May 12, 2026

# Ainolabs Cognitive Fitness: Automotive Sidebar

## Sidebar: Automotive example

Enterprise workflow, automation and possibility to apply ML techniques is very open-ended. There are too many possible activities and workflows to choose from for an example. A full enterprise model – "automated digital twin" would be excessively complex, too. Looking for a reasonably complex but still bounded and constrained setting we'll consider an autonomous vehicle.

There is a clear OODA (Observe – Orient – Decide – Act) loop to use as the background. A vehicle has multiple sensors, which provide observations to the first part of the loop. Question is – what happens with Orient and Decide? Action based those is more or less mechanical (actuators).

There are two layers:

1. **Functional decomposition of the A→B mission workflow**
2. **How subsystems interact with a central world model without corrupting it or each other**

We will address workflows first and the dive into subsystems, potential application of ML techniques in them and the characteristics of a World Model.

### Main Workflow

A summary of an autonomous vehicle's main workflow is simply "Move from place A to place B in a certain time." That needs to be augmented with safety concerns such as "avoid damage to the vehicle and to other parties", possibly with economics starting with "pay attention to fuel status and economy", and maybe including a payload by "carry the cargo and passengers safe and sound to the destination". The latter would be important to marine and airborne vessels, which could tolerate much harsher conditions than human passengers and most payloads.

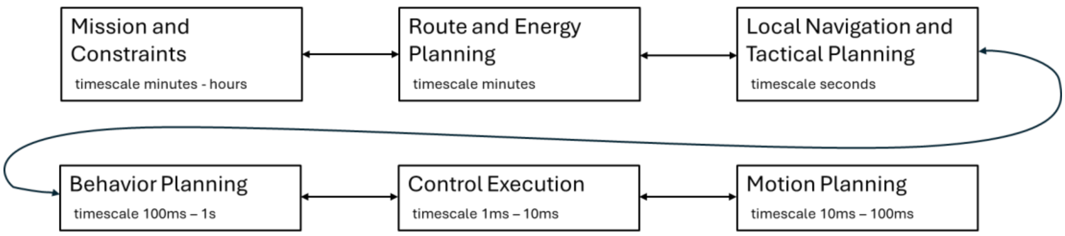
To keep the story and discussion constrained and somewhat clear the explanation below is about a land-based vehicle on common roads, or "a self-driving car".

This can ´be expressed as three separate subproblems, or problem classes:

1. **Mission planning** (what should happen)
2. **Behaviour and motion generation** (how to make it happen)
3. **Control execution** (making physics comply)

While it would be possible to build systems where these categories are tightly coupled, they are kept separate for clarity and for discussing concepts and design patterns that are used for other problem domains later.

The illustration below outlines the nested OODA loops on different levels. The levels, their actions and interplay are explained in later paragraphs.




Level *Mission and Constraints* addresses user intent and constraints. Inputs include aspects like origin and destination, expected arrival time, available energy, and possible external and environmental constraints such as weather, route and road closures and regulatory limits, including safe and prudent speed.

As output the layer generates a mission plan, possibly with trade-offs between goals and constraints. Mission plan is unlikely to change much or often once established.

Mission and Constraints layer should not control the vehicle or execution directly.

Level *Route and Energy planning* uses the output by Mission and Constraints with maps, traffic information, energy consumption and other similar and relevant data to generate a detailed route, analyse energy feasibility, and possible refuelling or re-charging plan, and an estimate of the duration of the trip.

Route and Energy planning output would be a concrete ordered list of route segments at appropriate level of detail.

Level *Local Navigation and Tactical Planning* operates closer to action in timeframes spanning seconds. This level would generate short term trajectory and direction goals such as “start moving forward and turn right or left from the driveway”. The route should be translated into feasible manoeuvres that can be performed in the then-current surroundings.

As inputs this layer uses the route generated earlier, a *World Model*, and vehicle state.  World model appears here for the first time with an important function: What is the meaning of environmental sensors and how do they match with e.g. maps and other representations of the surroundings?

