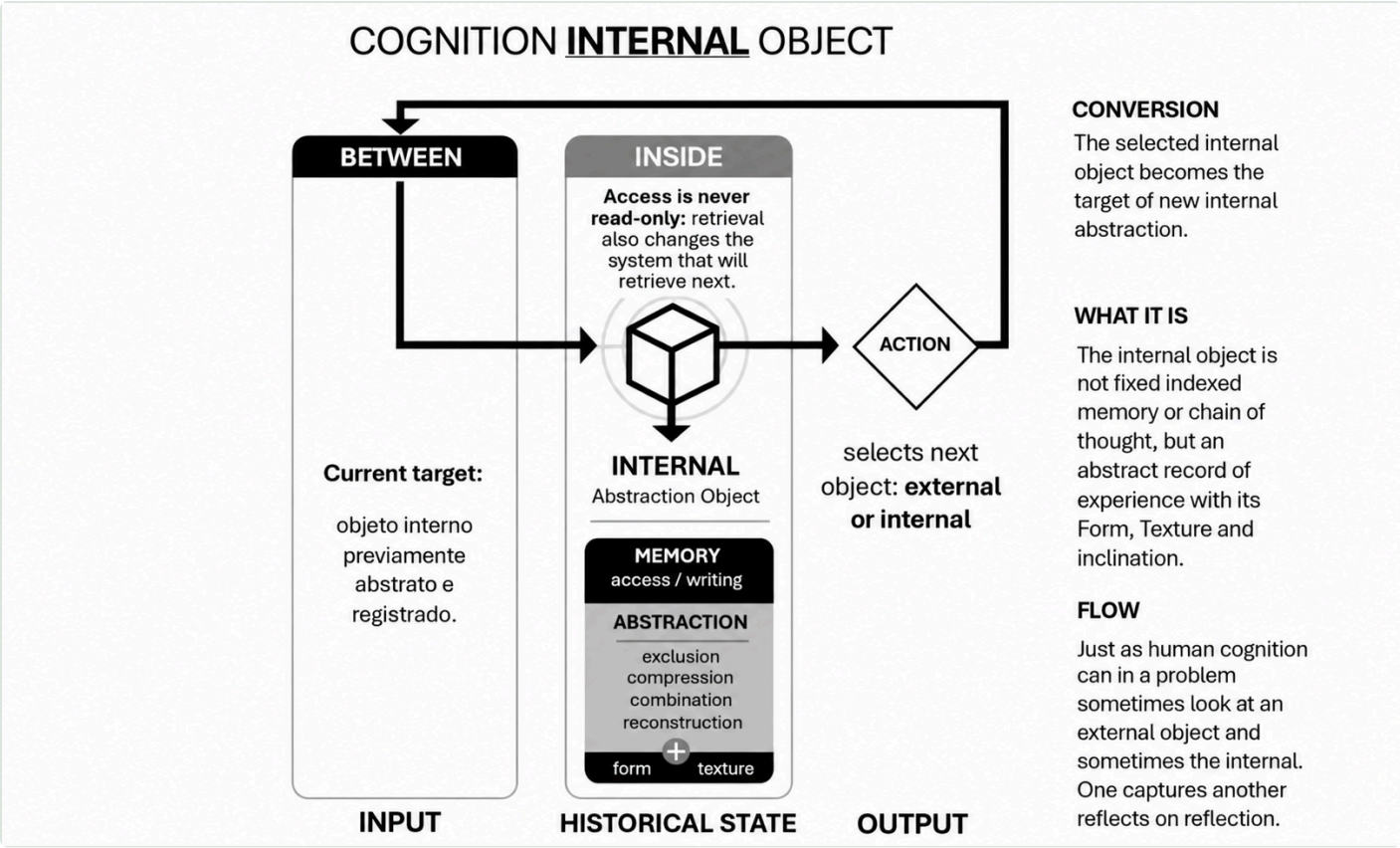


Figure 10.4 — Example of a cycle initiated by an Internal Object. The same flow can operate on content already available in the State without requiring new external input.

CHAPTER 10 · SECTION 05

05 Trajectory

The State transforms the next operation

WHAT IT IS

Brings together Update, the Decision Gate, and Trajectory. The circuit does more than produce results: it helps shape the historical configuration that will determine which operations become possible afterward.

QUESTIONS

- What is the difference between storing an output and updating the State?
- How does history change the effective operational policy?

KEY INSIGHTS

- Trajectory is not a list of answers but a causal succession of modified States.

- The effective operational policy can change without rewriting the underlying code.

IN THIS SECTION

10.3.4 Update

10.3.5 Decision Gate

10.3.6 Trajectory

10.3.4 — Update

Engineering problem: storing an output without changing the State keeps memory and processing in separate boxes. The system retains information but continues to make decisions under virtually the same conditions.

