

ITEM #82 — Environment & Choice Module: World Model and Planning Engine

Background & Inspirations

Originating from Matthew A. Wilson and Demis Hassabis's studies on hippocampal replay and model-based planning. Wilson's work on hippocampal replay shows how the brain compresses and replays spatiotemporal experience for optimization. Hassabis emphasizes building internal world models and planning mechanisms that enable goal-directed decision-making.

Position in DBM Architecture

The World Model and Planning Module form the third core layer of the Digital Brain Model (DBM) — the Environment & Choice layer. This layer grants the system the ability to internally simulate the environment and plan future actions, evolving beyond reactive perception-memory intelligence.

World Model Design

Purpose: to construct a simplified but evolving internal representation of the environment, integrating multi-modal perception and enabling future-state prediction. Java Interface Example: class WorldModel { EnvironmentState perceive(EnvironmentInput input); SimulatedOutcome simulateAction(ActionPlan plan); double evaluateOutcome(SimulatedOutcome outcome, Goal goal); }

Planning Module Design

Purpose: to perform goal-driven search within the simulated environment, integrating the Rules Engine for strategy generation, scoring, and selection. Java Interface Example: class PlanningModule { List generatePlans(Goal goal, EnvironmentState currentState); ActionPlan selectBestPlan(List plans, EvaluationPolicy policy); void learnFromOutcome(ActionPlan executedPlan, RealOutcome feedback); }

Integrative Value

- World Model — gives DBM an internal world, turning it from a passive perceiver into an active modeler.
- Planning Module — gives DBM the will and capacity to choose.
- Rules Engine — completes the 'Environment–Rules–Action' loop, integrating causal reasoning and strategy synthesis.

Summary

If the Differential Tree gives DBM its memory and perception, then the World Model and Planning Module give it its mind — the ability to imagine, simulate, and choose within an environment.