



Modelling Contemporary Complex Systems: From Structure to Possibility

Abstract

This piece explores the challenges of modelling contemporary complex systems, which are characterized by nonlinearity, feedback, adaptation, and emergence. It distinguishes complex systems from complicated ones using the Cynefin framework and highlights the limitations of traditional modeling approaches that rely on predefined structures and stable boundaries. A structural-semantic classification of modeling methods is proposed, emphasizing semantic substrate, structural commitment, and representational ontology. The Functional Resonance Analysis Method (FRAM) is introduced as a metamodel that focuses on functional dependencies and variability, enabling sensemaking under uncertainty. FRAM's application in digital twinning is discussed, showcasing its ability to dynamically adapt to real-world system behaviour. The document concludes by advocating for diverse modelling methodologies to address the complexity of modern systems, with FRAM playing a pivotal role in modelling emergent and unpredictable behaviours.

From Complicated to Complex Systems

Across engineering, safety, healthcare, infrastructure, finance, and AI-enabled socio-technical domains, there is growing recognition that many systems of contemporary concern are complex rather than merely complicated. This distinction, articulated clearly in the Cynefin framework, (Figure 1), is not semantic but foundational (Snowden & Boone, 2007). Complicated systems may involve many parts, yet their behaviour remains largely decomposable, analysable, and predictable given sufficient expertise. Complex systems, by contrast, are characterised by nonlinearity, feedback, adaptation, and emergence; their behaviour cannot be reliably inferred from the properties of individual components alone (Cilliers, 1998; Mitchell, 2009).

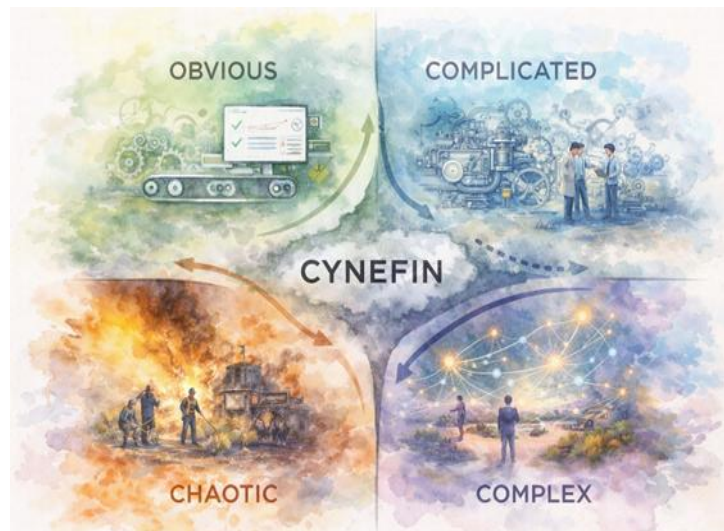


Figure 1 - The basic Cynefin system state categories

In complicated domains, reductionist techniques—deterministic causal chains, optimisation over fixed architectures, and exhaustive decomposition—remain effective. In complex domains, the same techniques routinely fail. Interventions produce unintended consequences, behaviour drifts over time, and identical actions yield different outcomes under apparently similar conditions. What matters is not simply what the system is made of, but how activity unfolds through time, how local variability interacts, and how global patterns emerge from contingent interactions (Holland, 1998).

This shift from complication to complexity has profound implications for modelling. Many traditional modelling approaches assume stable system boundaries, predefined structures, consistent cause–effect relationships, and repeatable execution paths. These assumptions are increasingly violated in real operational settings. Modern socio-technical systems are adaptive, partially autonomous, tightly coupled, and context-sensitive. Human operators, automated subsystems, organisational pressures, environmental conditions, and learning components interact in ways that cannot be fully anticipated at design time. As a result, understanding behaviour—not merely structure—becomes the central modelling challenge (Rasmussen, 1997).

The modelling question therefore shifts from “*How does the system work?*” to “*How can system behaviour vary?*”, “*Under what conditions do particular patterns of interaction arise?*”, and “*Which modelling commitments should be made before behaviour is observed, and which should be deferred until instantiation?*” These questions are not about optimisation. They are about sensemaking under uncertainty. Addressing them requires moving beyond any single modelling paradigm and towards a principled classification of modelling approaches based on what they fundamentally assume about systems.

