

Summer School for Systems & Software Engineering in Robotics (SSSER'26)
Delft, June-July 2026



CORESENSE

Developing Cognitive Robots with CoreSense

The CoreSense Process — from epistemic foundations to mission architecture

Ricardo Sanz and Ruth Medeiros

ASLab · UNIVERSIDAD POLITÉCNICA DE MADRID



The CoreSense project has received funding from the European Union's Horizon
Europe research and innovation programme under grant agreement No. 10107054



CORESENSE

Talk content

- So far, in this school, you have learned many **robotics software technologies** and tools to use them in your projects
 - With many problems still to be solved
- Today, we will address the task of scaling up these assets, to very complex robotics controllers, using a **systems engineering** approach
- In my talk I will tell you about the cognitive robotics developments that we are doing in the EU-funded **CoreSense project**
- Hopefully, it will provide some **concepts** on how to address **complex robot minds** construction, using a systems engineering approach



CORESENSE

Motivation

- There are cognition **problems yet to be solved**
- The construction of complex cognitive robots **does not scale well**
- We need **better methods, designs, assets**
- We approach this using an **architecture-centric approach**
- We need:
 - Better Methods
 - Adaptable Designs
 - Reusable Assets
- The CORESENSE EU Project
- The central idea: **MB ASys SE using the “model continuity principle”**



CORESENSE

PART 0 · MOTIVATIONS

The Good Robot

The reasons for seeking machine understanding.

What is **the most important property** of a robot?

Robot (System) qualities that matter



CORESENSE



Human qualities that matter



CORESENSE



Yes ...

Cuteness

Yes ...

Effectiveness

Yes ...

Understanding

Yes ...

Awareness



making sense
of the world



CORESENSE

The brittleness of shallow autonomy

Current autonomous systems rely on pre-programmed responses and shallow pattern matching. In open-ended, unpredictable environments this creates a critical vulnerability.



The catastrophic failure gap

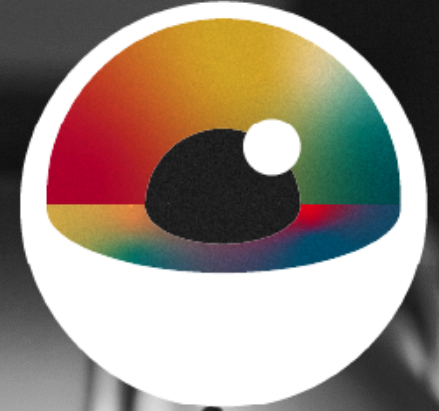
When a robot meets an unmodelled edge case it cannot adapt. Errors about the environment — or the robot's own state — lead to mission failure.



The need for dependability

True autonomy requires systems that safely navigate shifting contexts, unexpected human interactions, non parametric disturbances, and internal hardware degradation.

To achieve operational dependability, robots must move from merely reacting to actively understanding themselves, others, and their environment.



CORESENSE

CORESENSE Project

Develop a **cognitive architecture** for **deep understanding**



Funded by
the European Union

*In CORESENSE, we seek
an “understanding” and “awareness”
technology to make better autonomous systems*

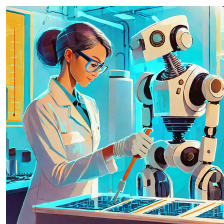
*Robots that see and understand.
Robots that are aware of what is going on.
Robots that can perform meaningful actions.*



CORESENSE

Applications vs Technology

social interaction
humanoids
alignment



aerial inspection
drones
resilience



factory handling
manipulators
flexibility



knowledge
modeling
mission
change
uncertainty
awareness
self
adaptation
explainability
dependability
constructability
integration
....

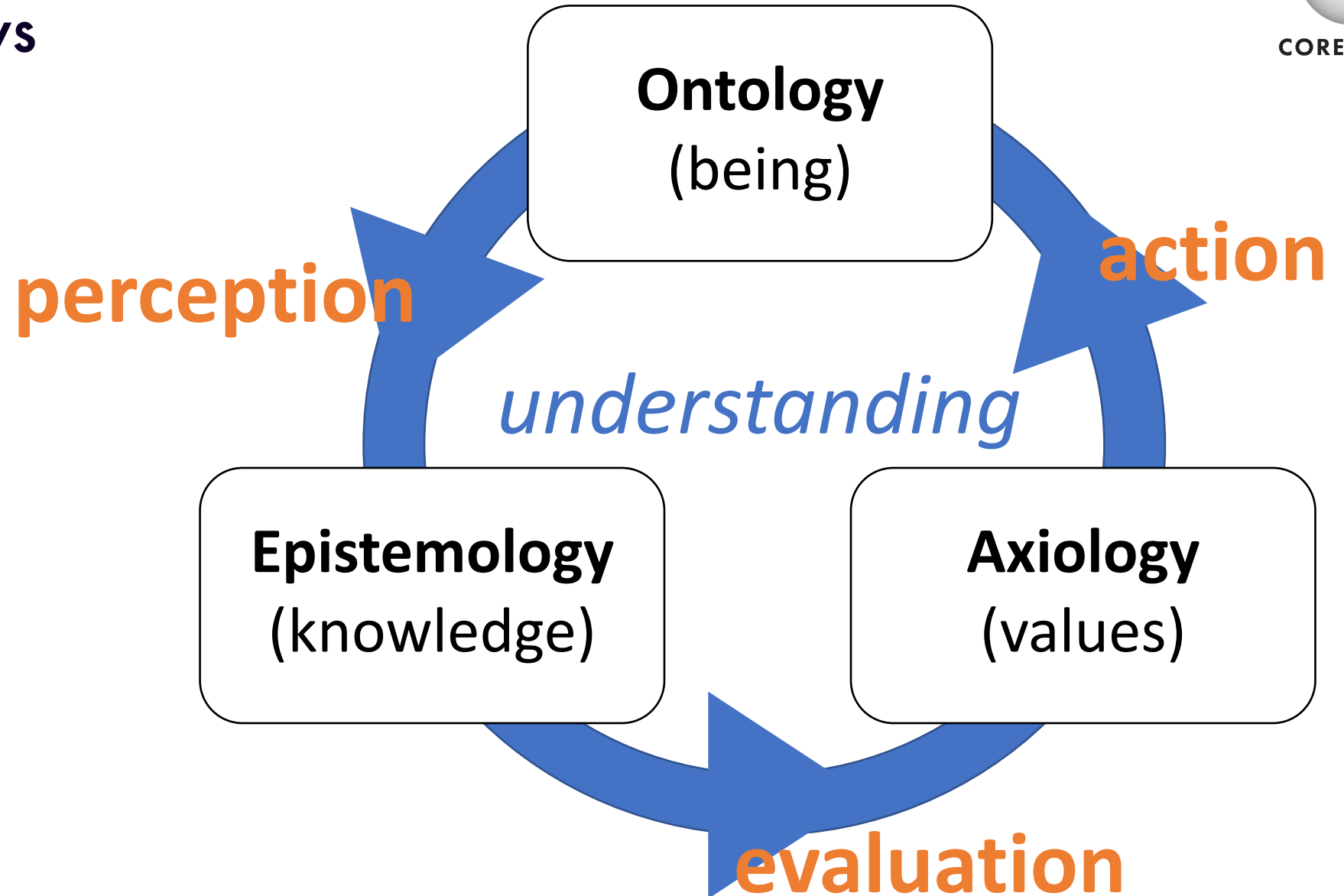


theory
architecture
engineering



CORESENSE

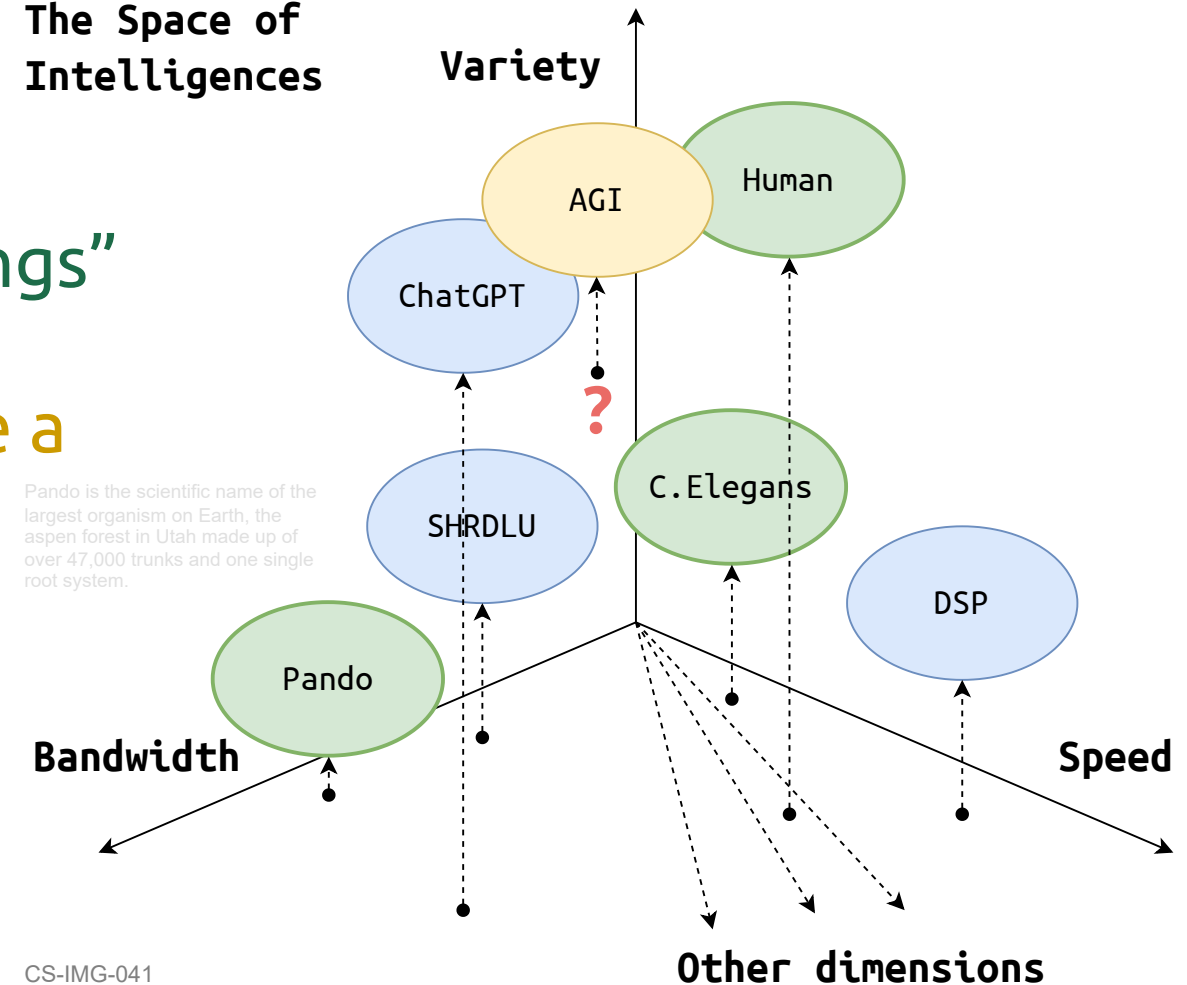
Three key views



One size does not fit all

- Protagoras was wrong:
"man is **not** the measure of all things"
- Cognition for robots does not have a single profile.
- We need technology to create bespoke cognitive architectures for autonomous robots

The Space of Intelligences





CORESENSE

The CoreSense purpose

- We see **missing capabilities** in robots and scattered, non-systematic methods to address them (i.e. missing capabilities in engineering)
- We think that it is necessary to get distance from the concrete problems to **have a bird eye view of the domain of autonomous systems** (esp. robots)
- We intend to provide:
 - A **general but rigorous domain conceptualisation**
 - Reusable **system architecture design patterns**
 - **Reusable software** implementations of key elements

understanding
awareness



CORESENSE

Content



Part 0 • Motivation — The Good Robot

The reasons for seeking machine understanding.

