

HOW TO DESIGN AN EMOTION · WORKING PAPER 4 OF 8

The Constructivist Computational Model of Emotions: Computational Preparation and Interpretive Commitment

Cézjah (Cecil John)

ORCID 0009-0005-1866-9700

New Intellectual LLC, Miami

Correspondence: cezjah@cezjah.com

24 October 2026 · Version 2 (26 October 2026)

DOI: Crossref DOI pending

Also available as https://doi.org/10.5281/zenodo.23676658 (Zenodo)

License: CC BY 4.0 (this paper only; the book is All rights reserved)

newintellectual.com/pages/how-to-design-an-emotion-wp-04

ABSTRACT

Computational models of emotion usually treat emotion as the output of appraisal. I present the Constructivist Computational Model of Emotions (CCME), a small, hypothetical, executable specification that draws the line elsewhere. In the CCME, computation prepares and stops. Predictive simulation, allostatic regulation of the body budget, retrieval of category schemas, and weighting under prediction error narrow the field of possible interpretations, form a situated referent, and rank candidate concepts. Emotion begins only when an agent interprets that referent against a concept's identity conditions under a governing goal, commits to the concept, and marks the commitment with an identifier. I call the transition the meaning boundary. The model separates learned category schemas, which predict and rank, from emotion concepts, which fix admissibility; it separates a default hedonic-regulatory goal, under which a system attributes, from a goal an agent adopts for the act, under which the agent interprets; and it locates error in the commitment, so that a feeling can remain accurate while the interpretation bound to it fails. Two C# programs check these distinctions: one prepares a full trace and constructs no emotion, and one constructs fear or interest from the same snake under different goals. What is new is a hard boundary between computational preparation and interpretive commitment.

The Constructivist Computational Model of Emotions: Computational Preparation and Interpretive Commitment

Cézjah
New Intellectual LLC, Miami
Open Researcher and Contributor ID (ORCID): 0009-0005-1866-9700

Working paper 4 of 8 in the series on the frameworks of *How to Design an Emotion*, October 2026
Digital object identifier (DOI): 10.5281/zenodo.23631056
License: CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/)

Abstract

Computational models of emotion usually treat emotion as the output of appraisal. I present the Constructivist Computational Model of Emotions (CCME), a small, hypothetical, executable specification that draws the line elsewhere. In the CCME, computation prepares and stops. Predictive simulation, allostatic regulation of the body budget, retrieval of category schemas, and weighting under prediction error narrow the field of possible interpretations, form a situated referent, and rank candidate concepts. Emotion begins only when an agent interprets that referent against a concept's identity conditions under a governing goal, commits to the concept, and marks the commitment with an identifier. I call the transition the meaning boundary. The model separates learned category schemas, which predict and rank, from emotion concepts, which fix admissibility; it separates a default hedonic-regulatory goal, under which a system attributes, from a goal an agent adopts for the act, under which the agent interprets; and it locates error in the commitment, so that a feeling can remain accurate while the interpretation bound to it fails. Two C# programs check these distinctions: one prepares a full trace and constructs no emotion, and one constructs fear or interest from the same snake under different goals. What is new is a hard boundary between computational preparation and interpretive commitment.

Keywords: Constructivist Computational Model of Emotions; computational models of emotion; meaning boundary; interpretive commitment; situated referent; category schema; emotion concept; allostasis; predictive processing; appraisal engines; Unified Modeling Language; semiotics; executable specification

Relation to the book. This paper restates, in new prose, Appendix K of *How to Design an Emotion: The Deconstruction and Computation of Emotional Meaning from First Principles* (Cézjah 2026, app. K). The appendix implements the Constructivist Semiotic Model of Emotions of Chapter 9, restated in working paper 3, as an executable specification. A selection of its diagrams, excerpts of its C# code, and their console input and output appear in the appendix to this paper. Every source cited here is cited in the book, with the same locators.

1. Introduction: Where Computation Stops

Can emotion be computed? Two easy answers fail. The first says yes and builds a model that outputs fear as soon as threat-related variables activate. Such a model collapses feeling, simulation, and prediction into emotion, and it cannot say how the same body state could have meant something else. The second says no and treats emotion as beyond formal treatment. It cannot then say what the brain's predictive and regulatory work contributes, or how a theory of emotion could be checked for consistency at all.

The Constructivist Computational Model of Emotions (CCME) takes a third path (Cézjah 2026, app. K). It grants computation everything it does well: simulating situations, preparing the body, retrieving experience, and learning from error. It denies computation one thing: the authority to decide what a feeling means. That decision belongs to an agent who interprets under a goal and commits. The model is a formal, hypothetical specification of functional roles and logical dependencies. It frames emotion as interpretive sign construction under biological constraint, and it makes no claim about neural implementation.

