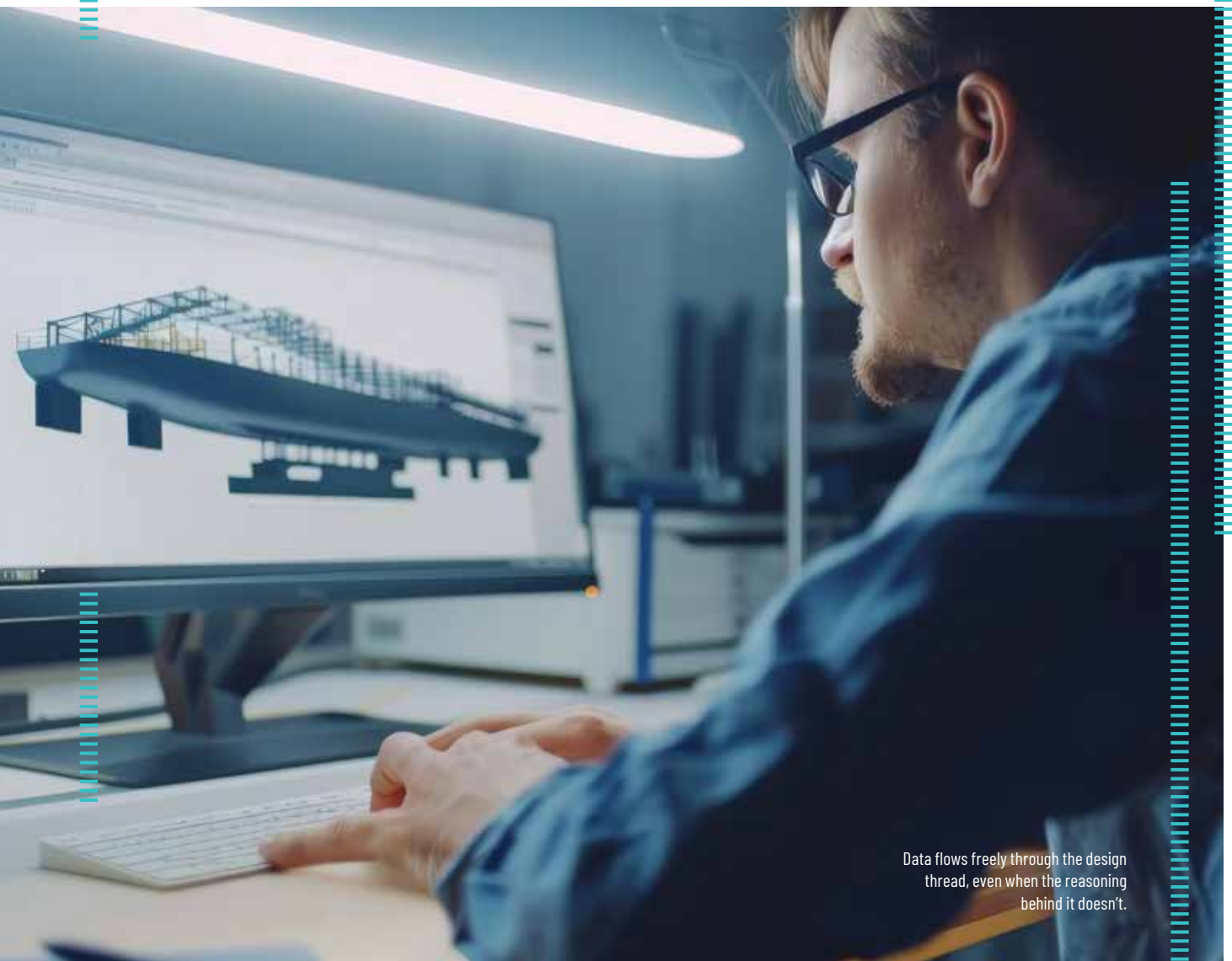


FROM DIGITAL THREAD TO DIGITAL COGNITION

THE NEXT ADVANCE
IN SHIP DESIGN
INTELLIGENCE IS
NOT MORE AI—IT IS
STRUCTURED COGNITION

BY STEVE BOISVERT
AND TERRA MULLIS



Data flows freely through the design thread, even when the reasoning behind it doesn't.