1

Part I • Foundations — The Cognitive Blueprint

The epistemology of machine understanding — why models, not data, are the basis of cognition.

2

Part II • The MBASE Approach — Engineering the Glass Box

Formal definitions of understanding and awareness — towards the CoreSense cognitive architecture.

3

Part III • Engineering Aware Systems — Model-Based Autonomous Systems Engineering

From MBSE to MBSA — the reference architecture, cognitive modules, and the epistemic control loop.

4

Part IV • Mission Architecture in Practice — The CoreSense Mission Architecture Guide

A standards-grounded method for architecting cognitive systems, with a worked example.



CORESENSE



PART 1 · FOUNDATIONS

The CoreSense Cognitive Blueprint

The epistemology of machine understanding:
why **models**, not data, are the basis of cognition.

The illusion of rote learning

Black Box

Current Deep ML

- Brittleness in open worlds
- Statistical rather than causal
- Lacks true awareness

The CoreSense Approach

Deep Models and Inference

- Model-Based Understanding
- Causal reasoning
- Robustness and Resilience

To prevent catastrophic failure in autonomous robots, we must move from rote, reflex-driven behaviours to rational, goal-oriented decision-making.

Three strategies for machine understanding



Mimesis

Attempting to mimic human biological understanding. Flawed and historically ineffective (e.g. Searle's critique or AI winter).



Specificity

Pure machine logic, creating a non-human machinic intentionality. Leads to isolated anthropo-semantics vs. robo-semantics.



Generality

The CoreSense approach. Universal semantics — a shared framework embracing both human and machine understanding to serve human needs.

Understanding is greater than knowledge

The Received View

Understanding = Knowledge

The standard view: understanding is simply knowing, especially the causes (deductive-nomological).

The Revisionist View

Understanding > Knowledge

The CoreSense stance. Understanding requires a cognitive achievement in actual capability — (mission-oriented) **actionability** of knowledge.

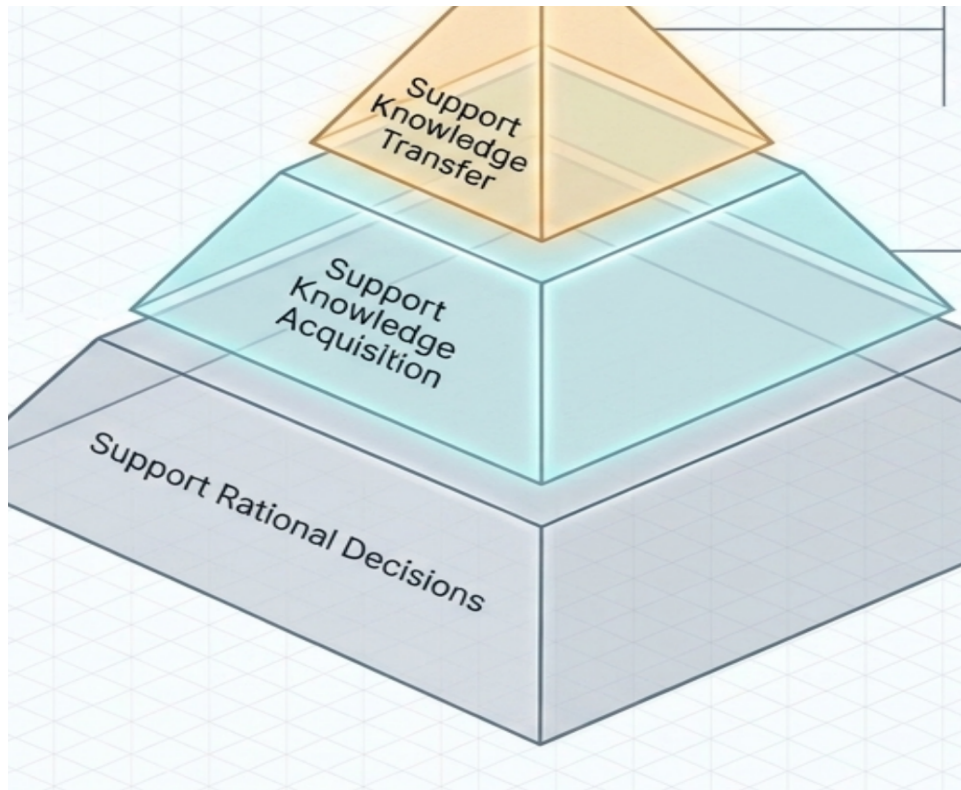
Takeaway. Mere propositional knowledge (knowing that) is insufficient. True robot autonomy requires an actionable ability to derive meaning and project future states.

Understanding and Knowledge vs Value

Some Classes of machine understanding

Class of Understanding	Grammar	Robot Context & Utility
Propositional	<i>That</i>	Acknowledging a fact (“I know that the temperature is rising”). Utility: Low — partial prediction without much decision capability.
Objectual	<i>What</i>	Grasping a domain or artefact (“I understand this object and how it works”). Utility: Medium — necessary to exploit object affordances.
Executable	<i>How</i>	Explanatory causal mapping (“I understand why the fluid is boiling and how to stop it”). Utility: High — the primary target for autonomous decision-making.

The practical utility of rational agency



Support Rational Decisions

Moving from reflex behaviours (stimulus-action) to bounded rationality. Optimising expected results by projecting future states.

Support Knowledge Acquisition

Metacognition. Recognising the robot's own degree of understanding, identifying information gaps, and actively gathering data or reasoning to resolve conflicts.

Support Knowledge Transfer

Shared meaning. Accelerating social adaptation by generating trust, providing maintainable explanations, and facilitating efficient human-robot collaboration.

Assessing some existing cognitive architectures

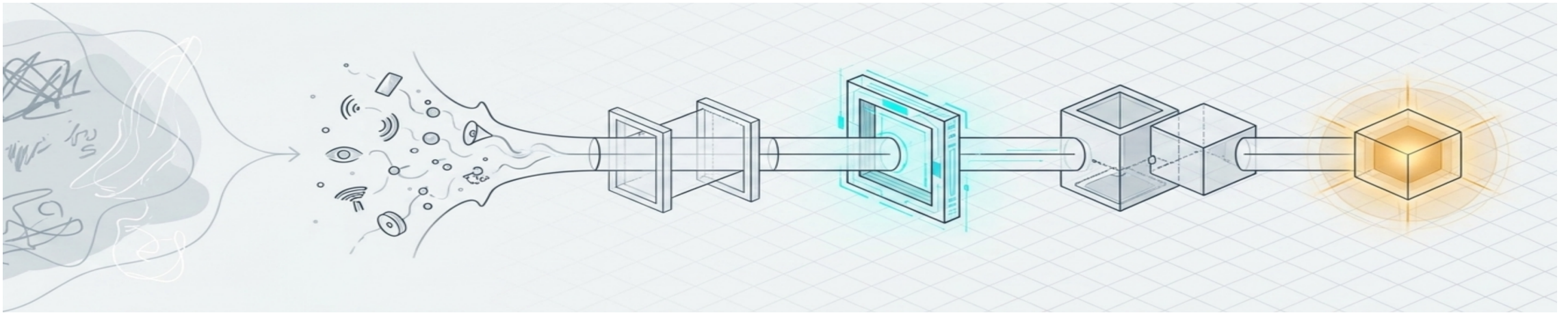
	SOAR	ACT-R	CRAM	4D/RCS
Knowledge Representation	Symbolic / Rule-based	Declarative / Procedural	Semantic (KnowRob)	Environmental models
Handling of Understanding	Chunking / Generalisation	Neurobiological simulation	Underdetermined action resolution	Situational awareness focus
Real-Time Perception / Action	Integrated	Time-based	Sensorimotor grounding	Multilevel continuous planning

Analysis. None of these architectures explicitly isolate “understanding” as a core mechanizable layer, though 4D/RCS and CRAM approximate it through deep situational awareness and semantic knowledge bases.

The CoreSense path to awareness



CORESENSE



World

The external, dynamic environment.

Sensing

Exteroceptive / proprioceptive data gathering.

Perceiving

Interpretation and context integration.

Understanding

Application of deep models to grasp causal relations.

Meaning

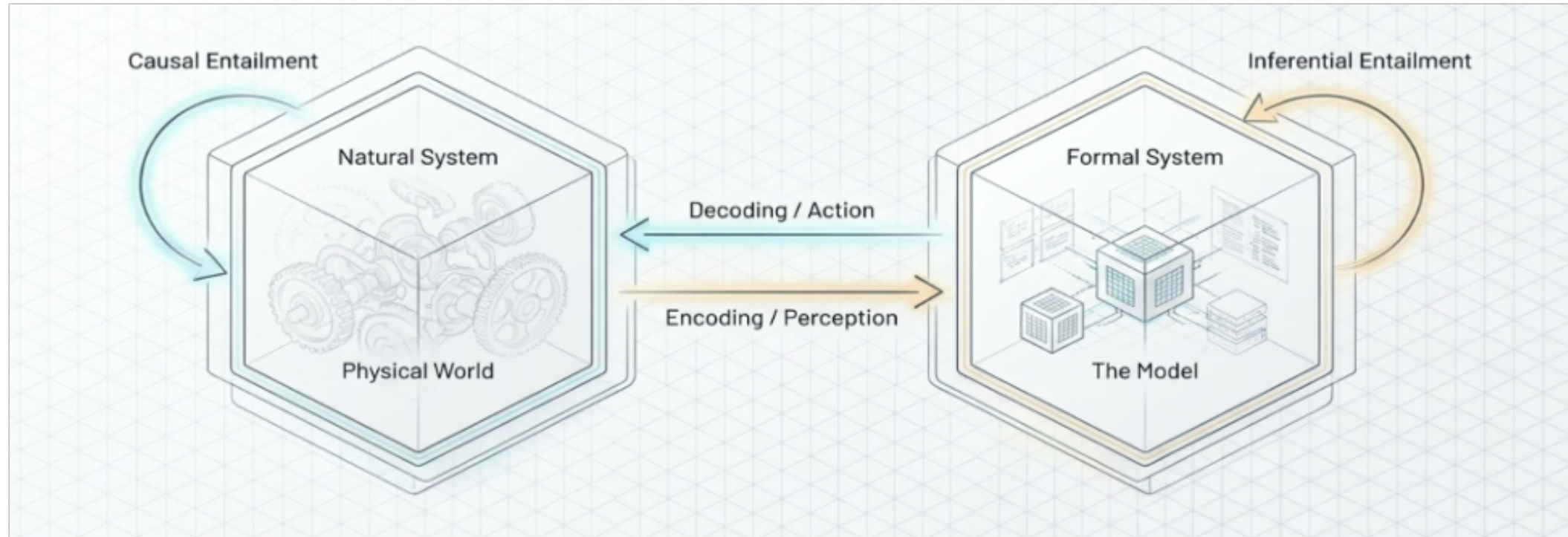
Extraction of actionable semantic value.