A Structural–Semantic Classification of Modelling Approaches

The proliferation of modelling methods applied to complex systems has created an apparent abundance of choice, yet also persistent confusion. Methods are often grouped superficially—by notation, computational technique, or application domain—rather than by their ontological and semantic commitments. To classify modelling approaches meaningfully in the context of

complexity, distinctions must be drawn at a deeper level: not what the model looks like, but what it presupposes about structure, dynamics, and representation.

The classification developed here rests on three orthogonal axes. Importantly, this classification concerns what is constitutive of a modelling approach—what it requires in order to function—not what may be added or imposed in particular implementations.

Semantic substrate: what actually “runs” the model

The first axis is semantic substrate—the mechanism by which the model evolves through time. Some approaches evolve by solving continuous equations (System Dynamics, hybrid state-space models, stochastic differential equations), others advance through discrete-event scheduling (DES, DEVS, Petri nets), by rule execution at the level of individual entities (ABM), or by probabilistic inference over time (Dynamic Bayesian Networks) (Sterman, 2000; Law, 2015; Macal & North, 2010; Pearl, 2009).

Each substrate embeds implicit assumptions about determinism, uncertainty, observability, and repeatability. Two models may both be “dynamic” while resting on entirely different semantic foundations. Treating them as equivalent because they generate time-series outputs obscures these differences and leads to inappropriate method selection. This distinction is rarely explicit, yet it matters.

How structure enters: predefined topology versus contingent realisation

The second axis concerns structural commitment. In structurally specified families—such as System Dynamics, Dynamic Bayesian Networks, Petri nets, and DEVS—the topology of interactions is an intrinsic property of the model. Edges, couplings, or arcs are declared a priori, and behaviour is constrained to unfold on that predefined structure. Altering the topology fundamentally alters the model (Zeigler et al., 2000).

By contrast, in process- or dynamics-specified families—such as ABM, DES, hybrid state-space models, or data-driven SDE discovery—behaviour can exist without any explicit topological commitment. Structure may be implicit, emergent, inferred, or transient. Where topology appears, it is secondary to the underlying process semantics. This distinction is critical because complex systems are precisely those in which structure is often context-dependent and temporally unstable.

What is represented: components versus functions and constraints

The third axis addresses representational ontology. Many approaches represent systems in terms of components, agents, resources, queues, or state variables—entities that function as surrogates for parts of a system. Other approaches instead represent systems in terms of functional necessities, constraints, and conditions—what must be done, what must be available, and what must hold for activity to occur.

This distinction is particularly important for socio-technical systems, where the same function may be realised by different actors or technologies over time, and where binding behaviour too tightly to components obscures adaptability and substitution (Checkland, 1999; Suh, 2001).

This gives us a useful way of categorising (Table 1), the different methods (horses for courses?). SysML is included although SysML and UML are *not executable semantics in themselves*. They are descriptive and prescriptive modelling languages. Any “behaviour” comes only once they

are mapped onto something else (simulation engines, state machines, code generators, DES back-ends, etc.). In other words, they *borrow* a semantic substrate rather than providing one.

Table 1 - Classification of Contemporary Complex-Systems Modelling Methods

Method family	Semantic substrate	How structure enters	What is represented	Structural status
UML / SysML	None, intrinsic (descriptive)	Explicit, predefined topology	Components, states, interfaces, flows	Strongly structurally specified
System Dynamics (SD)	Continuous stock-flow equations	Explicit, predefined topology	Abstract accumulations & feedback	Structurally specified
Dynamic Bayesian Networks (DBN)	Probabilistic temporal inference	Explicit, predefined DAG	Random variables & dependencies	Structurally specified
Petri Nets	Token flow & firing rules	Explicit, predefined bipartite graph	Places, transitions, tokens	Structurally specified
DEVS	Discrete-event semantics	Explicit, predefined modular couplings	Components & event ports	Structurally specified
Discrete-Event Simulation (DES)	Event scheduling	Implicit in process logic	Processes, queues, resources	Process-specified
Agent-Based Modelling (ABM)	Agent rule execution	Emergent or optional	Agents & local interactions	Process-specified
Hybrid dynamical systems	State-space flows + mode switches	Implicit in equations/guards	State variables	Dynamics-specified
SDE / data-driven dynamics	Stochastic state evolution	Implicit or inferred	State variables	Dynamics-specified
FRAM (model)	Functional availability & variability	None predefined (compatibility only)	Functions & aspects	Non-structural metamodel
FRAM (instantiation)	Conditional activation	Transiently realised	Event traces	Ephemeral structure

This classification clarifies why FRAM cannot be grouped with predetermined-topology digraph models, despite superficial visual similarities. At the model level, FRAM does not specify edges, paths, or couplings. Aspect labels define *compatibility*, not connectivity. The FRAM model therefore contains no topology to traverse (Hollnagel, 2012).