Every naval architect knows the design spiral is supposed to converge. It does not always cooperate. A change to top-side weight reopens stability. A revised propulsion arrangement shifts structural frame spacing. A late classification society query resets fire protection zoning. The spiral is not a flaw in the process. It is the process.

Ship design has built a digital thread. Geometry flows between design systems. Models are linked. Data moves across disciplines without being retyped. By most measures of digital transformation, the industry has made real progress.

What the thread does not carry is reasoning. It connects geometry, not the logic behind it. Requirements sit in documents the thread does not read. Class rules live in databases or spreadsheets the thread does not evaluate. The trade-off a naval architect made in the third iteration of the design spiral, why they accepted a reduced stability margin to hit a displacement target, is in meeting notes the thread has never seen. When that decision matters again in the fifth iteration, or on the next program, an engineer reconstructs it from memory, or loses it entirely.

Model-based cognition (MBC) is an artificial intelligence (AI) architecture for changing that. The goal is not to replace the digital thread. It is to give the thread something it currently lacks: the ability to manage (capture, codify, curate) and execute (provision, process, persist) the knowledge and intelligence behind the geometry. That is what tells engineers what the data means, what constraints apply, and what to do when conditions change. The aim is not to automate the spiral away. It is to give engineers intelligence that matches the complexity of what they are actually doing.

The automotive industry has been running MBC in production for over a decade. Global OEMs and tier 1 suppliers have deployed it across design review, requirements validation, lessons learned, and program management. Those results give ship designers a concrete reference point to jump start their own process improvements.

WHAT THE THREAD CONNECTS

Ship design has invested heavily in digital transformation. Model-based systems engineering (MBSE) frameworks, 3D-integrated design environments, digital thread architectures, and AI-assisted tools have all advanced the state of



practice. The thread is real and valuable. The problem is that it was designed to carry data and geometry, not the intent or reasoning behind a design decision. Two decades of investment has advanced the state of the practice, yet the thread goes quiet exactly where the most consequential decisions are made and where the interdependencies are densest.

A well-constructed digital thread connects geometry across systems, but it does not connect the reasoning behind that geometry. When a design change in one system requires a coordinated response across several others, the geometry propagates but the governing logic does not. Human coordination fills that gap. At the scale of a program with hundreds of interacting requirements and dozens of contributing disciplines, filling that gap manually is where programs lose time, quality, and institutional knowledge.

The models that ride the thread fall into recognizable categories. Parametric models, the computer-aided design (CAD), finite element analysis, and computer-aided engineering representations at the core of every engineering tool chain, capture geometry and physics. In short, they capture what a design is. Systems modeling languages such as SysML capture architecture and behavior, how a design decomposes, how its parts interface, and where requirements are allocated. Both are indispensable. Neither encodes a third category that matters just as much: the explicit, executable knowledge governing how the work is done. The requirements logic, acceptance criteria, process governance, and institutional knowledge that determine what compliance actually means and who may decide it.

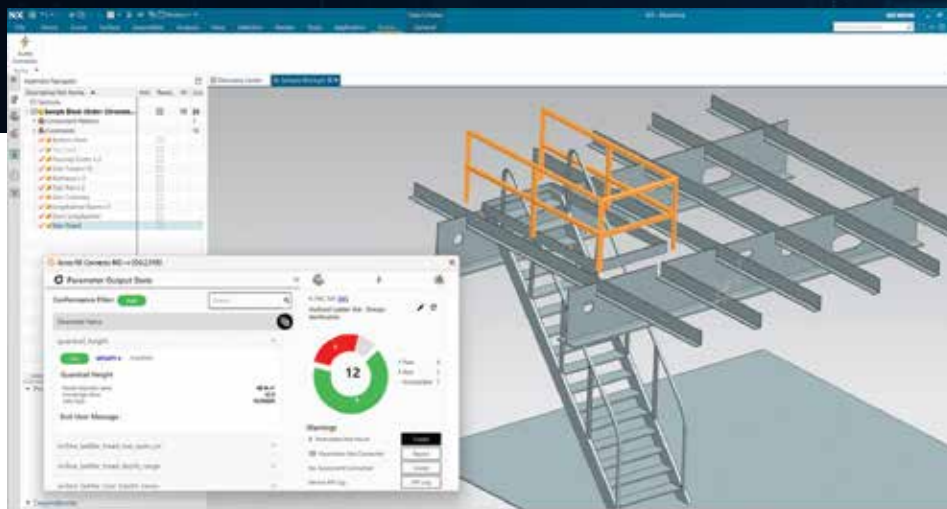
A cognitive layer fills that gap. It is complementary to parametric and systems models, not a replacement for them. Where parametric models are expressed in continuous mathematics and SysML in specialist graphical notation, this third category takes the form of no-code structured symbolic models, a discrete, logic-structured representation that a domain expert authors directly, without writing software or handing the work to a systems engineer.