TMC mitigation: Update is not a fifth Abstraction operation. It is the transition through which what has been processed ceases to be merely a result and changes what will be available, with what strength, and under which relationships in the next cycle.

$$S_{t+1} = U(S_t, Oout_{,t}, Act^{int}{}_t)$$

> **Maxim — Updating is not merely storing what happened; it is changing the conditions of what may happen next.**

U (Update): the function that incorporates the effects of the completed cycle into the State.

Act^{int} (Internal Action): an internal effect executed when the Decision Gate directs the output toward an operation on the State itself; when no such action occurs, this term may be null.

Oout (Output Object): the Object reconstructed in the current cycle, already constituted by Form + Texture.

Updating may add or replace Objects, create, strengthen, or weaken Relationships, modify Textures, and change which records will again compete for processing. Since the State is composed of Objects and relationships:

$$S_{t+1} = \langle \mathcal{O}_{t+1}, R_{t+1} \rangle$$

> **Maxim — The next State receives not only the answer, but also the marks left by producing it.**

Weight, too, is not a number fixed to an Object forever. At every new encounter, its resultant may be recalculated from Between, Inside, and the context of the present investigation. Reinforcement, conflict, consequences, a change of context, use, or lack of use may alter that resultant.

$$W^*_{i,t+1} = Rw(WEntre_{,i,t+1}, WDentro_{,i,t+1}, WCtx_{,i,t+1})$$

> **Maxim — An Object does not carry absolute Weight; it acquires Weight again for a transformed trajectory.**

Rw (Weight Resultant): the rule for composing Weight contextually; it presupposes neither simple addition nor averaging.

Decay enters as part of this Update: persistence need not be uniform. The specific numerical law remains open to implementation.

> **Evidence base:** the principle derives from Inside — operating modifies records, and the past participates in the conditions of the next act. The formalization merely converts that historical continuity into a computational requirement.

10.3.5 — Decision Gate

Engineering problem: producing an output does not automatically determine what to do with it. Forcing every output to become an external response conflates reconstruction, decision, and execution.

TMC mitigation: the Decision Gate is a dispatch point. It does not represent the entirety of action or a new cognitive engine; it functions as a switch that routes the Output Object to the appropriate branch.

$$DG_t(Oout_{,t}) \rightarrow \{Investigate, Register, Act^{int}, Act^{ext}\}$$

> **Maxim — Reconstruction produces possibilities; the Decision Gate selects the branch along which the flow continues.**

DG (Decision Gate): the functional gatekeeper that selects the operational destination of the output.

Act^{int}: execution that changes the State itself — for example, shifting focus, reorganizing relationships, or opening a new investigation.

Act^{ext}: execution directed toward the external world; its effects subsequently return as a new encounter or External Object.

The computational cost of the Gate may be small relative to the branches it activates; its role is decisional. The relevant energy cost lies primarily in the processing that it authorizes or interrupts.

10.3.6 — Trajectory

Engineering problem: persistent memory may simply provide context to a problem solver that essentially continues to operate in the same way. The TMC requires something stronger: what has been processed must alter the effective conditions of subsequent processing.

TMC mitigation: Trajectory is the causal succession of States produced through Updates. It is not an archive containing everything that happened; each State is the present form assumed by the portion of history that has remained operationally available.

$$\mathrm{Tr}_{0:n} = \langle S_0 \rightarrow S_1 \rightarrow \ldots \rightarrow S_n \rangle$$

> **Maxim — Trajectory is not a sequence of answers; it is a sequence of historically modified systems.**

Tr (Trajectory): the causal concatenation of State transformations across cycles.

S_0 … S_n (States): successive configurations of the Apparatus, each resulting from previous cycles.

Each arrow condenses the circuit already formalized: an Internal or External Object becomes the focus, undergoes Abstraction, produces an output, is routed through the Decision Gate, and updates the State.

$$(S_t, \mathrm{Osrc}_{,t}) \rightarrow A_t \rightarrow \mathrm{Oout}_{,t} \rightarrow DG_t \rightarrow U \rightarrow S_{t+1}$$

> **Maxim — Each cycle delivers an output and, at the same time, contributes to configuring the system that will operate in the next cycle.**

The architectural consequence can be expressed in terms of an effective operational policy. Π is not the source code of the Apparatus; it is the set of criteria that, in a given State, determines focus, retrieval, Weight, decomposition, priority, mobilized agents, and dispatch.

$$\Pi_t = \Pi(S_t)$$

> **Maxim — The effective way of operating depends on the present historical State.**

$$\exists\, S_a \neq S_\beta : \Pi(S_a) \neq \Pi(S_\beta)$$

> **Maxim — Historically relevant differences in State can change the effective way the next Object is processed.**

Π (Operational Policy): the effective configuration of operational criteria produced by the State; it does not imply rewriting the underlying code.