Structure appears only at the level of instantiation, when particular functions fire, outputs become available, and transfers occur. These transfers are contingent, transient, and repeatable only in a statistical sense. A single instantiation can be reconstructed retrospectively as a directed trace network, but this structure is not preserved nor guaranteed

across instantiations. The decisive distinction is therefore not whether a model can be drawn as a graph, but whether connectivity is an ontological commitment of the model itself (Hollnagel et al., 2015).

Why Metamodels Are Necessary for Complex Systems

As systems increase in scale, coupling, and socio-technical heterogeneity, the limiting factor in modelling is no longer computational power but cognitive and representational manageability. Large infrastructures and team-based operations rapidly exceed the limits of models that rely on detailed component descriptions or exhaustive process logic. As detail accumulates, interactions multiply faster than understanding; models become brittle, opaque, and difficult to validate (Rasmussen, 1997).

Metamodels address this problem by operating at a different level of abstraction. Rather than representing systems as assemblies of parts or fixed flows, they represent classes of behaviour, functional necessities, and constraints that remain meaningful across multiple instantiations. Their purpose is not to predict exact trajectories, but to make visible patterns of interaction, dependency, and variability without premature structural commitment.

From this perspective, detail is not inherently valuable. Detail is useful only insofar as it supports discrimination between relevant behavioural possibilities. Beyond a certain point, additional detail obscures insight rather than sharpening it (Cilliers, 1998).

Modelling for Emergence and Surprise

The defining challenge of contemporary complex systems is the persistent emergence of behaviours that were never explicitly designed or anticipated. Structurally specified models excel where topology is stable but implicitly assume that emergence occurs within known interaction frameworks. Process- and dynamics-specified models can generate emergence computationally, but often struggle to support interpretation, generalisation, and governance once surprise occurs.

Metamodels offer a different capability. By constraining the space of possible behaviours rather than enumerating outcomes, they support reasoned anticipation of the conditions under which surprise may arise, rather than prediction of specific emergent events. FRAM exemplifies this approach. By defining a stable set of functions and aspect compatibilities, it creates a behavioural landscape in which multiple, distinct instantiations are possible—some of which may be unexpected—without requiring the model itself to be rewritten (Hollnagel, 2012; Patriarca et al., 2017).

This represents a fundamentally different notion of prediction. It is closer to preparedness than to forecasting.

FRAM and digital twinning of complex systems

Beyond its role as a conceptual functional metamodel, FRAM has increasing practical relevance through its implementation in the FRAM Model Visualiser (FMV) and FRAM Model Interpreter (FMI). Together, these tools enable FRAM models to be instantiated, executed, and iteratively updated using live feeds from actual system behaviour. Functions may be activated or inhibited, couplings may materialise or decay, and variability can be explored across

repeated instantiations. This allows the model to evolve in response to *Work-as-Done*, rather than remaining fixed at the level of design intent or analyst expectation.

This capability positions FRAM as a credible candidate for a digital twin paradigm appropriate to complex socio-technical systems. (Grieves, M. and Vickers, J. (2017). In contrast to conventional digital twins—typically based on state-space representations, physical component fidelity, or tightly specified execution graphs—a FRAM-based twin does not attempt to reproduce system state exhaustively or predict precise trajectories. Instead, it mirrors the functional organisation and interaction logic of the system: which functions are performed, under what conditions, with what variability, and with what downstream effects. The twin therefore operates at the level of *functional behaviour* rather than *component state*.

Crucially, this shifts the role of the digital twin from prediction to continuous sensemaking and learning. FMI-driven instantiation allows observed behaviours to be replayed, compared, and contrasted across scenarios, revealing patterns of functional resonance, drift, and adaptation that are otherwise difficult to capture. In this sense, FRAM supports a form of digital twinning that remains meaningful under reconfiguration, substitution, and surprise—conditions under which state-centric or structurally fixed twins often degrade or fail. For AI-enabled and adaptive systems, where behaviour is shaped as much by interaction and context as by internal state, this functional digital twin perspective offers a complementary and, in some domains, more appropriate modelling foundation.

Conclusion

No single modelling methodology can address the full range of behaviours exhibited by modern complex systems. Structurally specified models remain indispensable for stable subsystems; process- and dynamics-specified models are valuable for exploring emergence through simulation. However, neither is sufficient where structure is fluid, coordination informal, and outcomes contingent.