Local Navigation and Tactical Planning is essential, core for an autonomous system, vehicle in this hypothetical scenario.

Navigation and Tactical Planning level generates manoeuvres. Level *Behavior Planning*  selects the next manoeuvre for execution. That decision is based in inputs such as tactical intent, world model and vehicle constraints (what is physically possible). This could mean decisions whether to stop, overtake someone else, yield etc and in urgent cases emergency brakes.

These decisions can be called target trajectory, i.e. what should happen within the next second.

Closing on actual physical events, level *Motion Planning* uses the target trajectory, or manoeuvre with known obstacle geometry and vehicle kinetics to establish a physically realistic and possible trajectory – direction – for the immediate 10-100ms. Physical attributes position, velocity, direction, with curvature if necessary are included in the output of motion planning.

Finally, the level *Control Execution* accepts the inputs by Motion planning and generates actions with steering angle, throttle, brakes, signals etc. On this level the vehicle moves, there are physical consequences but no Machine Learning, no AI, just actuators.

## The World Model: What It Is — and Is Not

Main workflow listed plenty of planning and actions, but nothing about the environment, no sensors, no common or artificial sense of what the measurements and input from the sensors might mean for the vehicle and the workflow.

As the goal of this example is to characterize the world model for more general purposes, the discussion below is light in actual vehicle related issues. Those needed to be addressed in an overview level as mentioned above to get to the essence of OODA-loops and the purpose and meaning of a World model in an automated and/or autonomous system.

Article "Critiques of World Models", Xing, Deng, Hou and Zu, https://arxiv.org/html/2507.05169v1 discusses world models as they are currently viewed in the AI industry. The prevailing direction is to attempt to train an ML model based on general data, including relevant sensory input. The authors propose a PAN model, which is similar to Ainolabs CMLS architecture, even if the latter is described on conceptual level only (see https://ainolabs.com/aino-tech/)

The prevailing meaning of a World Model is an all-powerful NN machine capable to observing, simulating and predicting real-world events. As a comparison Xing, Deng, Hou and Zu refer to science fiction such as the Dune and the divine forecasting abilities depicted in that and other fine pieces of art.

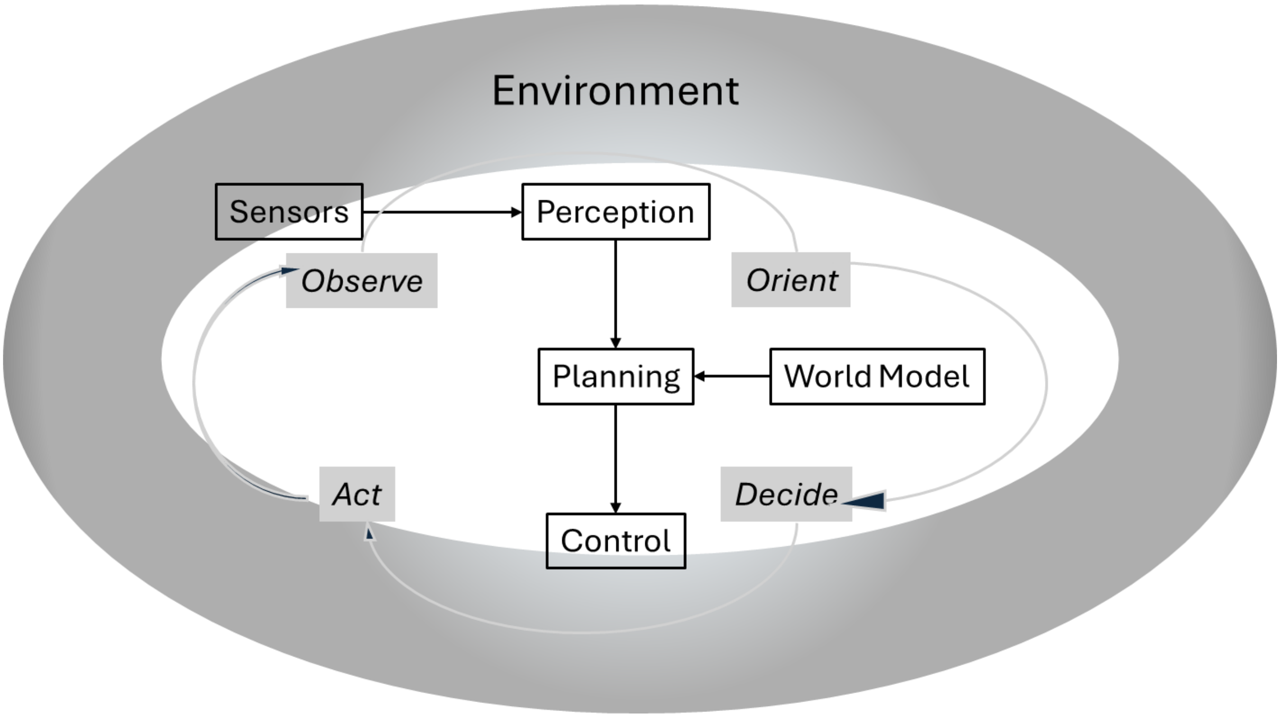
In this context, World Model has a substantially humbler meaning. A World Model is a part of a system intended to isolate uncertainty in a manageable manner. In other words, it is an engineering artifact, a component that can be build for certain purposes, such as automotive, and serves a purpose as part of a system. A World Model has defined interfaces so that it can be queried, and when necessary, updated with new relevant information.