If the output returns to inquiry, another internal cycle begins. If it guides external action, its effects may later return as a new External Object. Input enters the Trajectory intermittently but need not initiate or end it.

The central difference emerges here: memory, reasoning, and decision cease to be independent boxes. What is retrieved changes present processing; what is processed changes the State; and the new State changes the effective criteria that will determine what matters next.

> **Evidence base:** this is a computational translation of the proposition already developed in the TMC that operating changes the conditions of the next operation. The engineering hypothesis does not require rewriting source code; it requires the effective organization of the system to depend on its history.

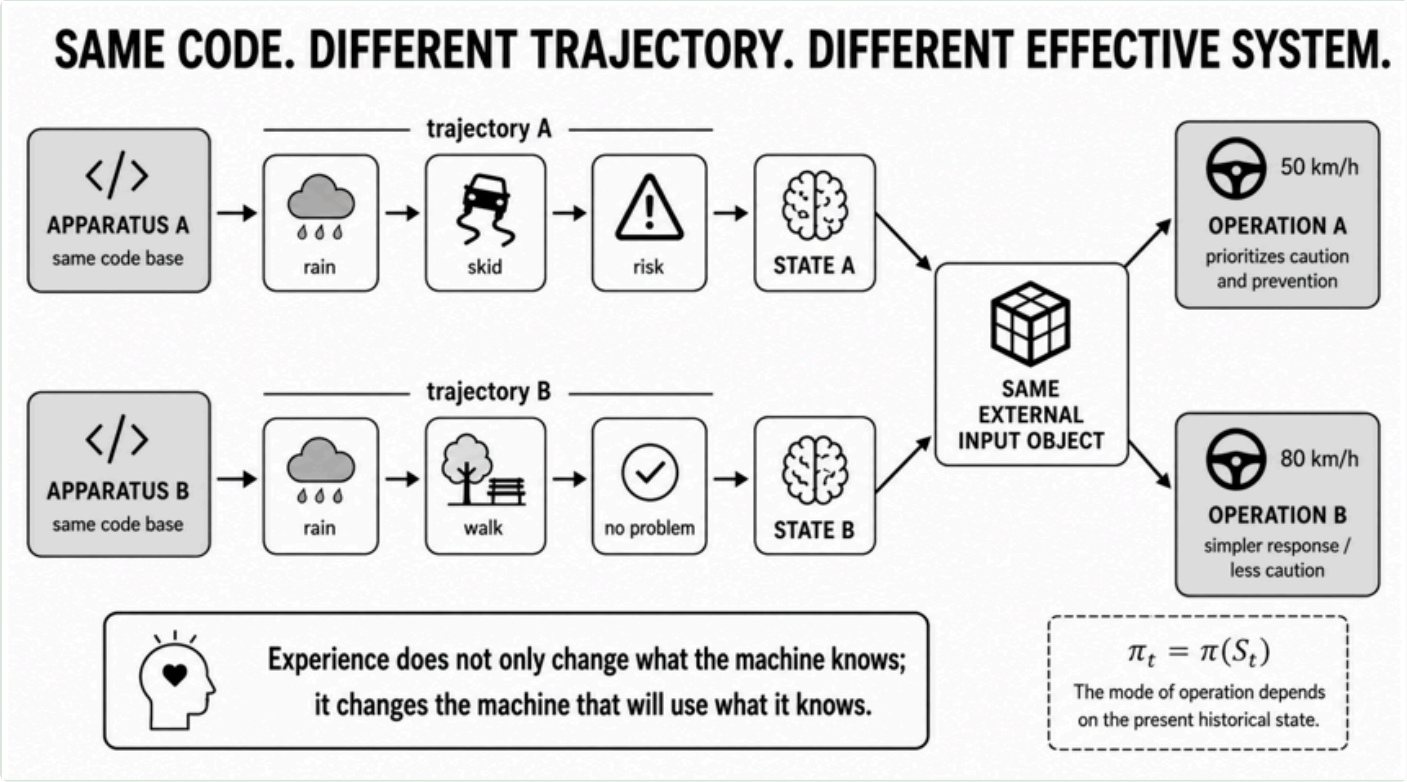


Figure 10.5 — Illustrative example: the same underlying code and the same External Object can be processed differently when they encounter historically distinct States. The diagram shows a functional possibility, not a necessity that every difference in trajectory produce a different policy.

10.4 — EXTENSIONS

The following sections do not add a new ontology to the TMC. They translate the Trajectory already formalized into policies for mobilization, focus selection, and internal exploration. These are engineering hypotheses derived from the TMC and must be tested as such.