The reason for this division is the book's general account of emotion. If emotion can go wrong, can change under correction, and depends on concepts, then the system that produces it must distinguish processes that simulate or regulate from the act that interprets and commits (Cézjah 2026, chap. 9, app. K). The CCME offers one internally consistent way to formalize that distinction. Section 2 fixes the terms. Sections 3 to 6 follow the architecture from stimulus to commitment. Section 7 describes the two programs that check it. Section 8 places the model among computational models of emotion and among uses of semiotics in software modeling. Sections 9 and 10 mark the scope and conclude.

2. Terms

The model uses several terms at distinct levels, and it does not treat them as interchangeable (Cézjah 2026, app. K).

A **feeling** is a metaperceived interoceptive or somatic state: the conscious registration of bodily affective conditions such as arousal, tension, warmth, or unease. Feelings arise from biological regulation and exist independently of any emotional interpretation. The definition draws support from work that describes feelings as "mental experiences of body states" and ties them to life regulation (Damasio and Carvalho 2013). It departs from that work in one respect. Those authors count fear and anger among the feelings that signal threats to the organism; the CCME treats fear and anger as emotions that interpret such a feeling.

A **situated referent** is a felt state that the system binds to a simulated situation under contextual constraint. It is a special case of the Referent of the Constructivist Semiotic Sign, the situation

toward which an interpreter directs a sign (Cézjah 2026, chap. 7). Here that situation is the world as the agent simulates and affectively registers it, not the world as such.

An **emotion construct** emerges only when the agent commits to an emotion concept under a governing goal and marks the commitment with an identifier. Emotional meaning therefore appears at the moment of commitment, not during activation or simulation.

A **category schema** differs from an **emotion concept**. A schema is a learned predictive structure that organizes prior experience and supports comparison, ranking, and prediction. A concept is an identity-bearing type that encodes the conditions under which an agent can endorse an emotional interpretation. Schemas help the system predict and prioritize; concepts constrain admissibility.

Computation, in the model, includes predictive simulation, allostatic regulation, retrieval of prior experience, prioritization under practical limits, and revision of schemas under prediction error. These processes prepare, bias, and constrain the field of possible interpretations. They set the conditions under which an agent can cross what the model calls the meaning boundary, and they do not confer meaning. The book draws its picture of prediction, allostasis, and categorization from the theory of constructed emotion (Barrett 2017), and it assigns all of them to this preparatory side. That assignment is the model's central departure from the theory it borrows from. For Barrett, prediction and categorization construct the emotion. For the CCME, they construct the field of candidates from which an agent constructs the emotion, and the difference is what lets a construction be checked against identity conditions rather than only against predictive success.

Finally, the model borrows terms from software, such as application server, stored procedures, and queries, as structural metaphors for functional roles. They do not describe neural mechanisms, and the C# code formalizes dependencies and order of execution without implying that the brain runs software.

3. The Architecture

The book presents the architecture in two diagrams with three lanes: stimulus activation, agency, and what it calls the Biological Intelligence Application Server (Cézjah 2026, app. K, figs. AK-1, AK-2). One diagram traces the path on which the agent adopts a goal for the act and interprets. The other traces the path under the default goal, on which the system attributes. Context has no lane of its own. It enters with the stimulus as a structural parameter that shapes simulation, modulates relevance, and changes which schemas memory returns. A database lane appears as a visually separate layer, but it is the server's persistent memory, not an additional domain: it stores learned schemas, stored procedures, and conceptual repertoires.

The flow is short to state. An exteroceptive input reaches the agent within a determinate context. The stimulus–context pair passes to the server, which simulates possible situations, manages the body budget, retrieves past experiences, and weights category schemas under prediction error.

From these operations, the system forms a situated referent. The referent, together with ranked schemas and prior emotion instances, reaches agency. The ranked schemas retrieve associated emotion concepts, which form the candidate set. Only then does interpretation occur: the agent interprets the referent against the candidates' identity conditions under its governing goal, commits to one, and assigns an identifier, which yields an emotion construct.

4. Preparation: Simulation, Regulation, and the Referent

The first operations inside the server set up the field within which interpretation later works (Cézjah 2026, app. K). They generate structured constraint, not emotion.

Simulation builds a situation model. Given a stimulus in a context, the predictive machinery projects what is happening and what comes next: affordances, candidate threats, gradients of relevance, and expected outcomes under the current goal. A snake on a dim trail activates a different simulated landscape from the same snake behind reinforced glass in a zoo. Simulation yields a scenario, not a verdict.

Allostatic regulation runs in parallel. The body budget prepares energetic resources in light of predicted demand, generating arousal, urgency, expected costs, and readiness. In the account the book follows, allostasis "proposes that efficient regulation requires anticipating needs and preparing to satisfy them before they arise" (Sterling 2012, 5). The book extends the point from needs to meanings: the organism does not wait for meaning before it prepares, because preparation precedes interpretation.

Felt awareness emerges from this regulation. Heightened arousal or urgency becomes phenomenally available and biases attention and action tendency. It supplies intensity and valence and leaves reference-marking to the interpretive stage.