A World Model is not an all-powerful sage, a neural network that can predict future scenarios to distant future. Instead, it is a probabilistic and time-indexed state estimate, i.e. a collection of possible events in the future. The life-span of events may vary a lot: Some are ephemeral, some may last for very long, or "forever".

The World Model establishes an abstracted view or reality, and it can and should be used as a contract facilitator between perception and planning.

The illustration below depicts the system modules and their relationships. Sensors read data from the environment. That is provided to Perception, who interprets the sensory data for Planning. Planning queries the World Model, and prepares the necessary action plan for Control for the next cycle of operations.
This cycle is the OODA – Observe, Orient, Decide, Act – loop for the system in its operating environment.




## Data, or facts, perception, World Model and decisions

Sensors, or facts, lead into perception. Perception, especially if there are conflicting data is checked with the World Model. Action planning, what to do next, uses all of the above, but should not modify facts, perceptions, nor the World Model.

If and when Action Planning fails, or can't produce a sufficiently robust sequence of actions for the next appliable time slot, the system must revert to human control. This obviously is not suitable to systems that need to operate faster than human capabilities. For such cases a fail-safe should be set up, e.g. "stop the vehicle immediately" for a self-driving car.

World model resolves conflicts between different perceptions ("a parked car of a plastic sheet"?), maintains belief state, and tracks uncertainty over time. World model is not an oracle; it is a statistical state estimate of reality beyond hard facts (perceptions). This prevents feedback loops within the sensory and perception modules and helps avoiding unchecked hallucinations in the system.

Planning does not modify or correct perceptions or observations Instead, possible contradictions and issues that require attention are escalated from truth-oriented system components to world model and it's capability to address and handle uncertainty and beliefs.

For all this to work, the communication interfaces need to be clear and well defined. At least update rates (speed), confidence representation and failure modes need to be defined for each interface between system modules.

While technical details for any specific example are omitted here for the sake of clarity, the World Model exports and provides belief distributions, not facts to the other parts of the system.

## How is this relevant to Organizational Workflows?

An enterprise separates strategy from operations, measurements (“sensors”) from decisions, execution from authority and protects shared models of essential corporate culture. This resembles the simpler example of an autonomous vehicle.

An enterprise is a more complex entity than a vehicle. The beliefs and parts of corporate culture and operational best practices are varied as the corporation consists of many different functions, organizations and people with presumably aligned, yet slightly different goals and priorities. Consider e.g. a large customer who requests a very big delivery on a tight schedule, and the possible different views to that by sales, sourcing, manufacturing, logistics and customer service. These different views would be expressed as beliefs such as “customer will be very satisfied if we delivery in 2 weeks” (sales), “there won’t be enough materials for other orders” (sourcing), “factory capacity is insufficient on week 3” (manufacturing).