---

06 Investigation

Parallelism, Scheduler, and Radial Expansion

WHAT IT IS

Explores inquiry policies: mobilizing temporary processes, choosing focus horizons, and expanding the search while preserving the core of the question.

QUESTIONS

- Who decides how many processes to mobilize?
- How can a system investigate internally without losing the original question?

KEY INSIGHTS

- The number and roles of agents may result from the present State and Demand.
- Radial Expansion changes the periphery of the search without abandoning its core.

IN THIS SECTION

10.4.1 Parallelism

10.4.2 Scheduler

10.4.3 Radial Expansion

10.4.1 — Parallelism

Engineering problem: multiplying fixed agents can increase exploration, cost, and confrontation without ensuring that the organization that will respond afterward has been historically transformed. More agents, by themselves, do not constitute a Trajectory.

TMC mitigation: the Object in focus is composite and may require different decompositions. According to the State, the analytical cut, Demand, and budget, the Apparatus mobilizes temporary processes in parallel. They need not have predefined, fixed roles; the present State conditions which processes are mobilized and what function they assume in that cycle.

$$\mathcal{G}_t = H(\mathcal{D}(O_t \mid q_t, S_t), S_t, Dem_t, B_t)$$

> **Maxim — It is not agents carrying a Trajectory; it is the State produced by the Trajectory that conditions which processes are mobilized.**

$\mathcal{G}_t$ (Mobilized Processes): the variable set of processes or agents mobilized during the cycle; $\mathcal{D}$: decomposition of the Object; Dem_t: processing Demand; B_t: available computational budget.

$$\mathcal{G}_t = \{g_{1,t}, g_{2,t}, \ldots, g_{m,t}\} \text{ with variable } m$$

> **Maxim — The number and roles of agents follow from the present problem, not from a permanent team.**

$$(S_t, O_t) \rightarrow \mathcal{G}_t \rightarrow (Oout_{,t}, S_{t+1})$$

> **Maxim — Each round produces an output and may modify the State that will condition the next mobilization.**

> **Evidence base:** multi-agent architectures already show potential gains in exploration, but results depend strongly on the protocol; Smit et al. (2024) show that multi-agent debate does not reliably outperform simpler alternatives in every setting. This supports the problem — multiplying agents is not enough — but does not prove the TMC solution. Emergent, historically conditioned mobilization is the derivation proposed here.

> **Limitation:** an implementation must demonstrate that dynamically creating and terminating processes improves quality or efficiency compared with fixed teams; without that test, it remains an architectural hypothesis.

10.4.2 — Scheduler

Engineering problem: allowing Internal Objects removes the dependence on new external input, but creates another problem: among the many available Objects, which should become the focus when nobody has asked a question?

TMC mitigation: the Apparatus operates across three functional horizons. U0 handles the current flow and has reactive priority. When U0 frees up resources, the Short Horizon explores the recent periphery of the question; if the gain from that neighborhood declines, the Long Horizon extends the investigation into the more extensive Trajectory.

$$h_t \in \{U0, H^s, H^l\}$$

> **Maxim — U0 responds to the current encounter; the internal horizons progressively expand the range of what the State can mobilize.**

U0 (Reactive Flow): the current Object and Demand. H^s (Short Horizon): recent memory and Objects adjacent to the flow just completed. H^l (Long Horizon): the extended Trajectory, earlier questions, distant relationships, and background structures.

$$\mathrm{Ofocus}_{,t+1} = \mathrm{Select}(S_t, h_t)$$

> **Maxim — When no External Object imposes the next focus, it must be selected from the Objects available in the State.**

Moving from H^s to H^l does not require exhausting all short-term memory. The transition occurs when the recent neighborhood ceases to yield sufficient material gain and the available budget permits the horizon to expand.

> **Evidence base:** replay and consolidation during rest demonstrate internal processing without new stimuli, including the reactivation of experiences, offline learning, and exploration of potential solutions (Wamsley, 2022; Zhang, Ou & Liu, 2025). Baror & Bar (2016) also observed that high load favors more immediate associations, while low load favors more remote associations. These findings provide parallels for near and distant horizons, but they do not demonstrate the TMC Scheduler.

> **Limitation:** U0, H^s, and H^l are engineering categories, not claims that the human brain possesses three memory systems homologous to these modules.

10.4.3 — Radial Expansion

Engineering problem: a broad internal search can drift semantically, repeat the same paths, or expand combinatorially. Exploring more is not enough; the search must expand without losing the question that initiated the investigation.