It is worth being precise about what is being managed and executed, because the value is in three things, not two. A system that stores knowledge but reasons over none of it (a library or a wiki) and a system that reasons but answers to no governed knowledge (a large language model or LLM chatbot) are each genuinely useful, and each incomplete in the same way. Both hold one of the two substances and not the relationship between them. That relationship, the explicit, governed link between a specific requirement and the specific reasoning that acts on it, is the co-equal third element, and it is the one neither a document store nor a generative model can carry. A structured symbolic model exists to carry it. The reasoning travels with the knowledge it applies to, the binding is explicit, and it stays auditable as either side changes.

Generative and large language models are powerful for drafting, pattern recognition, and document synthesis. In a design spiral context, where every recommendation must be auditable and where the knowledge base includes proprietary engineering logic that cannot be exposed to external platforms, black-box AI creates as many problems as it solves. A system that cannot explain why it flagged a stability

Firemain piping is simple to look at, but deceptively hard to coordinate across class rules, owner specs, and everything else riding on it.

FROM DIGITAL THREAD TO DIGITAL COGNITION



Guardrail height is checked automatically against a knowledge model instead of a calculation sheet built by hand.

concern is not a reasoning partner. The next advance is not more AI. It is structured reasoning an engineer can audit.

WHERE THE THREAD GOES QUIET

Three failure modes recur across shipyard design workflows. In each case, the digital thread is functioning as designed. The breakdown occurs because the engineering knowledge that determines what the data means is absent.

ONE: Late compliance discovery. Classification society rules (for commercial vessels) and navy specifications and design sheets govern virtually every major design decision in the spiral, from structural scantlings to fire protection zoning to piping layouts. Those rules exist as documents. Shipyards manually translate them into spreadsheet-based calculation sheets, a time-consuming process. By the time a sheet is built, the concept design has moved. If a classification society releases an updated rule during an active program, that update does not automatically reach the check sheets already in use. Engineers continue designing against the previous version. The error surfaces at drawing submittal, if it surfaces at all.

Ship structural design makes this tangible. Section modulus calculations, plating thickness, and frame spacing all depend on class rules that reference other rules, which reference standards revised at different rates. A structural engineer in functional design is navigating a document dependency tree no one has explicitly mapped, in a CAD environment with no awareness of the regulatory logic governing what they are drawing.

TWO: Requirement fragmentation. Firemain piping depends simultaneously on class rules for vessel piping systems, owner

specifications, producibility constraints from shipyard facility capabilities, and classification notations negotiated separately from the base rules. Those sources do not reference each other. Each discipline carries a partial view. When a piping layout conflicts with a structural arrangement updated since the last coordination meeting, neither system detects it.

THREE: Knowledge that never re-enters the spiral. Hull penetration design is one of the highest-frequency rework drivers in detail design. Penetration geometry, insulation requirements, fire boundary classifications, and structural compensation rules interact in ways not obvious from any single document. Every shipyard has engineers who know which combinations cause problems. That knowledge lives in those engineers. When they move on, it does not transfer. The next designer encounters the same problems by the same manual search process.

The common thread across all three failures is not a lack of data. The geometry is in the models. The rules are in the documents. The problem is that neither the data nor the documents are executable. They cannot perceive a design change, evaluate their own applicability, or surface a conflict when a decision is being made. Engineers spend an estimated 1.8 hours per day searching for information, time spent navigating what the thread cannot carry rather than designing.

EXECUTABLE IN PRACTICE

What changes when knowledge and intelligence become executable rather than passive can be measured. The automotive industry has run MBC in production long enough to produce numbers relevant to ship design.

In results reported in 2021, one major automotive OEM deployed MBC in its seating division, where engineers

Eight cognitive functions and one convergence point.
This is what a thread that can think actually looks like.



