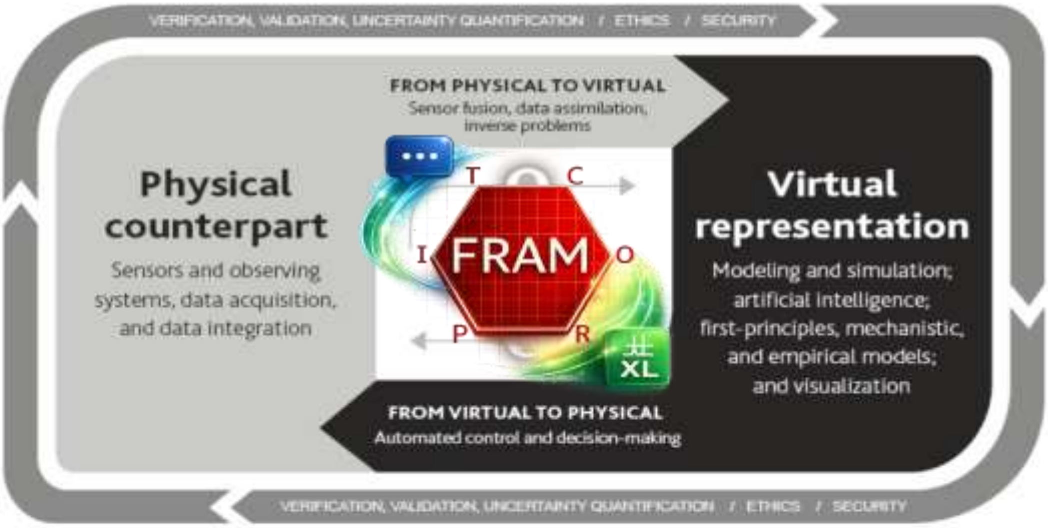

Epistemic Digital Twins (EDT)

The Epistemic Digital Twin can be understood as an agentic AI-based system management framework in which operational decisions are supported through continuous, functionally grounded modelling, hypothesis generation, and exploration of possible futures, rather than through static rules or optimisation alone.

What we are proposing can be stated quite cleanly, above, but its implications are far-reaching. At its core, the idea is to move beyond the notion of a digital twin as a passive representation of a system, and instead construct an **operationally closed, adaptive, epistemic agent**: a system that continuously incorporates real-world signals into an internally organised cycle of modelling, prediction, interpretation, and response, thereby maintaining an evolving, actionable understanding of the environment in which it is embedded.

In practical terms, this agent is realised through the coupling of four distinct but complementary elements. A **Functional System Model**[1] (**FRAM-based),** provides the underlying grammar of the system: what functions exist, how they depend on one another, and how variability propagates. A **dynamic execution layer** (FSM or FMI-style interpretation) allows that structure to be enacted over time, producing trajectories rather than static descriptions. A **metadata layer**, maintained in a transparent environment (such as Excel), carries the current condition of the system—its queues, delays, workloads, resource states, and signal qualities—so that the model is always anchored in a changing present. Finally, an **LLM-based epistemic layer** interprets incoming information, generates hypotheses, explains model behaviour, and proposes candidate interventions.

---

[1] For clarity, FSM is used here to mean Functional System Modelling, not finite-state machine.
Slater, D., (2026), The Evolution of Functional Systems Modelling, ResearchGate, https://www.researchgate.net/publication/406952025_The_Evolution_of_Functional_Systems_Modelling
Slater, D., and Hill, R., (2026) Interactive Functional System Modelling II: Toward an Executable Modelling of Emergence, ResearchGate, https://www.researchgate.net/publication/407030154_Interactive_Functional_System_Modelling_II_Toward_an_Executable_Modelling_of_Emergence

The innovation lies not in any single component, but in their integration into a closed loop of sense-making. Incoming observations are not merely displayed; they are mapped onto functional state. The model is not merely consulted; it is executed. Outputs are not merely reported; they are interpreted and questioned. The system does not simply predict; it generates alternative explanations and explores counterfactual futures. Human decision-makers are not replaced, but are provided with structured, auditable options grounded in both model dynamics and evolving evidence. This has been realised and implemented as a digital workbench **FRAIXL**[2].

This transforms the role of modelling. Instead of producing a static artefact that describes how a system ought to behave, the model becomes a **continuously operating explanatory process**. It can answer not only "what is happening?" but "why is it happening?", "what might happen next?", and "what would change if we intervened here?". In doing so, it shifts modelling from retrospective analysis or offline simulation into the domain of **ongoing operational support**.

An FSM as a dynamic state field, rather than a static spreadsheet

Basic Idea -

"The FRAIXL spreadsheet can be interpreted as a projection of a dynamic state landscape, and FMI cycles represent successive samplings of that landscape as the system evolves through it".

The apparent history—fatigue, depletion, prior decisions—is not lost; it is **encoded within the current state variables themselves**. Thus, the transition from one cycle to the next is governed by the present state and the defined functional relations (including any Δt effects), satisfying

[2] FRAIXL is the architectural workbench for recording, using and manipulating FSM built system models.

the Markov property that **the next state is conditionally independent of the past given the present state**. In this way, FSM–FRAIXL does not ignore history; it **compresses it into state**, allowing a path-dependent system to be treated as Markovian at the level of its state representation.