Awareness

The ultimate operational state affording the agent to act.



The core pattern: The modelling relation

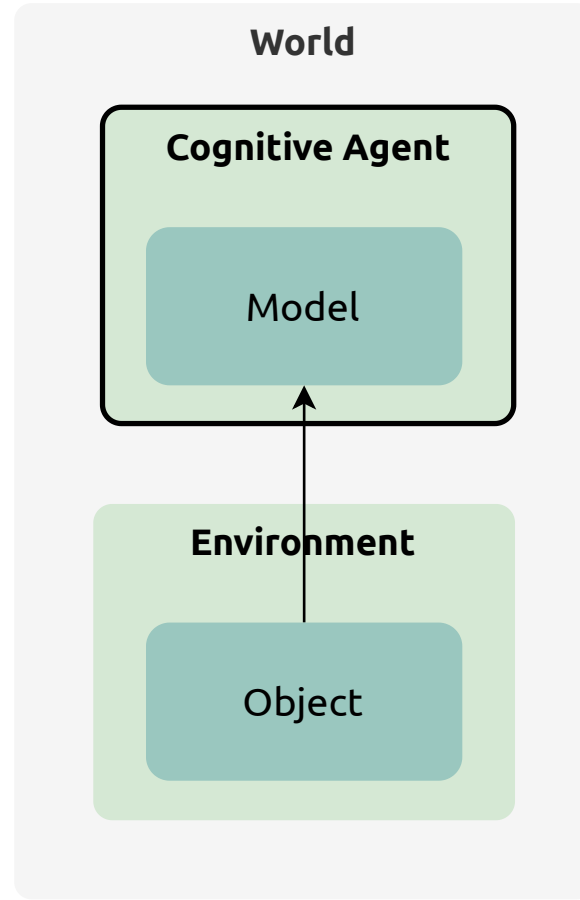


Takeaway. Understanding occurs when inferential entailment in the formal model “**perfectly**” maps to causal entailment in the natural world.

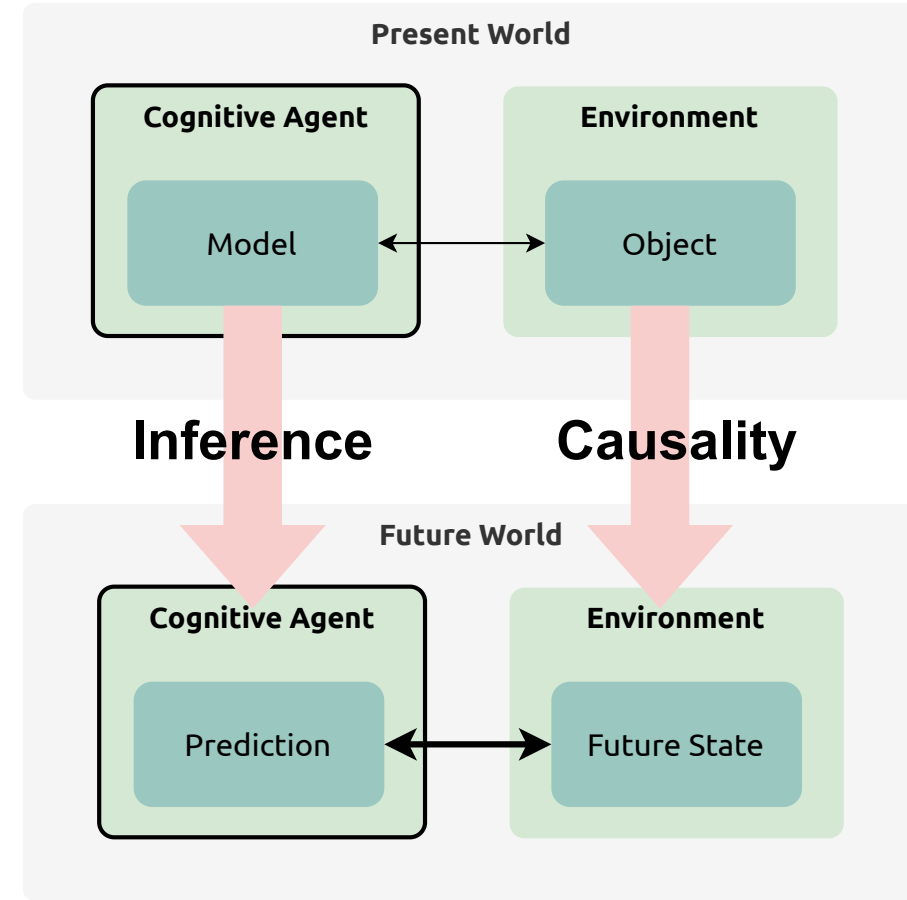
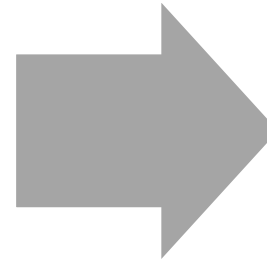


CORESENSE

Modelling the world



CS-IMG-034-V



CS-IMG-035



CORESENSE

Two Epistemological dogmas

THE EPISTEMOLOGICAL ANCHOR

Knowledge is always a model — of something

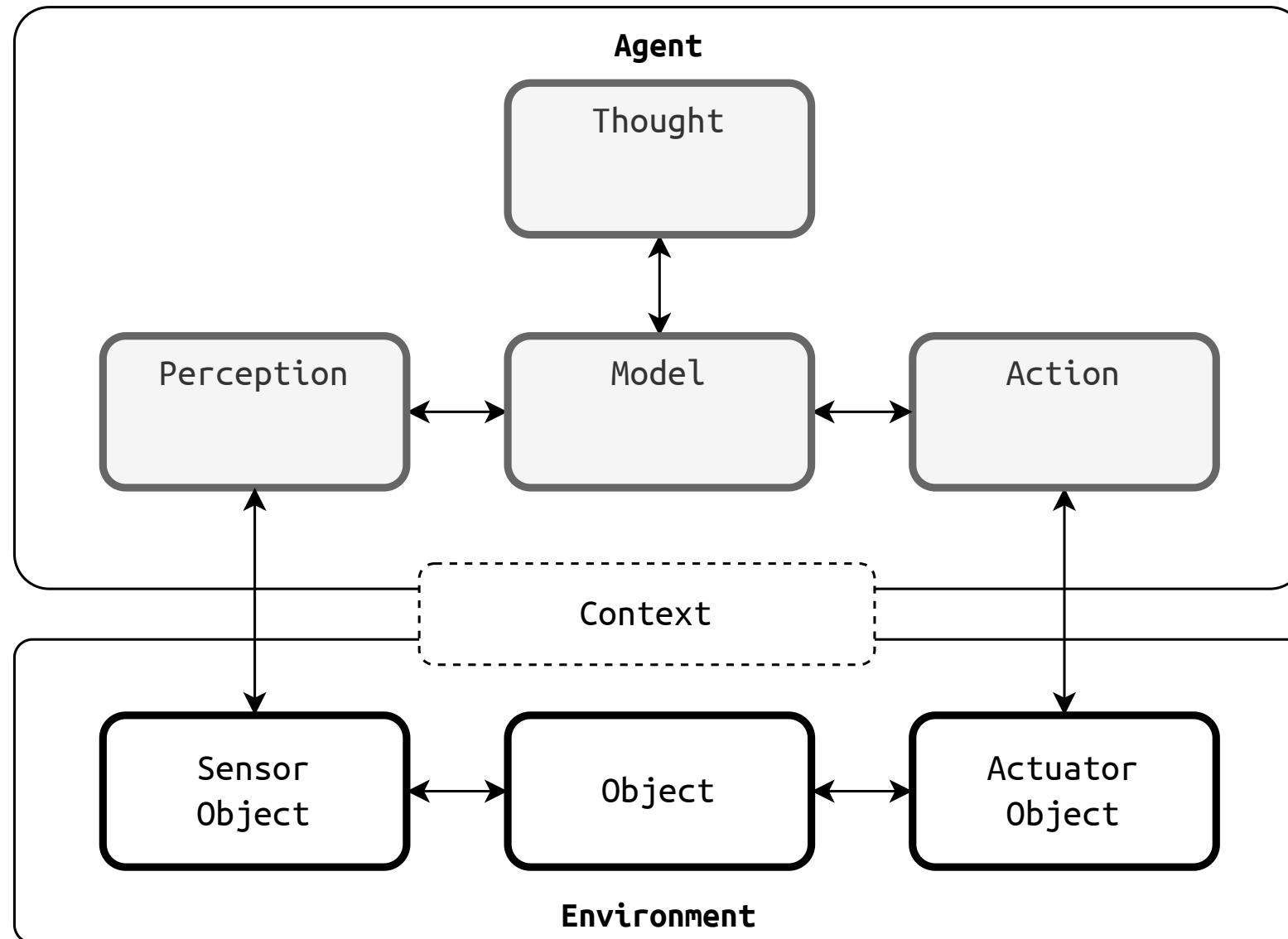
Exertable by an engine, grounded in the modelling relation.

Everything can be modelled — in some formalism

The world is knowable. The modelling relation is bidirectional.