Referent formation is the decisive preparatory move. The system binds the feeling to the simulated situation and forms a referent that combines bodily readiness with a projected model of the world. The referent has structure but no interpretation. Because the simulation is probabilistic and the feeling does not interpret itself, error remains possible. A state of high urgency in a dim café can bind to a simulation of danger that turns out to be mistaken, and the readiness can persist even when the simulation misrepresents the world, because the feeling asserts nothing about the situation. In the model, both affect and the felt state are non-propositional.

5. Retrieval, Ranking, and Learning

Once the referent forms, the server queries memory (Cézjah 2026, app. K). At this stage the question is not "Which emotion is this?" but "Which category schemas can plausibly explain this referent under the present goal?"

Category schemas are not mere collections of past episodes. They are concept-governed abstractions from experience, and each encodes predictive commitments: which features co-occur, which costs to expect, which actions tend to follow, and which outcomes have satisfied or frustrated the governing goal. Stored procedures retrieve them systematically and with sensitivity to the goal. Under a goal of avoiding harm, the server weights threat schemas over novelty schemas; under a goal of belonging, interpersonal schemas dominate. The result is a broad candidate set.

Each candidate then undergoes iterative evaluation. The schema generates predicted features and outcomes; the server compares them with the referent and computes prediction error. Schemas that reduce error gain posterior weight, and schemas that fail lose weight or trigger revision. Persistent error leads to refinement, splitting, or the creation of a new category. Learning occurs here, in the structure of the schema network. The organism does not revise its feeling; it revises the conceptual scaffolding that organizes feeling into interpretable patterns.

The server then ranks schemas by posterior probability, which keeps comparison manageable, and retrieves prior emotion instances that match the highest-ranked schemas. Those instances record past meanings and their outcomes. They serve as evidence, not as binding templates. What reaches the agent is a hierarchy of candidates that reflects experience, fit, and relevance to the goal. No determinate emotional interpretation yet exists.

6. Goals and the Meaning Boundary

6.1 Two Kinds of Goal

A goal is never absent in the CCME (Cézjah 2026, app. K). When an agent adopts no goal for the act at hand, a default hedonic-regulatory goal operates: the organism keeps working to maintain viability, reduce disruptive prediction error, preserve energy balance, and stabilize its affective equilibrium. Under that default, the system attributes. Through habit and association, it takes the referent as a sign of what it portends, but it constructs no meaning. When the agent adopts a goal for the act, such as avoiding harm, securing belonging, or pursuing achievement, that goal overrides the default, and interpretation becomes possible. The diagrams mark the switch with a decision labeled Conscious Override, which asks whether the agent adopts a goal for the act.

The diagrams carry a few further labels that the prose does not use, and the book glosses each (Cézjah 2026, app. K). The value judgments the server makes are its evaluations of the situation; they play the role of appraisal and feed interpretation, but they do not interpret the feeling. Recursive Modification and Update Schema name the schema revision described in Section 5. Exercise Rationality names the rational judgment under an adopted goal. Feeling and Fuel (Feelings) name the felt state that the referent binds to the situation, and Construct Emotion Sign names the commitment.

The distinction between the two goals does real work. It explains how much of emotional life can run quickly and habitually without making every habitual response an emotion in the full sense. Under the default goal, a startle at a shape on the trail is attribution: the system treats the shape as a sign of danger and readies the body. The startle is real and causally effective, and nothing in the architecture belittles it. But it carries no commitment, so it is not yet the kind of state that can be correct or mistaken as an interpretation. When the agent adopts a goal, such as getting home safely, the same referent becomes material for an act that can succeed or fail by that goal.

6.2 Crossing the Boundary

The meaning boundary is the transition from computation to commitment. It is a functional distinction within the model, not a place in the brain. On one side lie probabilistic weighting, predictive updating, and optimization for efficiency. On the other lies interpretation.

To cross it, the agent binds a candidate concept to the referent under the governing goal, and the commitment answers to two constraints. The referent must satisfy the concept's identity conditions, and the concept must serve the active goal. An agent can reject a construction that meets the structural criteria but undermines its purposes. The agent asks not only whether the concept fits, but whether it serves.

The book addresses a natural worry about determinism here. On its view, deterministic and probabilistic name different levels rather than rival kinds. A pseudo-random number generator is deterministic code: given the same seed, it yields the same sequence, yet the sequence behaves statistically as random. Biological intelligence behaves likewise under the default goal, lawful in mechanism and probabilistic in what it yields. The rational judgment under an adopted goal then applies a fixed standard to what that process yields, and it is deterministic in the sense that the same goal and the same inputs yield the same verdict (Cézjah 2026, chap. 9, app. K). The judgment is normative rather than computational, which is why a program can model it but not perform it.

When the agent commits, it assigns an identifier. The assignment marks the commitment, which is the interpreter's act of endorsement, and it is not a further constituent of the sign. What began as a field of weighted possibilities becomes a determinate construct: an emotion sign instantiated by binding the concept to the referent under the goal.