Within this landscape, FRAM occupies a critical and under-represented role. By separating model from instantiation, and functional necessity from realised structure, FRAM provides a principled means of modelling systems in which *Work-as-Done* diverges from *Work-as-Imagined* and where local variability is both inevitable and essential. Its strength lies not in predicting trajectories, but in making visible the functional dependencies and resonance conditions that shape emergent behaviour.

FRAM does not replace other modelling approaches; it enables analysts to model coherently genuinely complex domains. Where systems defy stable structural representation, and where the most consequential behaviours are those that were never explicitly designed, FRAM provides a means of modelling possibilities rather than (perhaps prejudiced?) certainty.

David Slater

References

- Checkland, P. (1999) *Systems Thinking, Systems Practice*. Chichester: Wiley.
- Cilliers, P. (1998) *Complexity and Postmodernism: Understanding Complex Systems*. London: Routledge.
- Grieves, M. and Vickers, J. (2017) 'Digital Twin: Mitigating unpredictable, undesirable emergent behavior in complex systems', in Kahlen, F.-J., Flumerfelt, S. and Alves, A. (eds) *Transdisciplinary Perspectives on Complex Systems*. Cham: Springer, pp. 85–113. https://doi.org/10.1007/978-3-319-38756-7_4
- Holland, J.H. (1998) *Emergence: From Chaos to Order*. Reading, MA: Addison-Wesley.
- Hollnagel, E. (2012) *FRAM: The Functional Resonance Analysis Method*. Farnham: Ashgate.
- Hollnagel, E. (2014) *Safety-I and Safety-II: The Past and Future of Safety Management*. Farnham: Ashgate.
- Hollnagel, E., Wears, R.L. and Braithwaite, J. (2015) *From Safety-I to Safety-II: A White Paper*. The Resilient Health Care Net. Available at: <https://resilienthealthcare.net> (Accessed: 15 December 2025).
- Law, A.M. (2015) *Simulation Modeling and Analysis*. 5th edn. New York: McGraw-Hill Education.
- Liu, M. and Xu, C. (2021) 'Review of digital twin about concepts, technologies, and industrial applications', *Journal of Manufacturing Systems*, 58, pp. 346–361. <https://doi.org/10.1016/j.jmsy.2020.06.017>
- Macal, C.M. and North, M.J. (2010) 'Tutorial on agent-based modelling and simulation', *Journal of Simulation*, 4(3), pp. 151–162. <https://doi.org/10.1057/jos.2010.3>
- Mitchell, M. (2009) *Complexity: A Guided Tour*. Oxford: Oxford University Press.
- Patriarca, R., Bergström, J. and Di Gravio, G. (2017) 'Defining the functional resonance analysis space: Combining abstraction hierarchy and FRAM', *Reliability Engineering & System Safety*, 165, pp. 139–152. <https://doi.org/10.1016/j.ress.2017.03.032>
- Patriarca, R. et al. (2020) 'Framing the FRAM: A literature review on the Functional Resonance Analysis Method', *Safety Science*, 129, 104827. <https://doi.org/10.1016/j.ssci.2020.104827>
- Pearl, J. (2009) *Causality: Models, Reasoning, and Inference*. 2nd edn. Cambridge: Cambridge University Press.
- Rasmussen, J. (1997) 'Risk management in a dynamic society: A modelling problem', *Safety Science*, 27(2–3), pp. 183–213. [https://doi.org/10.1016/S0925-7535\(97\)00052-0](https://doi.org/10.1016/S0925-7535(97)00052-0)
- Saporiti, N. et al. (2023) 'Challenges and countermeasures for digital twin implementation in manufacturing', *Journal of Manufacturing Processes*, 86, pp. 486–503. <https://doi.org/10.1016/j.jmapro.2023.01.017>
- Slater, D. and Hill, R. (2022). 'Quantitative FRAM and metadata-driven interpretation of functional variability', *Reliability Engineering & System Safety*, 221, 108315.
- Snowden, D.J. and Boone, M.E. (2007) 'A leader's framework for decision making', *Harvard Business Review*, 85(11), pp. 68–76.

Sterman, J.D. (2000) *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Boston, MA: McGraw-Hill.

Suh, N.P. (2001) *Axiomatic Design: Advances and Applications*. Oxford: Oxford University Press.

Zeigler, B.P., Praehofer, H. and Kim, T.G. (2000) *Theory of Modeling and Simulation*. 2nd edn. San Diego, CA: Academic Press.