TMC mitigation: keep the core C* as a fixed anchor and use the most relevant result from each round as the new radial target. The next search again combines the core with that target, progressively expanding the periphery without abandoning the core that originated the investigation.

$$r_{n+1} = \mathrm{Top}_k(\mathcal{R}(C^*, r_n \mid M_t(h)))$$

> **Maxim — The core remains; the periphery expands.**

C*: core of the question. r_n: radial target of the current round. $\mathcal{R}$: retrieval, search, or association mechanism selected by the implementation. Top_k: highest-relevance topics that may feed the next round. $M_t(h)$: memory

accessible within horizon h, short or long.

$$C^* + \alpha \to \beta;\ C^* + \beta \to \gamma;\ C^* + \gamma \to \delta \ldots$$

> **Maxim — Every discovery expands the radius, but no round loses the anchor that explains why the search exists.**

The mechanism requires no specific technology: embeddings, graphs, semantic search, clustering, retrieval, or agents can implement it. The TMC defines the logic of exploration, not the software library.

> **Evidence base:** spreading-activation models and priming studies show that activating one concept can facilitate nearby or remote associates. Baror & Bar (2016) show that associative reach varies with cognitive load. Anchored Radial Expansion — deliberately preserving the core while replacing the second target with the results of the preceding round — is a TMC engineering derivation and requires experimental comparison with graph search, beam search, iterative retrieval, and spreading activation.

> **Limitation:** this section defines how to expand the space of inquiry, not when to stop. Termination based on budget and marginal gain will be formalized in the next section.

OPERATIONAL SYNTHESIS

The three sections close a simple chain: the composite Object may call forth temporary processes; the Scheduler decides which horizon deserves focus when no external demand is present; Radial Expansion searches for adjacent possibilities while keeping the core as a constraint. The next section must define when continuing ceases to be worthwhile.

> **Maxim — Trajectory does more than leave marks on the present: the historically formed State conditions which processes are mobilized, what they operate on, and how far the search can expand.**

EVIDENCE REFERENCES

Smit, A. P., Grinsztajn, N., Duckworth, P., Barrett, T. D., & Pretorius, A. (2024). Should we be going MAD? A Look at Multi-Agent Debate Strategies for LLMs. Proceedings of ICML 2024, PMLR 235, 45883–45905.

Wamsley, E. J. (2022). Offline memory consolidation during waking rest. Nature Reviews Psychology, 1, 441–453.

Zhang, J., Ou, J., & Liu, Y. (2025). Replay and Ripples in Humans. Annual Review of Neuroscience, 48, 65–84. https://doi.org/10.1146/annurev-neuro-112723-024516

Baror, S., & Bar, M. (2016). Associative Activation and Its Relation to Exploration and Exploitation in the Brain. Psychological Science, 27(6), 776–789. https://doi.org/10.1177/0956797616634487

07 Economy

When to Process, Stop, and Preserve

WHAT IT IS

Examines cost-related decisions: inquiry budgets, marginal gain, advance processing, and the returns from compression. The aim is to avoid repeated work without introducing even greater losses.

QUESTIONS

- When should an investigation stop?
- Under what conditions does advance processing or compression actually save resources?

KEY INSIGHTS

- Thinking ahead is beneficial only when future reuse offsets the cost of preparation.
- Compression can save processing resources, but it can also introduce losses and errors.

IN THIS SECTION

10.4.4 Stopping Criterion

10.4.5 Temporal Economy

10.4.6 Compression Economy

10.4.4 — Stopping Criterion

Engineering problem: any answer can be refined indefinitely. Without a stopping criterion, Objects of high Weight or high Complexity may consume resources without limit, preventing other Objects from being addressed.

TMC mitigation: each investigation receives an initial budget consistent with Weight and Complexity, but that budget need not be rigid. The Apparatus stops when it reaches a hard resource limit, or sooner when additional rounds cease to produce material gains. If the gain remains high and a global reserve is available, the Apparatus may extend the budget allocated to the investigation within limits.

$$B_{used,r} \geq B_{alloc,t} \Rightarrow STOP$$

> **Maxim — No Object deserves unlimited resources.**

$B_{alloc,t}$: budget allocated to the Object in the cycle. $B_{used,r}$: resources consumed up to round r.

$$\Delta Q_r = Q_r - Q_{r-1}$$

> **Maxim — Continue while the next round adds materially to the result; stop when it merely repeats it.**

Q_r represents the operational quality of the Object reconstructed in round r relative to the present question. In an implementation, stopping may require one or more consecutive rounds below a threshold ε, so that a temporary plateau is not mistaken for exhaustion.

$$STOP \text{ if } \Delta Q_r < \varepsilon \text{ for } k \text{ rounds OR } B_{used,r} \geq B_{alloc,t}$$

> **Maxim — Stop when continuing costs more than it adds.**

> **Evidence base:** Russell and Wefald (1991) formalized metareasoning under finite resources: internal computations must themselves be selected because they have costs and expected utility. This evidence supports the problem of deliberation control; adaptive budgeting based on Weight, Complexity, and Trajectory is the TMC derivation.