6.3 Error and Correction

Because commitment occurs at the boundary, error stays intelligible (Cézjah 2026, app. K). The feeling can remain accurate as bodily registration, and the simulation plausible, while the committed concept fails relative to the outcome. If avoiding a harmless stimulus undermines the agent's purposes, the mismatch becomes an error signal, and the next iteration revises schema

weights or category structure. The feeling remains what it was while the sign system adjusts. The book puts the point briefly: the organism does not correct feeling; it corrects its sign system.

This placement of error has a practical consequence that the book draws out. If emotional error lived in the feeling, correction would require suppressing affect. Because error lives in the commitment, learning occurs through the revision of schemas and category structures rather than through the suppression of affect. In the dim café, the urgency stays what it was. What changes is the weight the system gives to the danger schema in similar settings, and, at the level of agency, the concept the agent is willing to commit to the next time. Retrieval biases candidates, simulation constrains affordances, and posterior weighting keeps the comparison manageable, but none of these operations decides the case.

6.4 The Sign Inside the Program

The narrated trace maps each stage of the program onto a constituent of the Constructivist Semiotic Sign, and the mapping makes the model's debt to the book's semiotics explicit (Cézjah 2026, chap. 7, app. K). The interpreter, with its governing goal, appears first, before any simulation or prediction, so that evaluation never occurs in a goal vacuum. The goal therefore functions as an interpretive constraint, not as an outcome of computation. The situated referent corresponds to the Referent: it integrates the representation of the environment with interoceptive awareness and remains pre-interpretive. Retrieved categories supply candidate interpretations without settling which one the agent will commit to. The concept, fear in the example, is an identity-bearing type that specifies what must be true for an episode to count as fear. The identifier, Fear(snake), marks the commitment the agent made. The construct is the concrete instance of fear that satisfies the concept's identity conditions and integrates the concept, the referent, and the interpretation into an episode with bodily manifestation and an orientation to action.

The mapping matters because it makes plain where each constituent enters and in what order. A model that let retrieval deliver the identifier, or let the referent carry its own concept, would place meaning on the computational side of the boundary. The CCME orders the steps so that the concept, the commitment, and the identifier arrive only after preparation is complete, and only through the agent.

7. Two Executable Specifications

The book checks the architecture with two C# programs and a narrated trace (Cézjah 2026, app. K).

Project 1 implements the preparation layer and stops before commitment. It takes a stimulus and returns a stipulated trace: a simulated situation, a prediction, a body-budget state, a referent, and candidate schemas. The values are fixed in the code; what the program checks is the order of the

steps and the absence of an emotion. It runs under the default hedonic-regulatory goal, so its trace carries no adopted goal. The project is a negative example, deliberately incomplete: even a rich preparatory state remains pre-emotional. A second run with an unfamiliar stimulus, a dog, changes the simulation, prediction, body budget, referent, and candidates, and still refuses to construct an emotion.

Project 2 implements the interpretive layer. It begins from the same referent Project 1 forms: a snake in dim light bound to high arousal. Under the goal of safety, the agent selects the concept fear, assigns the identifier Fear(snake), and orients action toward withdrawal. Under the goal of curiosity, with the same stimulus, the agent selects interest, assigns Interest(snake), and orients action toward cautious approach. With a dog under the goal of safety, the program forms the referent but returns no qualifying concept, no construct, no identifier, and no action, because the input does not satisfy the available conditions. The goal parameter is what crosses the meaning boundary. The agent does not merely process the snake; it constructs the snake as relevant to fear or to interest. If the goal changes, the meaning changes, even when every preparatory step stays the same.

A **narrated trace**, from a separate illustrative program that prints each stage as a first-person report, then walks through the same sequence: the interpreter declares its governing goal before any simulation; simulation models the snake as potentially dangerous in low visibility; prediction expects danger and prepares withdrawal; the body budget raises arousal; the system binds feeling to situation; memory returns candidate categories; the agent selects fear as the concept whose identity conditions the referent satisfies; and only then does the agent commit, assign the identifier, and produce the construct with its bodily manifestation and action orientation. The order is the point. Selecting a concept is not yet committing to it, and the walkthrough keeps the two steps apart.

The book does not overstate what the programs do. They enforce distinctions; they do not predict data. The walkthrough is limited to the path from stimulus to construct and does not model later reinterpretation, which Chapter 9's fifth project addresses (Cézjah 2026, chap. 9).