ECL

Epistemic Control Loop



CS-IMG-045



CORESENSE

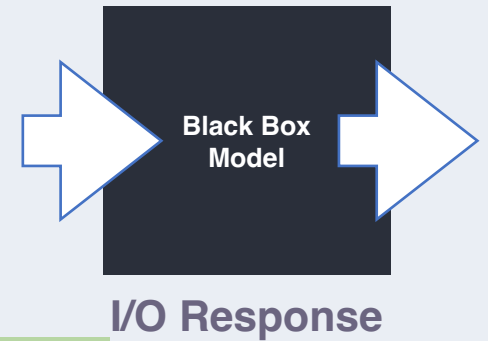


CORESENSE

Learning vs. understanding

Simulation / e.g. DNN model

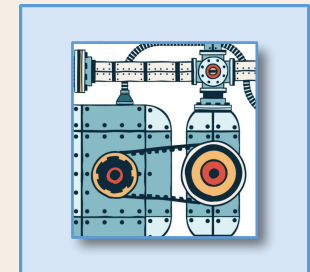
The machine gets responses and understands with the model, but “lacks” structural morphism. Also, cannot tell much about the model. It is merely an opaque statistical simulation.



understanding comes in degrees

CoreSense / Structurally Correlated Model

The machine understands both with the model and the model itself. The structure of the modelled system maps directly to the model, allowing structural adaptation and deep (e.g. counterfactual inferences).

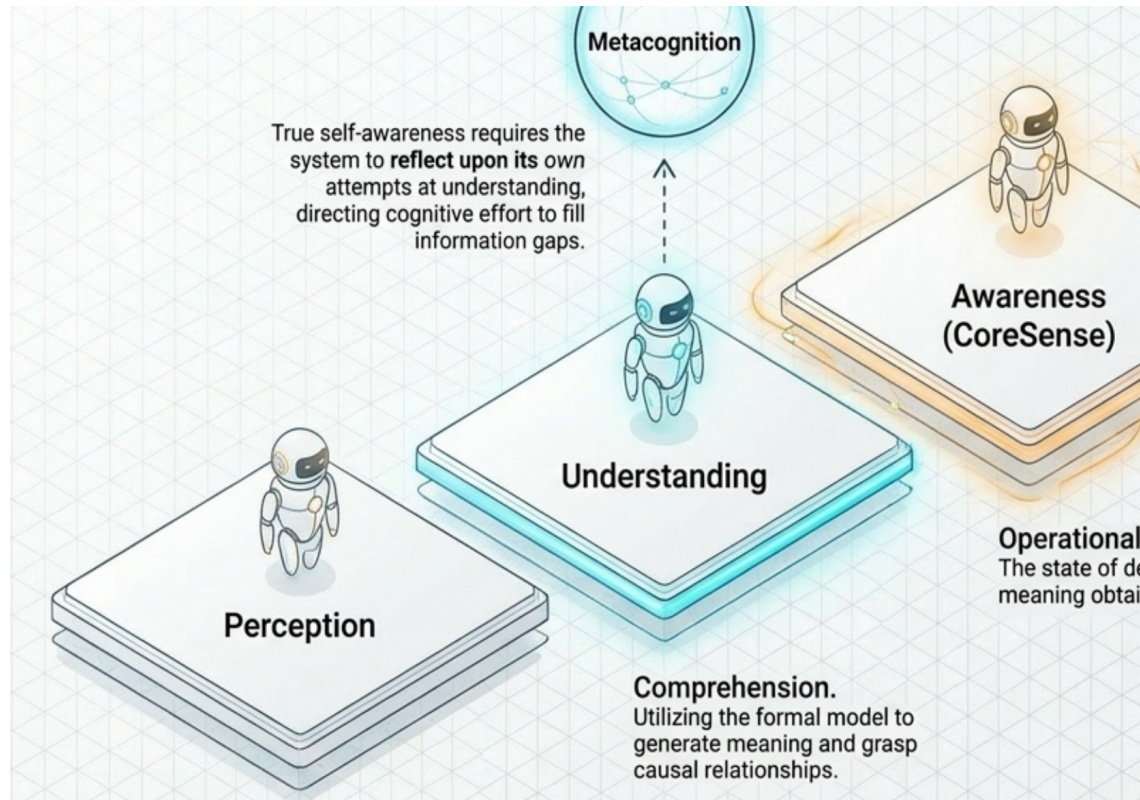


Causal inference

The threshold(s) of awareness



CORESENSE



Perception

A basic state of knowing about something.
Recognition of an external stimulus or data point.

Understanding

Comprehension. Utilising the formal model to generate meaning and grasp causal relationships.

Awareness

Operational actualisation. The state of computing value from the meaning obtained by understanding.

Metacognition

True self-awareness requires the system to reflect upon itself; e.g. about attempts at understanding, directing cognitive effort to fill information gaps.

Practical impact on CoreSense testbeds



Manufacturing

Mobile robots navigating workshops.

CoreSense application.

Continuous fusion of lidar and camera signals to maintain an up-to-date occupancy model (modelets) to safely avoid erratic forklifts.



Social Robotics

Robots operating in human spaces.

CoreSense application.

Modelling phenomena like “patterns of human movement” to support safe, non-intrusive navigation and accurate path planning.



Inspection

Object property identification.

CoreSense application.

Applying the collective awareness pattern to reliably operate a drone team in an open environment inspection mission.



The CoreSense stance: a unified systems perspective

-  **Continuous Perception.** Live sensing feeds the cognitive loop.
-  **Deep Modelling.** Correlated formal system does inferential work.
-  **Value-Driven Meaning.** Meaning is extracted towards goals.
-  **Actionable Awareness.** Executes towards physical environment change.

CoreSense **machine awareness** is not mystical sentience.

It is the inevitable emergent property of a perfectly closed loop between structural modelling, continuous perception, and value-driven meaning.

Seeking a rigorously engineered blueprint for trustworthy machine understanding.



CORESENSE



PART 2 · THE CORESENSE APPROACH

Engineering the Glass Box

Formal definitions of understanding and awareness,
and the cognitive architecture patterns and assets.

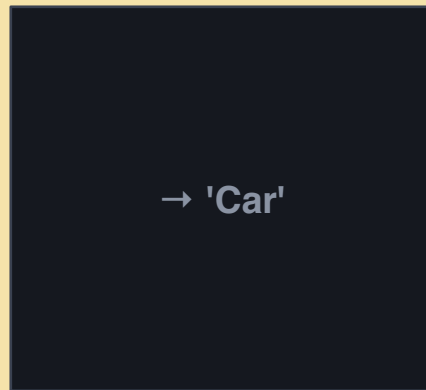


CORESENSE

From the black box to the glass box

THE BLACK BOX

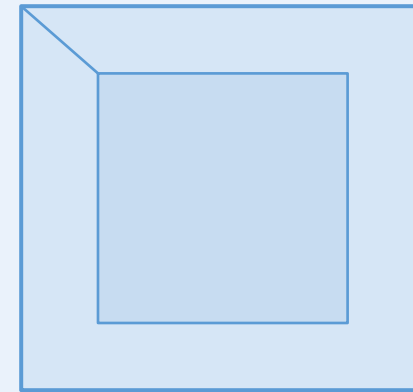
Shallow Understanding



Current systems generate representations but cannot consider their meaning. When training/environmental assumptions are violated, performance drops drastically.

THE GLASS BOX

Deep Understanding



CoreSense systems use deep modelling relations to establish meaning, common sense, and causality to gracefully handle perceptual variety and unexpected situations.

A paradigm shift (back to the sixties)

Dimension	DNN Robotics	CoreSense Robotics
Representation	Implicit, opaque weights in neural nets.	Hybrid. Explicit, actionable models. Also implicit ones-
Adaptability	Requires design-time retraining; struggles with edge cases.	Metacognitive adaptation; synthesises new models in real time.
Trust Mechanism	Empirical — trusted because it worked 10,000 times before.	Intrinsic / systemic — trusted because its beliefs structurally match physical reality.

Defining **understanding**

$$\boxed{U} = \boxed{S} + \boxed{\Phi} + \boxed{M\Phi} + \boxed{Inf}$$



Subject

The autonomous agent.



Target Phenomenon

Any distinguishable aspect of a system in context.



Actionable Models

The possession of integrated model fragments mapping the real world.



Sound Inferences

The exertion of these models by engines to produce valid conclusions.

Key insight. *Understanding is not an abstract feeling — it is a functional state:
the possession of actionable sound models.*

Defining awareness

Awareness is the process of real-time understanding of a situation and its implications.

Ontic Awareness

Focus: States & Processes

Real-time understanding of what is physically happening in the ongoing situation — e.g. a human is crossing the factory floor.

Axic Awareness

Focus: Value & Implications

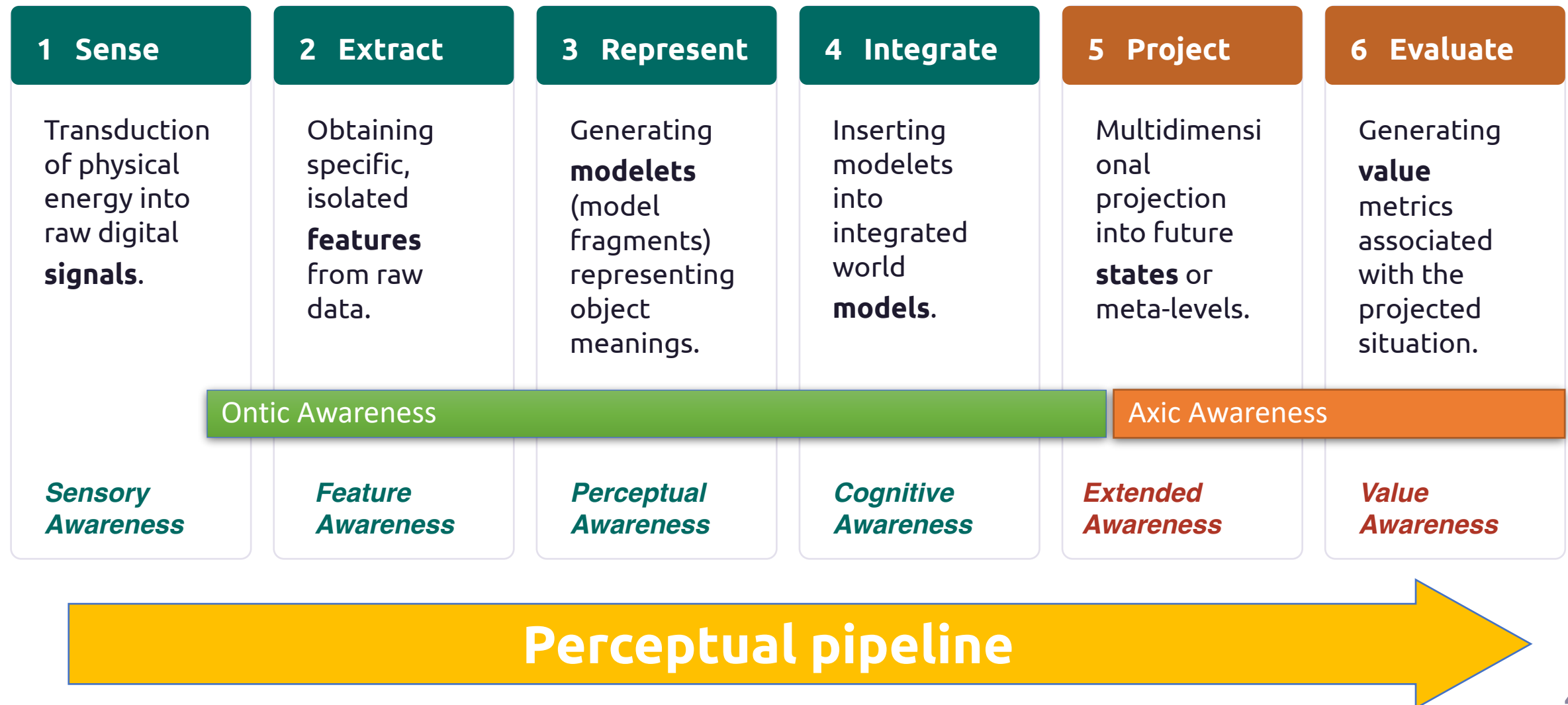
Real-time understanding of the implications of the situation for the subject's goals — e.g. the human is on a collision course; this violates the safety goal.