The World Model is a place, tool and component to capture and process different expectations – beliefs – about the future, and that is how and why the discussion of an autonomous vehicle was necessary and useful for the author to outline the concept of a World Model in the context of Enterprise Workflows.

### Practical implications for Enterprise Workflow automation

The system and its parts, workflows, should be split by time scale and belief (epistemic) certainty. Very fast processes with very high certainty are easy to automate with traditional algorithmic solutions – that’s run of the mill industrial automation, and not in the direct scope here.

While decisions in the automated and non-automated workflows should be as local as possible, the state of the enterprise – sum of all workflows – should be as unified or centralized as possible. The opposite, distributed and internally conflicting state would be a poor foundation for running an organization.

Metrics, measurements, sensors and other input devices and systems should be calibrated to produce comparable and compatible information. If this is done properly, it is possible to detect disagreements between different system components. For example, vision (camera) and radar can produce different views under poor visibility and / or loads of clutter (e.g. dust) in the radar view. Similar discrepancies happen in organizations, and they need to be visible so that they can be addressed and handled. It is essential for sensory input to disagree in compatible ways.

World Model can then resolve the ambiguity, and still preserve the actual observed facts, the ambiguous results from by the instruments. This implies almost certainly a slower, smarter process than straightforward automation – ultimately “Stop the production line” -button in a plant.

### Facts or beliefs?

Between sensors, perception, and the world model, the system is not exchanging facts. It is exchanging beliefs about state over time. Each message between components should indicate the entity or hypothesis it is about (“visual target #99”, the current belief(s) (“could be a car, could be a plastic sheet in the wind”), confidence (“fifty-fifty”), and time horizon (“3 seconds out from our current position”).

Using the vehicle makes the examples values concrete for the discussion here. Applying the same in an enterprise workflow would involve several functions and most likely substantial differences in time scales.

The system should exchange beliefs with time and probability. Beliefs can then be handled, conflicts addressed so that planning and execution can proceed.

These questions need to be asked and answered, even if the enterprise workflow is completely manual. Workflow Automation without the answers to these would be very difficult:

- Who owns truth?
- How conflicts are resolved?
- What happens when data is missing or stale?
- How uncertainty propagates?

"Truth" in the above means "Belief", i.e. a statement of future with time and probability distribution.

### System components and their roles

Sensors, or other data measurement and input components in the system are there to measure, i.e. to provide as hard data as possible in their domain. Hard data does not mean infallible, so the correctness probability is almost never 100%.

Perception modules should be separate from sensors so that the interpretation of sensory data with errors in it is done separately and consistently. Perception publishes hypotheses, i.e. possible observations with probabilities.

Planning modules should query other parts, and reason the next actions for the separate control modules of the system.

World Model is there to fuse beliefs, remember them and handle conflicts. World Model owns the system state, it maintains the necessary objects (beliefs) and the identifiers globally, tracks those over time, and manages the lifecycle of these objects. As the last item implies, a belief that is past its time x probability window is no longer relevant.

Control is the execution part of the system.

## Summary – what is a World Model?

A World Model in this context is a system component that stores and manages beliefs, hypothesis and similar with time and probability. World Model's purpose is to address questions and queries along the lines of "What behaviors are consistent with current evidence, and which are dangerous?"

That is useful in an automated system in an open-ended, open-world context. The World Model stores, maintains and processes the boundaries of system capabilities and as such provides mechanisms to resolve conflicts and escalate to outside the system, to humans, when the situation so requires.

For an autonomous system, disagreements between parts are signal, not noise. The disagreements need to be resolved and noted so that the World Model can handle it with its machinery to handle ambiguity.

In terms of Ainolabs CMLS Architecture, a World Model is Contradiction Processing with the help of Long Term Memory. The general architecture is designed for fully autonomous systems, while the World Model in this context has mode modest goal to support Enterprise Workflow Automation by Smart Systems.