8. What Is Shared and What Is New

8.1 Among Computational Models of Emotion

Computational models of emotion have a long history, and the CCME joins it late; a review of the field calls for ways to make its many models comparable (Reisenzein et al. 2013). Most existing models are appraisal engines (Cézjah 2026, app. K). The OCC model, which takes its name from its three authors (Clore and Ortony 2013, 336), types emotions by how an agent evaluates events, the actions of agents, and objects, and its authors now read it as compatible with psychological construction (Clore and Ortony 2013). EMotion and Adaptation (EMA) (Marsella and Gratch

2009, 76) computes appraisals over a causal model of an agent's plans and lets coping feed back into them; its authors write that "Coping strategies essentially work in the reverse direction of the appraisal that motivates them" (Marsella and Gratch 2009, 82). The Fuzzy Logic Adaptive Model of Emotions (FLAME) (El-Nasr, Yen, and Ioerger 2000) maps events to emotional states with fuzzy logic and adds learning; its authors write that "The model uses a fuzzy-logic representation to map events and observations to emotional states" (El-Nasr, Yen, and Ioerger 2000). The group of Klaus R. Scherer (b. 1943) models appraisal as a sequence of checks that drive change across several components (Sander, Grandjean, and Scherer 2005; Scherer 2009).

These models are richer than the CCME. They simulate behavior, coping, and time course, and the CCME does not. The difference lies in what each treats as the output. In appraisal engines, emotion is what appraisal computes. In the CCME, computation prepares candidates and stops. Its prediction step stands in for appraisal, recording what the situation means for the agent, such as possible danger, but emotion begins only when an agent selects a concept, binds it to the referent under a goal, and marks the commitment with an identifier. Until then, the program reports that no emotion has been constructed. The model also keeps emotion concepts apart from learned schemas, so that a construction can be checked against identity conditions, not only scored. The book states the comparison in one pair (Cézjah 2026, app. K):

What is shared: emotion as a structured, goal-relative process that can be computed. What is new: a hard boundary between computational preparation and interpretive commitment, with identity conditions built into the model.

8.2 Among Semiotic Uses of the Unified Modeling Language

The appendix also places its modeling method among earlier meetings of semiotics and the Unified Modeling Language (UML), and it orders its differences from least to most consequential (Cézjah 2026, app. K). The least is direction. Most earlier work applies semiotics to UML as a notation or practice: one study reports that "the use of iconic signs as UML graphical notations leads to representation that is more accurately interpreted" (Siau and Tian 2009); another uses semiotic engineering to assess and improve UML as a modeling language (Abdelzad, Lethbridge, and Hosseini 2016); a third maps the signs of an organization onto UML class diagrams (Bonacin, Baranauskas, and Liu 2004). The book reverses the direction and uses UML to model the sign itself.

The second difference is structure. The closest work models the sign of Charles Sanders Peirce (1839–1914) in OntoUML class diagrams (Amaral et al. 2025, 150–51). Its sign is Peirce's triad. The book's sign has five constituents, the Identifier, the Concept, the Construct, the Referent, and the Interpreter with the goal it adopts for the act, so that a class and its instance, the Concept and the Construct, stand inside one sign as distinct constituents. The third difference is level: the book models the sign both as classes and as objects, so that a UML object diagram fixes one act of

interpretation. The strongest difference is execution. Each diagram corresponds to C# code that can pass or fail, so the goal decides whether a single interpretation succeeded, and a construct can fail its concept. The book states that it knows of no other framework that models a sign of these five constituents in UML class and object diagrams, executes those diagrams as tested code, and applies them to emotion.

9. Scope and Limits

The book states the model's main limit plainly: the CCME is small and hypothetical (Cézjah 2026, app. K). Its code enforces distinctions; it does not predict data, and its values are stipulated. Its software vocabulary is metaphor for functional roles, and its meaning boundary is a functional distinction, not an anatomical one. It makes no claim that the brain executes code, and it leaves the neural realization of simulation, regulation, and retrieval to the sciences that study them.

The model also covers less than the book's full account of emotion. It traces the path from stimulus to construct and stops there; reinterpretation, in which the same feeling stabilizes under a different concept, belongs to the Constructivist Semiotic Model of Emotions (Cézjah 2026, chap. 9), restated in working paper 3. The line between appraisal and interpretation, which the CCME draws by assigning appraisal's role to its prediction step, is argued in full in working paper 5 (Cézjah 2026, app. L). And the model does not compete with richer appraisal engines on behavior, coping, or time course; its contribution is the placement of the boundary, not the breadth of the simulation.

10. Conclusion

The CCME divides the labor of emotion between a system and an agent. Biological intelligence supplies readiness and probabilistic structure. Memory supplies conceptual resources shaped by past success and failure. Interpretation binds a concept to a referent under a goal, and commitment makes the binding authoritative. Predictive simulation, allostatic regulation, retrieval, and probabilistic updating narrow and structure the field of possibilities, and none of them has authority over meaning. By isolating the meaning boundary, the model preserves three features any viable theory of emotion has to explain: error and correction, goal relativity, and conceptual identity. Computation prepares. It does not decide. The agent does not calculate meaning; the agent achieves it. The full specification, with its diagrams and code, appears in Appendix K of *How to Design an Emotion* (Cézjah 2026, app. K).