Epistemic Awareness

Focus: Knowledge & Inference

Real-time understanding of the knowledge that the agent has to deal with this situation — e.g. the robot knows that it does not have a real-time trajectory replanning capability.

The CoreSense awareness process



The CoreSense architecture core elements



Runtime Subsystem

Manages cognitive modules and structures for basic sensor drivers or actuators, and complex cognitive processes.



Understanding Subsystem

Houses the knowledge bases, model repositories, and the inference engines that compute meaning on demand.



Awareness Subsystem

Orchestrates the perceptual pipeline, actively maintaining a federated world model and directing top-down attention.

Three cooperating subsystems to turn raw signals into trustworthy, value-driven action.

The anatomy of machine understanding



Understanding is not a human trait — it is a measurable system capability.

Definition. Understanding is the capability of an entity to construct and use models of a phenomenon or system in order to make reliable inferences about it.



The Object. A **phenomenon** — e.g. human movement patterns, battery decay.



The Asset. **Modelets**: discrete, derived pieces of information encoding characteristics of the system or context.



The Action. **Cognitive functions** execute these modelets to generate actionable, integrated representations — letting the robot predict and adapt.

The Understanding Core



Modelets

Compact composable representations of meaningful information. Instead of monolithic code, the system relies on these agile, atomic model fragments. Can use multiple formalisms: LTL, BT, SM, SysML, iHTN, BPMN, etc. Only some are used in CoreSense implementations.

Engines

The mechanisms that exert or run the modelets to deduce causality, simulate future states, or infer missing data. They are formalism-dependent but cohere at the meta level.

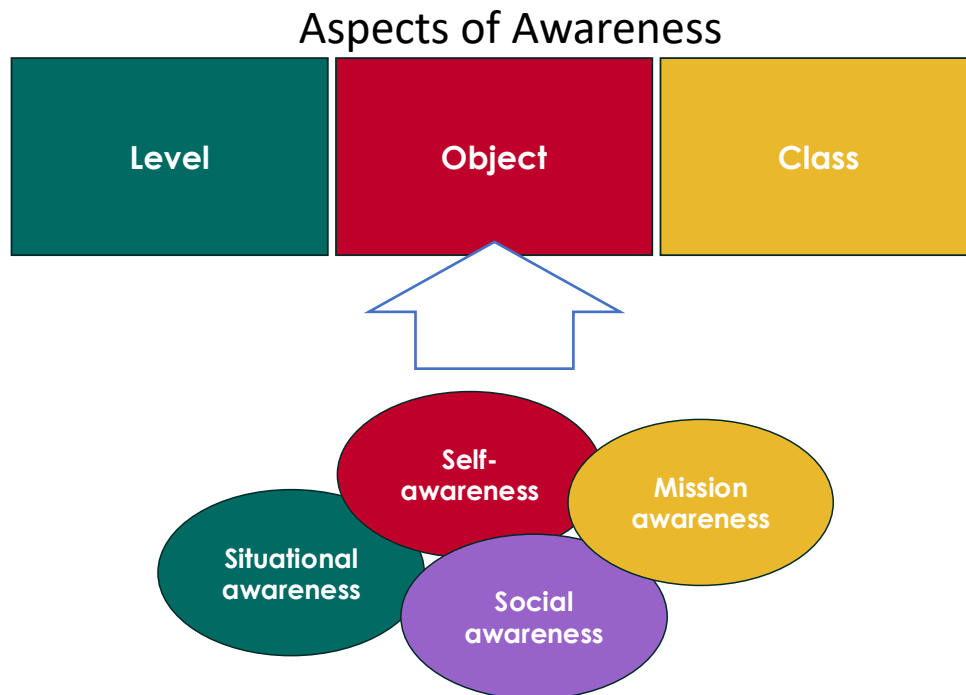
Metacognition

The core plans the act of improving understanding. If a modelet is missing, the

Awareness: understanding in real time



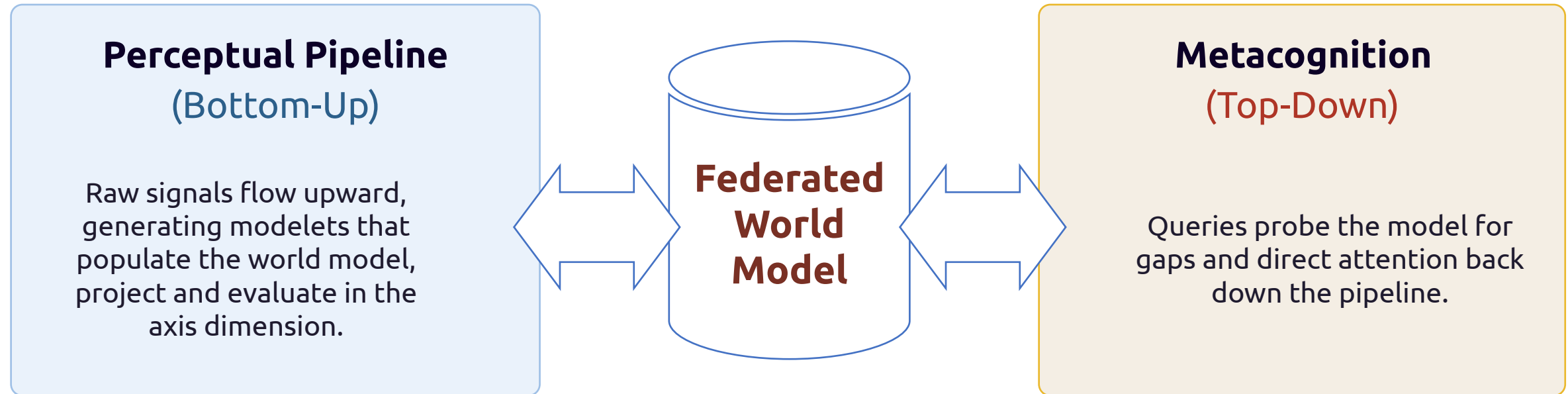
Definition. Awareness is the capability of an entity to maintain timely understanding of a phenomenon and its implications for the entity.



For a system to be aware, it must satisfy three formal conditions:

1. Possess sound models of a phenomenon.
2. Receive and process object signals to keep those models updated.
3. Use the updated models to project into the value dimensions associated to its goals.

The Awareness Subsystem



Key insight. Awareness isn't a passive filter — it is a control mechanism. The Federated World Model is a dynamic, real-time data structure of spatial layout, object semantics, mission objectives, and the robot itself. When a gap in understanding is signaled, the awareness subsystem actively modulates perceptual pipeline activity to fill it (may require also action pipeline activation).