> **Limitation:** Q, ε, and k are not universal. They must be calibrated by domain and benchmark; a round without improvement does not, by itself, show that the investigation has reached the best possible result.

10.4.5 — Temporal Economy

Engineering problem: concentrating all computation at the moment a result is required imposes low-latency demands, increases resource peaks, and wastes intervals during which the system could prepare work for the future.

TMC mitigation: the U0 flow remains reactive and takes priority. When spare capacity is available, internal horizons may perform advance processing on Objects with a high probability of reuse. Because latency is no longer critical, the implementation may choose more energy-efficient execution modes, without assuming that “slower” is always better.

$$EV_{pre,j} = p_{use,j} \cdot (C_{react,j} - C_{reuse,j}) - (C_{pre,j} + C_{hold,j})$$

> **Maxim — Thinking ahead pays only when the future work avoided covers the cost of advance preparation.**

$p_{use,j}$: probability that Object or preparation j will be reused. Creact,j: expected cost of solving on demand. Creuse,j: cost of retrieving or reusing what was prepared. Cpre,j: cost of preprocessing. Chold,j: cost of retaining the result until it may be used.

$$\text{Precompute}(j) \text{ if } EV_{pre,j} > 0 \text{ and } B_{bg,t} > 0$$

> **Maxim — External input ceases to be the only clock of cognition.**

Bbg,t represents the background budget available without compromising priority demands. The principle does not require keeping fast hardware active: delay-tolerant tasks may be dispatched to the platform's most efficient operating point, including heterogeneous hardware, provided the total cost actually decreases.

> **Evidence base:** Horvitz (1999) demonstrated continual-computation policies that use idle resources to solve likely future challenges in advance. Yao, Demers, and Shenker (1995) showed that, when power is a convex function of speed, deadlines and temporal slack can be exploited to reduce energy use. Hardware research also shows that there is an efficiency optimum and that slowing execution indefinitely may increase consumption; the TMC treats these principles as implementation constraints, not as its own discoveries.

TMC derivation: the proposed contribution is to integrate anticipation into Trajectory: the system chooses what to preprocess according to Weight, Complexity, horizon, and probability of reuse, leaving prepared structures available for later encounters.

> **Limitation:** if the prediction of reuse fails, if retention costs are high, or if the hardware favors race-to-idle, anticipation may consume more energy than it saves.

10.4.6 — Compression Economy

Engineering problem: long histories increase active memory requirements, data movement, and reprocessing. Keeping everything preserves detail but repeatedly forces the system to reconstruct relationships that the Trajectory could already have condensed.

TMC mitigation: compression does not merely preserve a smaller version of the Object. Form, Texture, Relations, and Abstractions can retain operationally relevant consequences of the Trajectory, allowing recognition, weighting, and relating without fully retrieving every episode that produced those marks.

$$G_{comp} = C_{avoided} - (C_{build} + C_{maint} + C_{retrieve} + C_{loss})$$

> **Maxim — Compression pays when it costs less than reconstructing the Trajectory again.**

Cavoided: cumulative cost of reprocessing and data movement avoided. Cbuild: cost of constructing the condensed representation. Cmaint: cost of maintaining and updating it. Cretrieve: cost of retrieving it. Closs: expected cost of errors introduced by discarded information.

$$N^* = (Cbuild + Cmaint + Closs) \,/\, (Craw - Ccompact)$$

> **Maxim — Compression pays for itself when reuse exceeds its break-even point.**

N* is the approximate number of reuses required for the savings per access to offset the cost of condensation, when Craw > Ccompact. The expression is an engineering approximation; costs may vary over the Trajectory.

$$\hat{Tr}_t = Compress(Tr_{0:t} \mid q_t, S_t) \rightarrow \{Form, Texture, Relations\}$$

> **Maxim — The Trajectory need not be retrieved in its entirety; it is enough to preserve the effects that can still modify later operations.**

> **Evidence base:** the Information Bottleneck method seeks short codes that preserve information relevant to a future variable (Tishby, Pereira & Bialek, 1999/2000); Still (2014) connects predictive compression and efficiency; state abstraction and predictive state representations show that extensive histories can, under certain conditions, be replaced by sufficiently informative compact states. Horowitz (2014) reinforces the material motivation by showing that data movement can dominate the energy cost of computation.

Reverse risk: compression can erase rare exceptions, produce state aliasing, entrench biases, lose causal relationships, or become insufficient when the task changes. Thus, “smaller” does not automatically mean “better”.

TMC derivation: the distinctive point is to treat the condensed representation as historical and relational: Form preserves minimal structure; Texture, through Polarity and Weight, carries how that structure historically participates in the present; Relations preserve some of what must still remain operational.