Appendix: Code, UML Diagrams, and Input/Output

The figures below are the book's own diagrams in the Unified Modeling Language (UML), reproduced unchanged. The listings are excerpts from the book's C# companion projects, and the input/output blocks were captured by running those projects with the .NET 8 software development kit; I keep only the lines that bear on the argument. The full programs and the complete transcripts sit beside this paper in `code-samples/`. As in the book, the code checks the consistency of the architecture, not the truth of the theory. The diagrams and programs come from Appendix K (Cézjah 2026, app. K).

A.1 The Architecture

The swimlane diagram places stimulus, agency, and the application server in separate lanes and marks the point at which the agent, not the server, constructs the emotion sign (Section 3).

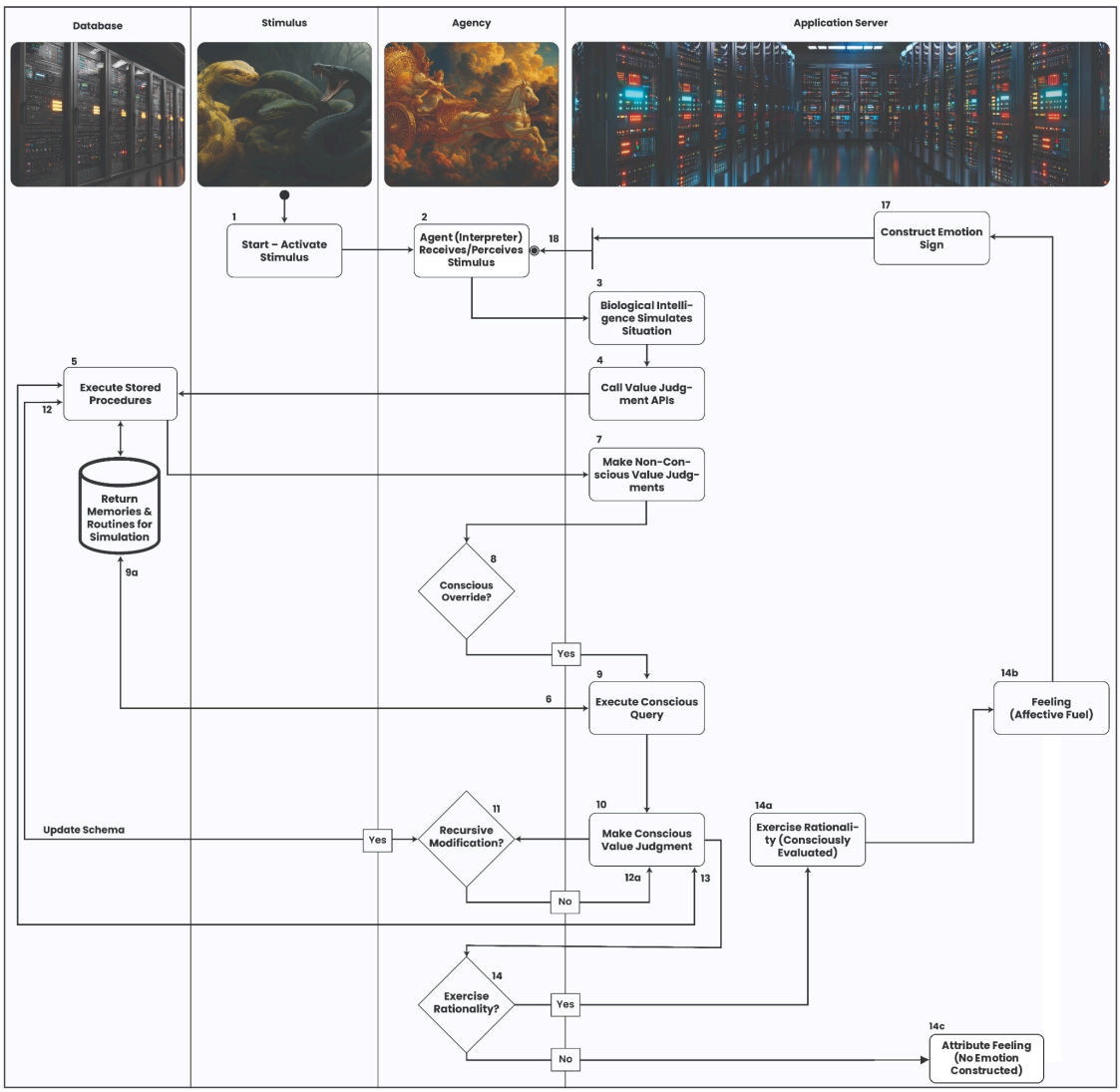


Figure 1. CCME architecture: conscious emotional cognition (book Figure AK-1).

A.2 Preparation Without Meaning (Project 1)

Project 1 runs simulation, prediction, bodily regulation, referent formation, and schema retrieval, and then refuses to construct an emotion because interpretation has not occurred (Section 4).