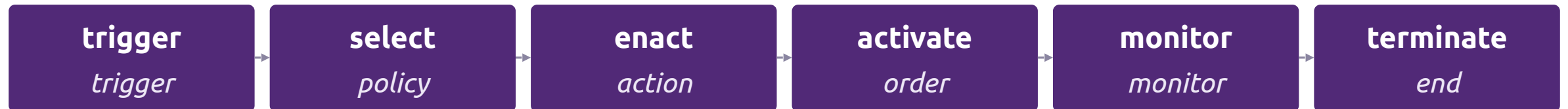
CORESENSE

The Essential Perception and Action Pipelines

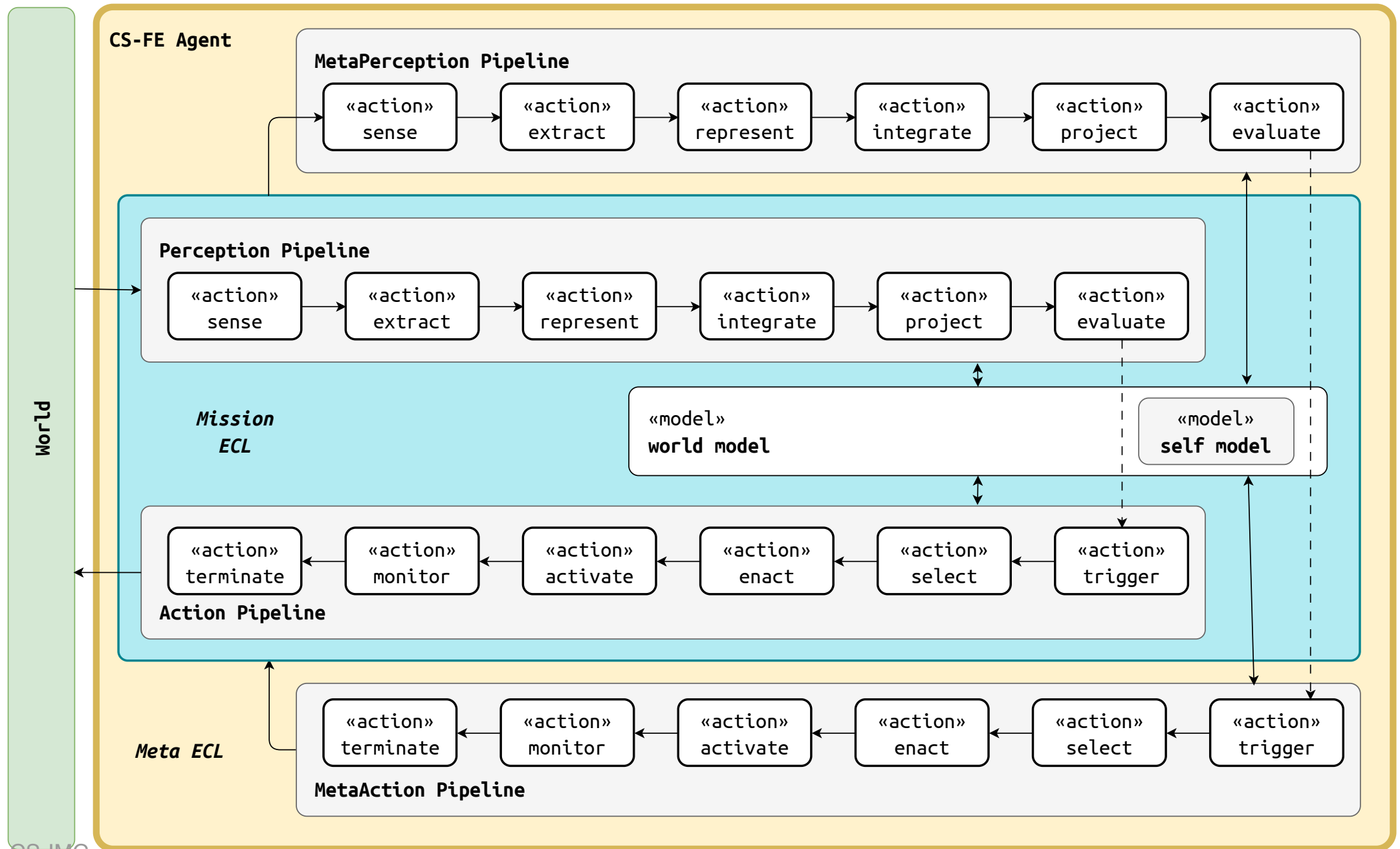
Perceptual pipeline



Action pipeline



Fundamental Essential



Cognitive capabilities in action



CORESENSE

	The System Risk	Required Understanding	Awareness Action
Manufacturing	Dynamic obstacle collisions; workflow bottlenecks.	Predicting aisle congestion using traffic-history and layout modelets.	Proactively pausing non-critical tasks to yield to high-priority machinery.
Inspection	Sudden UAV battery decay over restricted zones.	Modelling state-of-charge evolution against flight profiles and wind data.	Instant swarm replanning; reallocating panel targets and commanding emergency safe landings.
Social	Misinterpreting vague human instructions (e.g. “carry my luggage”).	Relating speech/gesture inputs to internal ontologies of objects and destinations.	Suspending action to request clarification if user-intent models show low confidence.

CoreSense core building blocks



Cognitive Module — A subsystem realising the model–engine concept: it uses models as representations and exploits them via engines to derive relevant information.



Cognitive Structure — An aggregate of cognitive modules that realises a function or capability: the "realisation view," showing how capability is structurally implemented across the CoreSense system.



Cognitive Process — A distributed process among the cognitive modules of a cognitive structure: cooperating processes executing concurrently across modules to realise the structure's aggregated capability as behaviour.

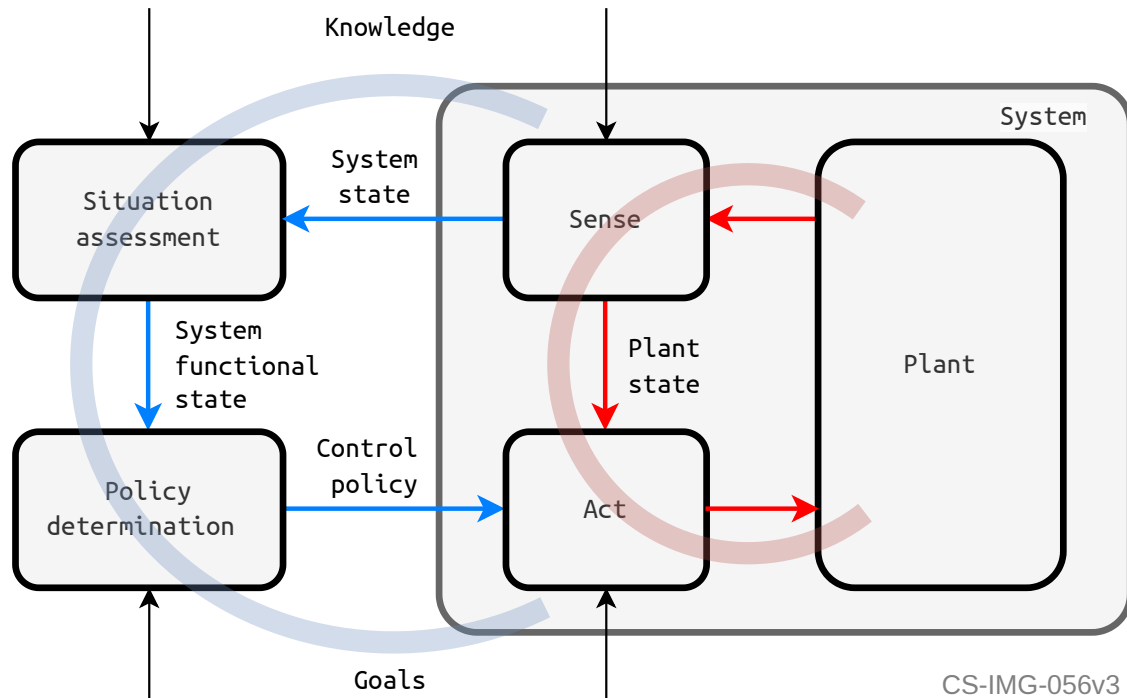
Modules are the *functional atoms* — they carry functions, models, and engines. They connect to one another to *form* structures. **Structures** are the *realisation* layer — capability may exist in a module alone (atomic capabilities) but essentially in an aggregate. **Processes** are *distributed and concurrent* — they run *over* structures, not modules directly, as cooperating sequential processes coordinating through module interfaces (and, at runtime, over ROS2 middleware).



CORESENSE

The Epistemic Control Loop

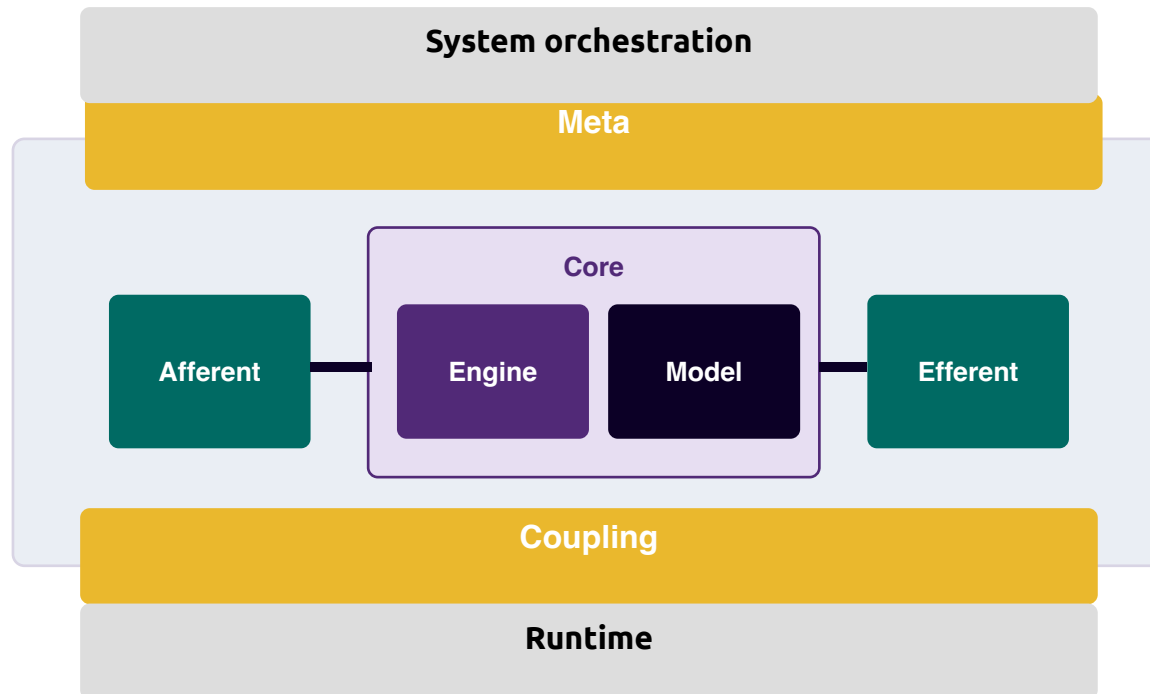
Moving beyond traditional self-adaptive systems, CoreSense implements an advanced epistemic control loop.



- **Situation Assessment.** The Awareness Subsystem continuously evaluates the System state against its Knowledge base.
- **Functional Translation.** It generates a System functional state — a deep understanding of what is actually happening.
- **Policy Determination.** The Understanding Core aligns that functional state with the system's Goals, outputting a dynamic Control policy.
- **Execution.** The policy adapts the Act and Sense functions in real time, completing the loop.

Anatomy of a cognitive module

The architecture is highly modular and composable. Every cognitive module follows a strict, formal pattern to ensure seamless integration.



- **Afferent layer.** Ingests incoming features and signals from the runtime environment.
- **The Core (Engine + Model).** The inference engine executes specific models against incoming data — e.g. a fault-diagnosis engine running an image-analysis modelet.
- **Efferent layer.** Outputs the newly derived understanding back into the system to drive policy or action.
- **Meta & Coupling.** Interfaces that let the module be integrated in a structure (e.g. self-report its status) and connect to the broader architecture.

Some Project-built Cognitive Modules

- **Higher-Order Reasoning** — Vampire-based logic reasoner supporting concurrent reasoning sessions and consultation of external predicates.
- **Causal Reasoning** — Maintains and exploits a causal model for inference; drives policy selection (“what if?” queries) and policy learning (interventions).
- **Self-Adaptation** — Model-based architectural adaptation that infers system state solely from ROS topics, goals and service requests, encoded semantically.
- **Decision Making** — Assembles decision heuristics from the knowledge base with behaviour-tree control flow, producing explainable choices.
- **Risk Awareness** — Image-based human-risk estimation: depth, physically calibrated proximity, 2D head-pose gaze, path-aware and approach-direction risk, multi-sensor.
- **Physics-Awareness** — Symbolic-regression module that learns analytic physical models from data, e.g. the wind field for the a done team.
- **SysML v2 Reasoner** — Vampire + Z3 gatekeeper that grounds PDDL plans against physics/governance rules.

Some Project-built Cognitive Structures

- **Multimodal Awareness Structure** — Composes vision, sound and a contextualiser / risk-assessment component that aggregates perception into FMEA-defined risk properties using current context (location, time), yielding a risk-priority number and level.
- **Explainability (Explanation) Structure** — An explainer selector routes a user question to heterogeneous component explainers (LLM-based, templated, causal model, LIME) and aggregates one answer for the whole application.
- **Anticipation** — Temporal awareness beyond the present: Isaac Sim plus a Forward Simulation Engine validate horizon trajectories (reachability, friction, kinematic safety).
- **EasyNav** — Plugin-oriented cognitive navigation framework with semantic and awareness map representations.