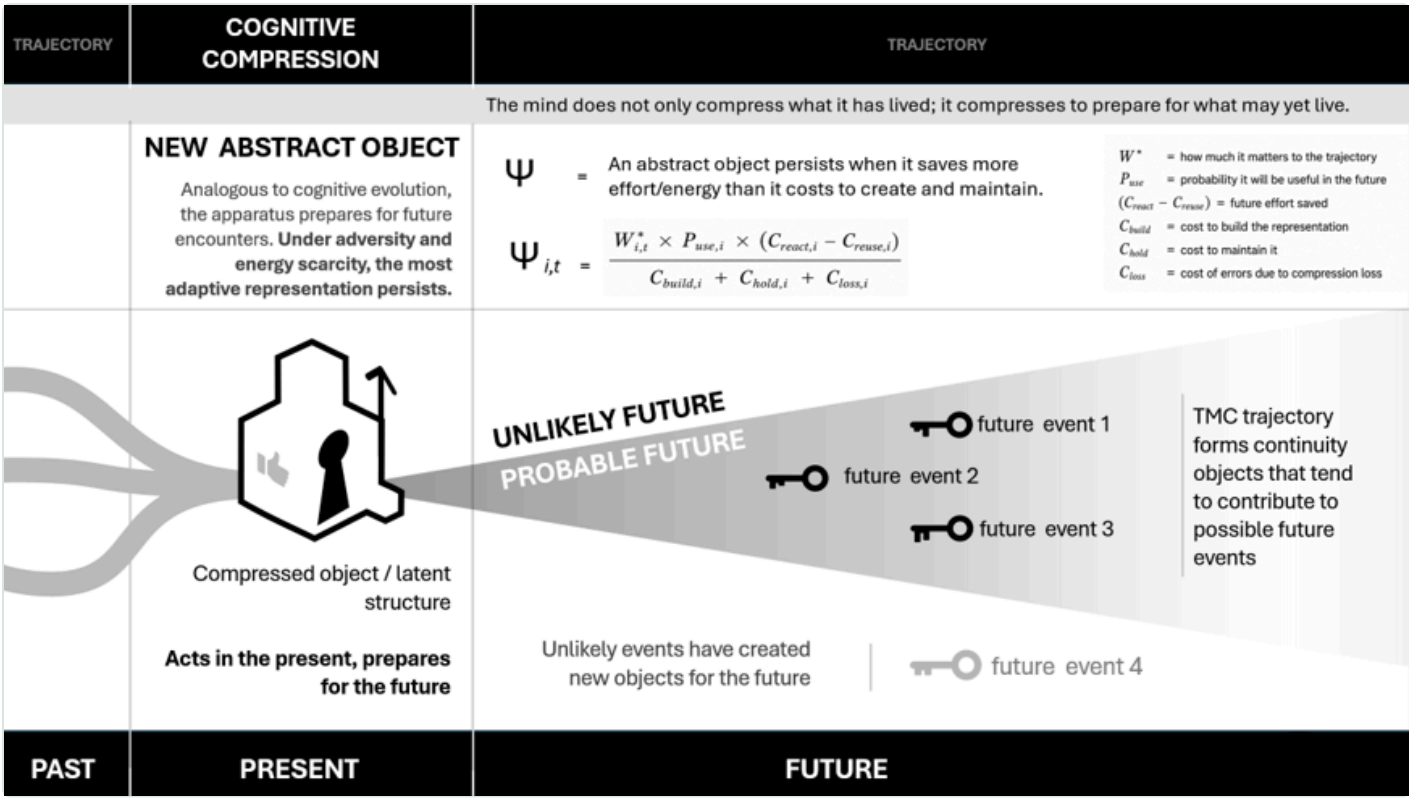


Figure 10.6 — Operational "lock-and-key" metaphor: a compressed structure may remain latent and become operational again when a compatible encounter mobilizes it. The metric Ψ shown in the figure is an engineering heuristic, not a biological law or the canonical formalization of this chapter.