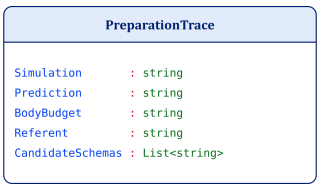


Figure 2. UML class diagram for PreparationTrace (book Figure AK-3).

Listing 1. The preparation trace holds no emotion field (code/appendix-k/project-1/Program.cs, lines 4–11)

```
class PreparationTrace
{
    public string Simulation { get; set; } = "";
    public string Prediction { get; set; } = "";
    public string BodyBudget { get; set; } = "";
    public string Referent { get; set; } = "";
    public List<string> CandidateSchemas { get; set; } = new();
}
```

Listing 2. Input/output: preparation completes, and no emotion is constructed (code/appendix-k/project-1; captured with .NET 8)

```
$ dotnet run -- snake
Simulation: dim trail
Prediction: possible danger
BodyBudget: high arousal, negative valence
Referent: snake-in-dim-light + high arousal
CandidateSchemas: threat, caution, false alarm
Result: No emotion constructed. Interpretation has not yet occurred.
```

A.3 Commitment Under a Goal (Project 2)

Project 2 starts from the same referent. The agent, as Interpreter under a goal, commits to a concept, assigns the identifier, and yields the construct; a change of goal changes the emotion while every prior step stays the same, and an unsupported case yields no construct (Sections 6.1 and 6.2).

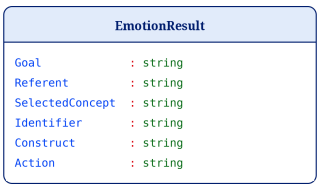


Figure 3. UML class diagram for EmotionResult (book Figure AK-5).

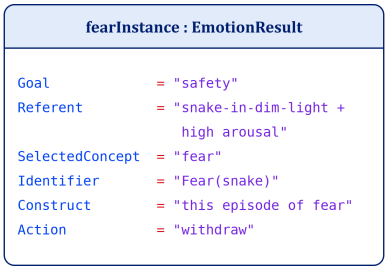


Figure 4. UML object diagram for the fear instance under the goal safety (book Figure AK-6).

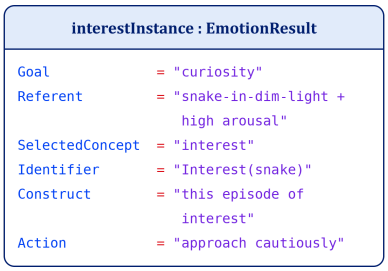


Figure 5. UML object diagram for the interest instance under the goal curiosity (book Figure AK-7).

Listing 3. Goal-relative commitment, identifier, and construct (code/appendix-k/project-2/Program.cs**, lines 41–66)**

```
static EmotionResult Interpret(string stimulus, string goal)
{
    string referent = FormReferent(stimulus);

    (string concept, string action) = (stimulus, goal) switch
    {
        ("snake", "safety") => ("fear", "withdraw"),
        ("snake", "curiosity") => ("interest", "approach cautiously"),
        _ => ("undetermined", "no action")
    };

    if (concept == "undetermined")
    {
        return new EmotionResult
        {
            Goal = goal,
            Referent = referent,
            SelectedConcept = concept,
            Identifier = "None",
            Construct = "None",
            Action = action
        };
    }

    string identifier = $"{char.ToUpperInvariant(concept[0])}{concept.Substring(1)}
        ({stimulus})";
    string construct = $"this episode of {concept}";
// ... (returns the EmotionResult)
```

Listing 4. Input/output: one referent, two goals, and a case with no qualifying concept (`code/appendix-k/project-2`; captured with .NET 8)

```
$ dotnet run -- snake safety
Goal: safety
Referent: snake-in-dim-light + high arousal
SelectedConcept: fear
Identifier: Fear(snake)
Construct: this episode of fear
Action: withdraw

$ dotnet run -- snake curiosity
Goal: curiosity
Referent: snake-in-dim-light + high arousal
SelectedConcept: interest
Identifier: Interest(snake)
Construct: this episode of interest
Action: approach cautiously

$ dotnet run -- dog safety
Goal: safety
Referent: dog-in-context + moderate arousal
SelectedConcept: undetermined
Identifier: None
Construct: None
Action: no action
```

References

Abdelzad, Vahdat, Timothy C. Lethbridge, and Mahmood Hosseini. 2016. “The Role of Semiotic Engineering in Software Engineering.” In *Proceedings of the 5th International Workshop on Theory-Oriented Software Engineering*, 15–21. ACM. https://doi.org/10.1145/2897134.2897136.

Amaral, Marcelo Jaccoud, Vânia Borges, João Luiz Rebelo Moreira, and Maria Luiza M. Campos. 2025. “Applying Principles of Ontology-Driven Conceptual Modeling to the Interpretation Process.” In *Proceedings of the 18th Seminar on Ontology Research in Brazil (ONTOBRAS 2025)*, 147–59. CEUR Workshop Proceedings 4177. CEUR-WS.org. https://ceur-ws.org/Vol-4177/paper11.pdf.