CS Ontology

CoreSense Ontology structured into **three layers**:

CSBase

Systems engineering concepts

- Entity, System, Environment, ExtendedSystem
- Agent, Actuator, Sensor, Stakeholder
- Goal, Mission, Context
- ...

CSCore

CoreSense-specific concepts

- Model, Modelet, Engine, Exert
- Understanding, Awareness
- Phenomenon, Feature
- Meaning, Projection, Evaluation
- ...

CSActionControl

Action-related concepts

- Trigger
- Policy
- Action Specification
- Action Order
- Action End



CORESENSE

Eight principles that shape the architecture



Cognitive

Based on knowledge
— not imitating
human thought



Model-centric

Models are the core
asset, build & run time



Value-oriented

Deliver value to a
principal, not mere
change



Understanding

Sound inferences from
the agent's models



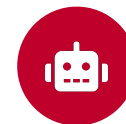
Self-aware

Awareness turned on
the agent itself



Hybrid

Heterogeneous
modelling frameworks



Autonomous

Decides from its own
understanding



Trustworthy

Dependability from
model + value
grounding



From cognitive system theory to reality

Theory. Formal definitions of understanding and awareness processes.

Design patterns. Reusable design patterns at different abstraction levels.

Discipline. —> Map all these results into SE (e.g. using SysML v2 models).

Because CoreSense defines cognition functionally, executes it via formal models, and governs it with international engineering standards, the result is not just statistically successful — it is **intrinsically trustworthy.**

**Cognitive autonomy is no longer a black box.
It is a rigorously engineered blueprint.**



CORESENSE

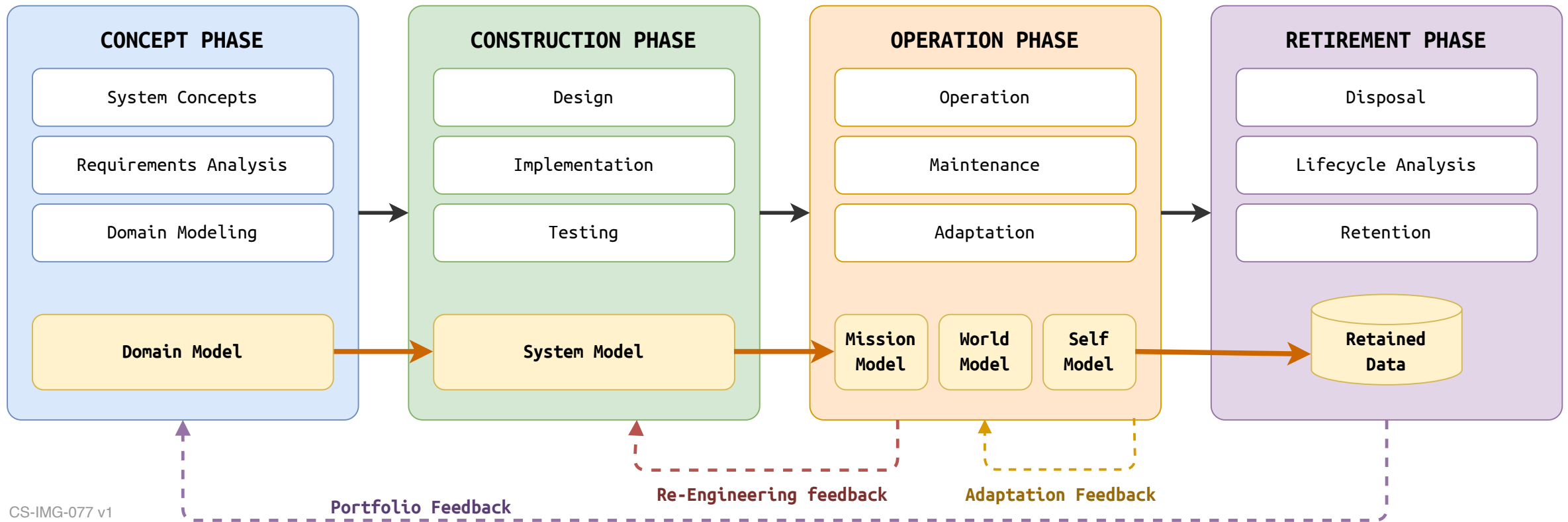


PART III · ENGINEERING AWARE SYSTEMS

Model-Based Autonomous Systems Engineering (MBASE)

From MBSE to MBSA — lifecycle processes to apply the reference architecture and the cognitive modules.

Building systems



In CORESENSE we use a **systems engineering (SE)** approach to the construction of cognitive systems

CoreSense targets autonomous systems engineering, not just individual systems — applicable to both greenfield and brownfield (existing ROS 2).

Bridging cognition and systems engineering

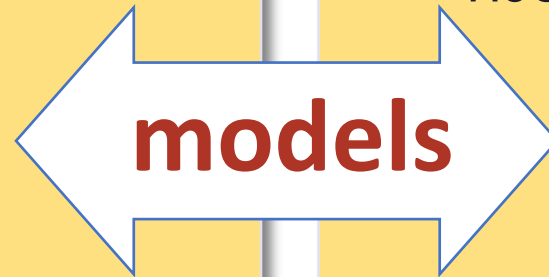


Cognitive Science

Mental models
Theories of Mind
Real-time Awareness
Metacognition
Meaning Computation
• • •

Systems Engineering

Model-based systems engineering
System lifecycles
Traceability
Verification
Formal logic
• • •



The MBASE Thesis

Engineering-grade autonomy requires formalised cognition. Internal mental models must be built using the exact same rigorous, standard-compliant engineering logic used to build the physical hardware. Ad-hoc software patches are a no-go.

Beware !

We use the term “model**” in a specific, general sense.**

From MBSE to MBSA

The fundamental leap in CoreSense: treating models not as passive design documentation, but as actionable data objects that fuel metacognition.

MBSE

Model-Based Systems Engineering

Phase

Engineering time.

Purpose

Design, specify, and verify complex systems.

Asset type

Static blueprints, requirements, system structures.

MBSA

Model-Based System Autonomy

Phase

Operation time.

Purpose

Enable self-awareness, diagnosis, and dynamic adaptation.

Asset type

Live, executable runtime models driving the Epistemic Control Loop.

The standards constellation

ISO/IEC/IEEE 15288 - SE

System Life-Cycle Processes — framework for stakeholder needs, validation, and continuous assurance.

ISO 42020 - Architecture

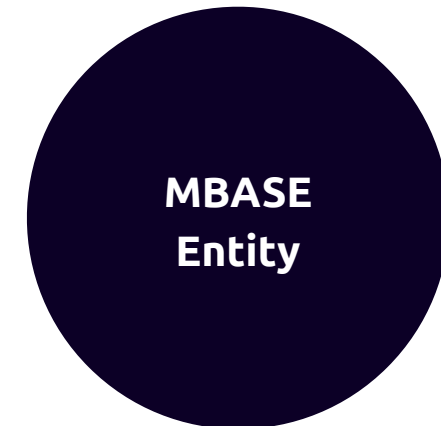
Architecture Processes — governs how the complex multi-view architecture of the cognitive agent is formally described.

ISO/IEC/IEEE 24641 • MBSSE

The formalised application of modelling to support systems engineering, enforcing digital single-sources of truth.

Digital Single Source of Truth

One coherent **SysML v2** model binds requirements, architecture, and conceptualisation.



The language of CoreSense cognition: SysML v2



```
package CoreSense {  
    import OMG_SysMLv2::*;  
  
    part def CognitiveAgent {  
        part perception : PerceptionSystem;  
        part reasoning   : ReasoningEngine;  
        part action      : ActionController;  
  
        connect perception to reasoning;  
        connect reasoning  to action;  
    }  
}
```

The Tool
OMG SysML v2.



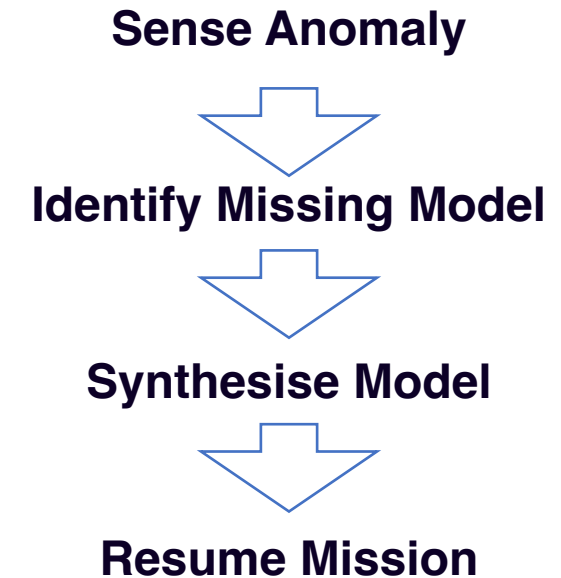
The Function
Executable models. The CoreSense abstract architecture and ontology are captured entirely in executable model form.

The Benefit
A single source of truth. Engineers create logically consistent, multi-view architecture descriptions that are verifiable, traceable, and simulable before a single line of control software is deployed.

Models at engineering-time vs. operation-time

Dimension	SE	CoreSense
Primary Actor	The Human Engineer	The Autonomous Robot
Tooling	SysML v2 environments	The Understanding Core & Modelets
Goal	System Delivery & Verification	Mission Resilience & Anomaly Resolution

The Meta-Epistemic Control Loop



The robot constantly acts as its own systems engineer in the field.