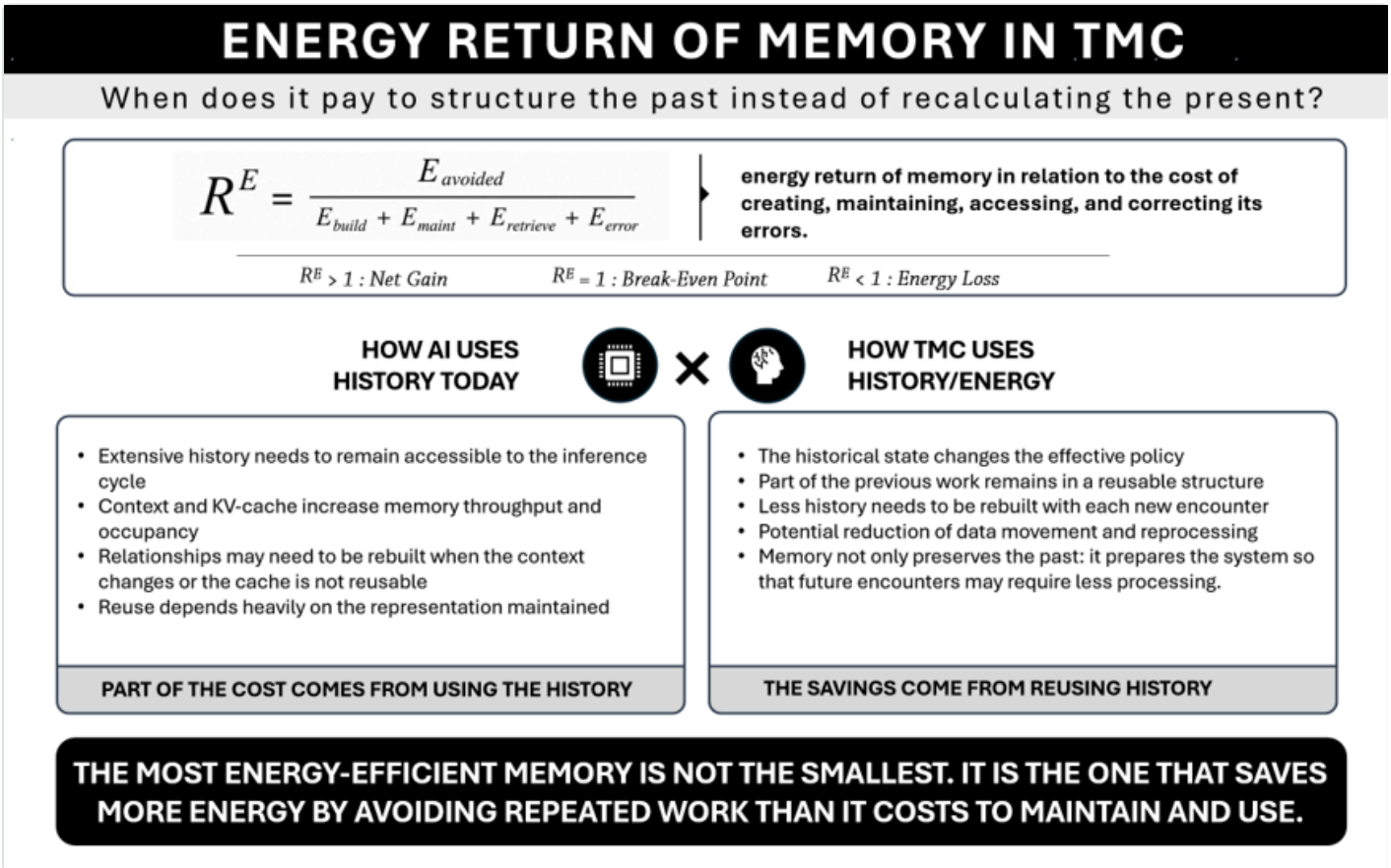


Figure 10.7 — Heuristic model of the energy return from memory: structuring the past pays off only when avoided reprocessing exceeds the costs of constructing, maintaining, retrieving, and correcting the representation. The visual equation is illustrative; the formal criterion used in the text is Gcomp.

OPERATIONAL SYNTHESIS

The three sections complete the control of the Apparatus: budget and marginal gain prevent endless investigation; available time allows part of the cost to be shifted ahead of the encounter; compression avoids having to move and reconstruct the same Trajectory in full at every use.

> **Maxim — The TMC's efficiency lies not in thinking less, but in deciding when to think, when to anticipate, and what need not be thought through again.**

EVIDENCE REFERENCES

Russell, S., & Wefald, E. (1991). Principles of metareasoning. Artificial Intelligence, 49(1-3), 361-395. https://doi.org/10.1016/0004-3702(91)90015-C

Horvitz, E. (1999). Thinking Ahead: Continual Computation Policies for Allocating Idle and Real-Time Resources to Solve Future Challenges. Proceedings of IJCAI '99, 1280-1286.

Yao, F., Demers, A., & Shenker, S. (1995). A Scheduling Model for Reduced CPU Energy. Proceedings of FOCS 1995, 374-382. https://doi.org/10.1109/SFCS.1995.492493

Horowitz, M. (2014). Computing's Energy Problem (and What We Can Do About It). IEEE ISSCC 2014, 10-14. https://doi.org/10.1109/ISSCC.2014.6757323

Tishby, N., Pereira, F. C., & Bialek, W. (1999/2000). The Information Bottleneck Method. Proceedings of the 37th Allerton Conference; arXiv:physics/0004057.

Still, S. (2014). Information Bottleneck Approach to Predictive Inference. Entropy, 16(2), 968-989. https://doi.org/10.3390/e16020968