Barrett, Lisa Feldman. 2017. “The Theory of Constructed Emotion: An Active Inference Account of Interoception and Categorization.” *Social Cognitive and Affective Neuroscience* 12 (1): 1–23. https://doi.org/10.1093/scan/nsw154.

Bonacin, Rodrigo, M. Cecília C. Baranauskas, and Kecheng Liu. 2004. “From Ontology Charts to Class Diagrams: Semantic Analysis Aiding Systems Design.” In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, 389–95. SciTePress. https://doi.org/10.5220/0002631103890395.

Cézjah. 2026. *How to Design an Emotion: The Deconstruction and Computation of Emotional Meaning from First Principles*. Universal Moral Concepts III. Miami: New Intellectual LLC. International Standard Book Number (ISBN) 979-8-9898135-9-9 (paperback), 979-8-9898135-3-7 (Kindle). https://doi.org/10.5281/zenodo.23630995.

Clore, Gerald L., and Andrew Ortony. 2013. "Psychological Construction in the OCC Model of Emotion." *Emotion Review* 5 (4): 335–43. https://doi.org/10.1177/1754073913489751.

Damasio, Antonio, and Gil B. Carvalho. 2013. "The Nature of Feelings: Evolutionary and Neurobiological Origins." *Nature Reviews Neuroscience* 14 (2): 143–52. https://doi.org/10.1038/nrn3403.

El-Nasr, Magy Seif, John Yen, and Thomas R. Ioerger. 2000. "FLAME—Fuzzy Logic Adaptive Model of Emotions." *Autonomous Agents and Multi-Agent Systems* 3 (3): 219–57. https://doi.org/10.1023/A:1010030809960.

Marsella, Stacy C., and Jonathan Gratch. 2009. "EMA: A Process Model of Appraisal Dynamics." *Cognitive Systems Research* 10 (1): 70–90. https://doi.org/10.1016/j.cogsys.2008.03.005.

Reisenzein, Rainer, Eva Hudlicka, Mehdi Dastani, Jonathan Gratch, Koen Hindriks, Emiliano Lorini, and John-Jules Ch. Meyer. 2013. "Computational Modeling of Emotion: Toward Improving the Inter- and Intradisciplinary Exchange." *IEEE Transactions on Affective Computing* 4 (3): 246–66. https://doi.org/10.1109/T-AFFC.2013.14.

Sander, David, Didier Grandjean, and Klaus R. Scherer. 2005. "A Systems Approach to Appraisal Mechanisms in Emotion." *Neural Networks* 18 (4): 317–52. https://doi.org/10.1016/j.neunet.2005.03.001.

Scherer, Klaus R. 2009. "Emotions Are Emergent Processes: They Require a Dynamic Computational Architecture." *Philosophical Transactions of the Royal Society B: Biological Sciences* 364 (1535): 3459–74. https://doi.org/10.1098/rstb.2009.0141.

Siau, Keng, and Yuhong Tian. 2009. "A Semiotic Analysis of Unified Modeling Language Graphical Notations." *Requirements Engineering* 14 (1): 15–26. https://doi.org/10.1007/s00766-008-0071-7.

Sterling, Peter. 2012. "Allostasis: A Model of Predictive Regulation." *Physiology and Behavior* 106 (1): 5–15. https://doi.org/10.1016/j.physbeh.2011.06.004.

How to cite

The Constructivist Computational Model of Emotions: Computational Preparation and Interpretive Commitment

APA

Cézjah. (2026). *The Constructivist Computational Model of Emotions: Computational preparation and interpretive commitment* (Working paper 4). New Intellectual LLC. https://newintellectual.com/pages/how-to-design-an-emotion-wp-04 (Crossref DOI pending)

CONTACT THE AUTHOR

Cézjah (Cecil John)
cezjah@cezjah.com
ORCID 0009-0005-1866-9700
newintellectual.com

ALSO AVAILABLE AS

https://doi.org/10.5281/zenodo.23676658 (Zenodo, version 2; all versions: https://doi.org/10.5281/zenodo.23631055)

LINKS

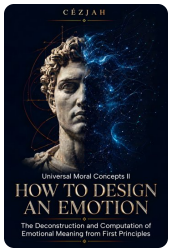
Landing page: newintellectual.com/pages/how-to-design-an-emotion-wp-04

Product page: newintellectual.com/products/working-paper-04-ccme

License: CC BY 4.0 (this paper only; the book is All rights reserved)

KEYWORDS

Constructivist Computational Model of Emotions · computational models of emotion · meaning boundary · interpretive commitment · situated referent · category schema · emotion concept · allostasis · predictive processing · appraisal engines · Unified Modeling Language · semiotics · executable specification



This paper restates, in new prose, Appendix K of *How to Design an Emotion* (Cézjah 2026, app. K). The book is the full treatment.

CC BY 4.0