FSM Temporal State Principles

Together, these principles define a transition from static FRAM representations to an FSM - dynamic, temporally grounded, state-based systems in which behaviour emerges through iterative state evolution, real-time synchronisation, and progressive model refinement.

Principle 1 — Initial State Representation

An FSM model constitutes a fully initialised representation of system state at time t_0, defined by its functions (structure), couplings (dependencies), and metadata (state values).

Principle 2 — State Synchronisation (Observed System)

When coupled to a live system, the FSM model can be continuously updated such that its metadata reflects the current observed state of the real system in (near) real time.

Principle 3 — Selective State Update (Functional Locality)

During each FMI execution cycle, only those functions that are activated update their associated state variables, while all other state values persist unchanged unless explicitly modified.

Principle 4 — Dual State Evolution (Event vs Time Driven)

System state evolves through both function activation (event-driven change) and elapsed-time processes (time-driven change), requiring explicit representation of variables that change independently of function execution.

Principle 5 — State Persistence and Path Dependence

Each FMI cycle operates on the state produced by the previous cycle, not the initial state, resulting in a history-dependent system in which behaviour reflects accumulated prior conditions.

Principle 6 — Cycle Time vs Temporal Time

FMI cycles represent discrete state update steps (k), whereas temporal time (t) represents real-world elapsed duration; these are independent unless explicitly linked within the model.

Principle 7 — State-Triggered Execution

FMI execution cycles can be triggered by temporal thresholds or state conditions rather than fixed iteration, enabling event-driven and asynchronous model evaluation.

Principle 8 — Dual State Representation (Observed vs Modelled)

A FRAIXL implementation can maintain concurrent representations of system state: one derived from real-world observation and one generated by model execution, each providing a complete but distinct view of the system.

Principle 9 — Model Refinement Through State Discrepancy

The difference between observed and modelled system states provides a structured basis for refining the model, including adjustment of parameters, couplings, and functional structure.

Principle 10 — LLM-Assisted Model Refinement

An LLM can be employed to analyse discrepancies between observed and predicted states and propose structured, traceable modifications to improve the fidelity of the FSM/FMI model.

Principle 11 — LLM as Epistemic Reasoning Layer

An LLM can operate over the FSM–FRAIXL state representation as an epistemic reasoning agent, using the structured model (functions, couplings, metadata, and dual state sheets) to generate, evaluate, and compare hypotheses about system behaviour.

The FSM–FRAIXL model provides a structured epistemic representation of the system, over which an LLM can perform constrained reasoning, enabling explanation, prediction, and hypothesis generation grounded in the model's functional dynamics.

Principle 12 — LLM-Constructed Epistemic FSM Models

Given sufficient process documentation and an appropriate prompt structure, an LLM can generate internal representations of dual FSM-like models (observed and predicted states) and use them as a basis for reasoning, without explicitly instantiating them in FRAIXL.

An LLM can internally construct and reason over FSM-like dual state representations derived from process descriptions, but these remain implicit and lack the structural rigor, persistence, and executability of a formal FRAIXL implementation.

Discussion

The importance of this shift is particularly evident in complex socio-technical systems, where behaviour emerges from interactions rather than from the reliability of individual components. Traditional digital twins, grounded in physics or statistical prediction, struggle to represent the adaptive, context-sensitive nature of human and organisational activity. By contrast, a functionally grounded, executable model can represent interdependencies, timing effects, informal adaptations, and the propagation of variability across organisational boundaries. When coupled with an epistemic layer capable of interpreting narrative information and generating hypotheses, the result is a system that can engage with both the formal and informal dimensions of real-world operation.

This also reframes established practices such as hazard analysis, simulation, and incident investigation. Rather than being discrete activities conducted at particular moments, they become modes of a single underlying capability. The same system that monitors current operations can be used to explore potential failures, reconstruct past events, test design changes, or rehearse future scenarios. In this sense, the agentic twin becomes a **unifying platform for understanding, exploring, and governing system behaviour**.

Conclusion

The conceptual shift can be summarised succinctly. A conventional digital twin is a model of a system. An agentic digital twin, as defined here, is a **participant in the ongoing interpretation and management of that system**. It does not merely mirror reality; it engages with it. It maintains a continuously updated account of how the system is functioning, explores how it might evolve, and supports decisions about how it should be guided.

This is why the idea is both innovative and important. It opens a path toward systems that do not simply analyse complexity after the fact, but operate within it—helping to make sense of it, simulate it, anticipate it, and shape it in real time.

So, by combining functional modelling, dynamic execution, and epistemic interpretation within an operationally closed loop, the EDT moves beyond the limitations of conventional digital twins and opens new possibilities for continuous, model-driven sense-making. Rather than treating models as static artefacts, or black boxes of statistics, the EDT treats modelling as an ongoing process of understanding, in which observations, explanations, and interventions are continuously integrated and refined, supporting both operational decision-making and the broader goal of managing uncertainty in an increasingly dynamic world.

David Slater