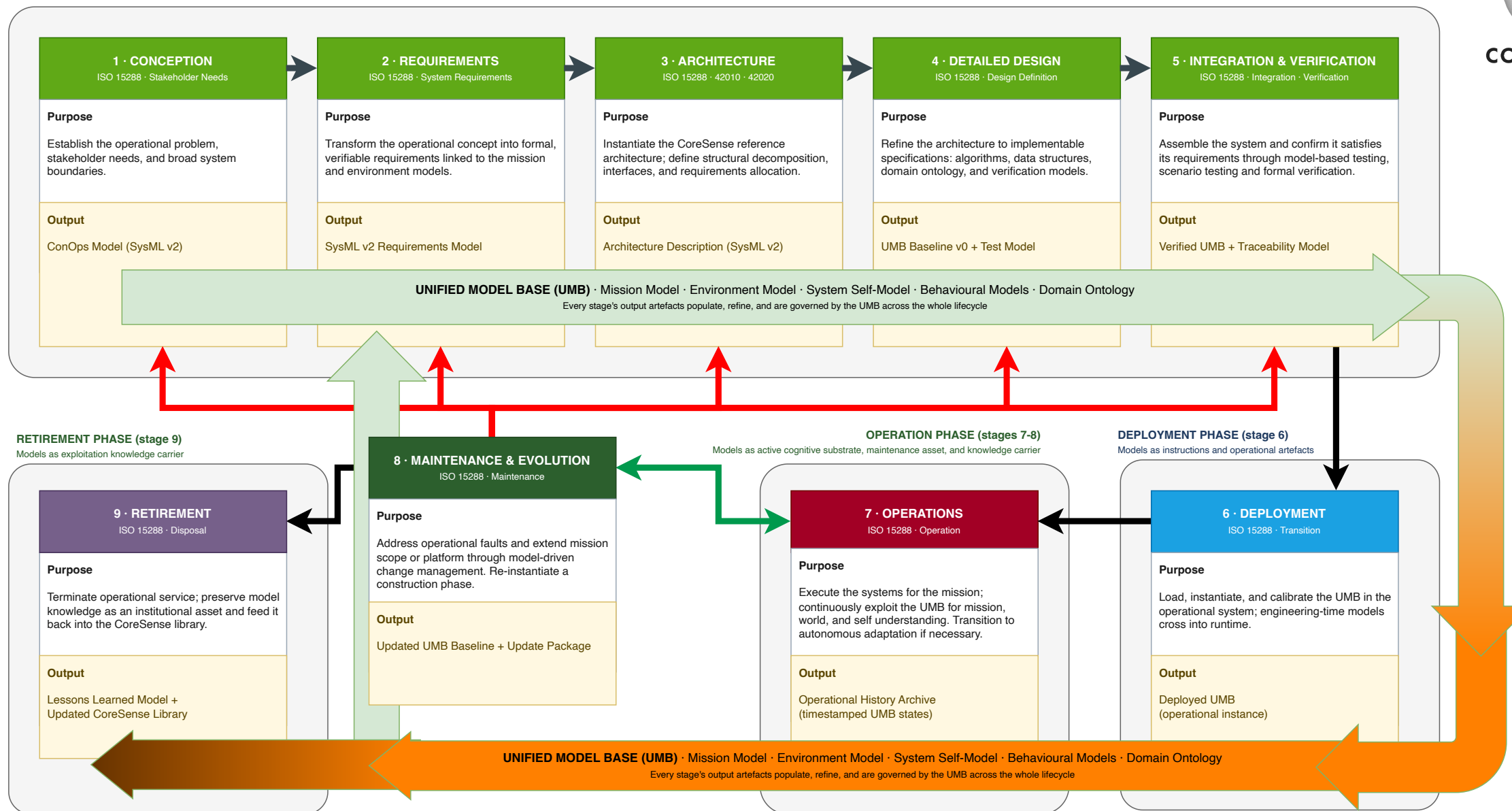
MBASE — Model-Based Autonomous Systems Engineering Lifecycle

CONSTRUCTION PHASE (stages 1–5)

Models as communication, specification, implementation and verification artefacts



CORESENSE



From blueprints to brains

The future of autonomous robotics cannot be hard-coded. Dependability in the real world requires systems that truly know themselves and their context.

- ✓ **Standardised.** Built on ISO/IEC/IEEE 15288, 42020 and SysML v2.
- ✓ **Formal.** Driven by executable modelets and valuation engines — not black-box guesses of LLMs.
- ✓ **Aware.** Constantly bridging the gap between design-time models and run-time reality.

Takeaway. By transitioning from Model-Based Systems Engineering to Model-Based System Autonomy, CoreSense provides the formal blueprint for the next generation of trustworthy, cognitively aware robots.



CORESENSE



PART IV · MISSION ARCHITECTURE IN PRACTICE

Developing a CoreSense Mission Architecture

A standards-grounded lifecycle and method for
architecting cognitive systems.

The 9-Stage Lifecycle at a Glance

1. Conception — define scope, stakeholders, ConOps
2. Requirements Definition — formalize requirements
3. Architecture Definition — instantiate the CoreSense reference architecture
4. Detailed Design — build modules, structures, modelets, and ontologies
5. Integration & Verification — assemble on ROS 2 and formally verify
6. Deployment & Operational Transition — calibrate and load the mission
7. Operations — run concurrent cognitive loops in the field
8. Maintenance & Evolution — impact analysis and asset upgrades
9. Retirement — harvest lessons learned into the reusable library

MBASE Lifecycle → CoreSense Architecture

Step 1: Conception

Goal:

Define system boundaries, identify stakeholders, establish operational problems, and outline the Concept of Operations (ConOps).

Procedure:

- Scenario Modeling: formulate operational scenarios using SE templates; focus explicitly on anomalous and failure scenarios requiring self-awareness, understanding, and adaptation — not just nominal behavior.
- Bounded Modeling: create an Operational Concept Model, Stakeholder Need Models, and a Mission Thread Model (seed) to establish the robot's environment and target goals.

Output:

A formalized ConOps Model representing a shared, unambiguous operational context.

Step 2: Requirements Definition

Goal:

Translate stakeholder needs into formal, verifiable, model-linked system requirements.

Procedure:

- Categorize requirements using the CS taxonomy: Context (RC), System Functional (RF), System Non-functional (RN), Process (RP), and Testbed-specific (RT).
- Specify core cognitive demands — e.g. Model construction capability (ARCH-RF-001), Self-reconfiguration (ARCH-RF-004), etc.
- Build core models: an initial Mission Model and a Reference Environment Model, maintained via a centralized requirements database for traceability.

Output:

A verified Requirements Model linked directly to the ConOps.

Step 3: Architecture Definition (1/2)

Goal:

Instantiate the CoreSense Reference Architecture patterns and establish the robot's structural and behavioral layout.

Procedure:

- Apply the Epistemic Control Loop (ECL) as the core design pattern.
- Incorporate the fundamental CoreSense structures: Runtime, Perception, Action and Metacognition (e.g. using the Fundamental Essential or the Navigation Essential).
- Define cognitive processes as mission threads.

This step is the essence of the CoreSense approach.

Step 3: Architecture Definition (2/2)

Select Reusable Cognitive Modules and Structures for the Cognitive processes:

- Navigation — EasyNav (semantic navigation) with Risk Awareness (hazard calculation).
- Goal-Directed Control — Decision Making via behavior trees.
- Anticipation — predictive simulations of future states and horizon trajectories.
- Explainability — Explainer hierarchy for human-robot interaction.

Define Viewpoints and Views:

Model the architecture across Operation, Context, Functional, Baseline, Cognition, Cognitive Module, Cognitive Structures, and Implementation viewpoints, in a SysML v2 model.

Output:

An Architecture Description outlining component layouts, interfaces, and model structures.

Step 4: Detailed Design

Goal:

Refine the architecture into algorithms, data structures, and ontological models.

Procedure:

- Design ontological foundations by mapping structural layers to the CoreSense ontologies in SysML v2.
- Formulate Modelet Specifications for all exchanged data: ID, Timestamp, Precision, Uncertainty, or raw data.
- Instantiate cognitive modules using the CoreSense module pattern: Afferent (inputs/subscriptions), Core (processing logic, model, engine), Efferent (publishers), and Meta/Coupling (lifecycle, structures services).

Output:

A **Unified Model Base** (UMB) Baseline v0 and a derived Test Model containing the software specifications and formal ontologies.

Step 5: Integration & Verification

Goal:

Assemble the system on the target platform and verify correctness via model-based testing and formal methods.

Procedure:

- Build the ROS 2 workspace, mapping the SysML v2 design to ROS 2 lifecycle nodes using the CoreSense runtime framework.
- Verify modelet flow by testing the chaining of cognitive modules in both push (subscription-driven) and pull (on-demand) modes.
- Formally verify properties using the SysML v2 Reasoner to check safety, validate PDDL plans, and confirm governance-rule adherence.

Output:

A Verified UMB alongside a Traceability Model confirming coverage of all functional and non-functional requirements.

A source implementation according to the verifiedUMB.

Step 6: Deployment & Operational Transition

Goal:

Have a deployed system integrating the robot's physical sensors and system models for its specific mission.

Procedure:

- Load the engineering-time models (UMB) directly into the robot.
- Load the mission instance and populate site-specific models, e.g. concrete 2D occupancy grids or topological navigation maps of the deployment site.
- Calibrate sensors to models: align LIDAR, camera, and IMU readings with semantic Afferent inputs.

Output:

A Deployed UMB (operational instance) ready for live execution.



CORESENSE

Step 7: Operations (1/2)

Goal:

Execute the robot's mission while continuously leveraging the model substrate for understanding, anticipation, and self-adaptation.

Procedure:

- Maintain live models: keep runtime models updated through continuous sensor input processing (e.g. wind estimation, structural deformation, obstacles).
- Execute concurrent cognitive loops operating at different timescales — detailed on the next slide.

Step 7: Operations (2/2)

Concurrent Cognitive Loops:

- Reactive (ms): basic reflex loops mapping percepts directly to actions.
- Deliberative (s–min): plan and execute actions (e.g. via the behavior-tree logic).
- Adaptive (s–min): use Causal Reasoning and the Self-Adaptation module to adjust action parameters for minor environment variations.
- Reflective (min–h): activate the Metacontrol/SAS layer; on anomaly detection (e.g. sensor failure), dynamically transition ROS 2 node lifecycles to isolate the fault and activate a backup pipeline.

Archive Operational State:

Continuously stream localized state information and UMB states to the Operational History Archive.

Steps 8 & 9: Maintenance, Evolution & Retirement

Goal:

Address the extended lifecycle needs of the system.

Procedure:

- Perform impact analysis: when upgrading physical parts or software capabilities, use the UMB's Impact Analysis Model to predict which parts of the cognitive architecture will be affected.
- Harvest lessons learned: upon decommissioning or at key milestones, analyze the Operational History Archive to uncover design-assumption gaps.
- Package newly refined models or custom modules and feed them back to upgrade the core CoreSense reusable library.



From framework, to system

Across foundations, theory, engineering, and practice, a single thesis holds: knowledge is always a model — exorable by an engine, grounded in the modelling relation.

- ✓ **Cognitive Blueprint.** The epistemology of machine understanding — models, not data, are the basis of cognition.
- ✓ **Engineering in Transparency.** Formal definitions of understanding and awareness, and the CoreSense cognitive architecture.
- ✓ **Model-Based Autonomous Systems Engineering.** From MBSE to MBSA — the reference architecture, cognitive modules, and the epistemic control loop.
- ✓ **The CoreSense Mission Architecture Synthesis.** A standards-grounded method for architecting cognitive systems, with the CoreSense platform and assets.

From a stated mission to a trustworthy cognitive architecture — one model, end to end.

Summer School for Systems & Software Engineering in Robotics (SSSER'26)
Delft, June-July 2026



CORESENSE

Developing Cognitive Robots with CoreSense

The CoreSense Process — from epistemic foundations to mission architecture

Ricardo Sanz and Ruth Medeiros

ASLab · UNIVERSIDAD POLITÉCNICA DE MADRID

coresense.eu



The CoreSense project has received funding from the European Union's Horizon
Europe research and innovation programme under grant agreement No. 10107054