TABLE 1: EIGHT COGNITIVE FUNCTIONS

COGNITIVE FUNCTION	WHAT IT DOES	IN THE DESIGN SPIRAL
LANGUAGE	SMEs author structured, symbolic models that express meaning, not just geometry	Requirements, rules, and constraints encoded once, reusable across programs
PERCEPTION	Execution models interpret design changes and measurements automatically	CAD changes assessed against requirements in real time
ATTENTION	Dynamic provisioning filters knowledge to what is context-relevant	Each engineer sees only the constraints that apply to current work
LEARNING	New information is consolidated and integrated into the knowledge base	Design decisions, conflicts, and resolutions become persistent, reusable knowledge
MEMORY	Schema-bound persistence stores and retrieves knowledge across time and teams	Next vessel or spiral pass inherits full reasoning context, not just data
REASONING	Logic processors evaluate design conditions against requirement models	Compliance status, consequence chains, and trade-off projections computed continuously
PROBLEM SOLVING	Generative capability proposes targeted, model-derived interventions	Resolution options surfaced at the moment of conflict, not at the next review
DECISION MAKING	Heuristic tools evaluate and prioritize options; engineers remain the decision makers	Every proposal is transparent, traceable, and subject to human acceptance or rejection

managed hundreds of design requirements across materials, manufacturing, and supplier constraints. The team replaced PowerPoint-based peer reviews and static knowledge repositories with structured models that evaluated compliance in real time as designers worked. Peer review duration dropped from 30 hours to two. Warranty claims for seating declined steadily, contributing to top industry quality rankings. The division's knowledge management technical specialist described MBC as the backbone of every knowledge-based engineering application the division built, not merely a productivity tool.

A global tier 1 automotive supplier tested MBC differently, with results reported in 2019. Some 32 program managers across 10 countries needed a common structure for gate reviews, launch planning, and KPI reporting. The supplier used MBC to standardize the product development process and embed design guides and lessons learned into daily workflow. Within two years, 220 engineers across 18 locations had adopted it as their primary daily interface. The platform knew what step each engineer was on, what constraints applied, and what prior decisions mattered.

In both cases, the digital thread was not replaced. It was given something with which to work. That something was knowledge that was able to evaluate what the data meant, surface conflicts as decisions were being made, and accumulate rather than reset when engineers changed programs. In neither deployment did MBC replace engineering judgment. It gave engineers structured access to the right knowledge, at the right moment, in the context of the work already underway.

FROM DATA TO INTELLIGENCE

MBC adds a cognitive layer that gives the digital thread what it has always been missing. Rather than simply moving data,

The design review is the meeting where a trade-off gets made, and too often, where it quietly stays.



the thread can now interpret that data against governing logic, reason through the consequences of a change, and surface conflicts before they propagate. It does not replace the thread. It gives the thread the knowledge and intelligence to manage and execute, not merely the data to transport. Five principles define how it works.

ONE: A language domain experts can speak. Most knowledge management systems in ship design require knowledge to be formalized by someone other than the person who holds it. A systems engineer translates a naval architect's stability logic into an MBSE block diagram. A software developer encodes frame spacing rules into parametric models. At each step, fidelity is lost and latency is introduced.

MBC uses a structured, symbolic, no-code modeling language that domain experts use directly. The naval architect builds the stability requirement model. The structural engineer encodes frame spacing logic. The classification surveyor authors compliance rules with the full intent of someone who has actually applied them.

A classification society structural rule encoded as an executable model, rather than a PDF clause, can evaluate section modulus compliance against the current design geometry in real time. It does not wait for someone to build a calculation

sheet. When the rule changes, the model updates, and every active program that references it recalculates automatically.

TWO: Top-down and bottom-up. Expert-authored models only deliver value if they remain interoperable across disciplines, programs, and organizations. A stability model built by a naval architect needs to interact with a structural model, a regulatory model, and a production constraint model, without manual translation or one-off integration work.

MBC pairs top-down coordination with bottom-up authorship through a standardized semantic framework, a shared vocabulary of model types, syntax conventions, and structural patterns that ensures independently authored models remain interoperable. Think of it as a coordinate system for knowledge. Standardization need not be imposed up front. Measured reuse surfaces where a shared standard is already emerging, and that element can then be placed under a formal authorship contract that fixes how and by whom it may change.

Hull penetration design shows exactly why this matters. Penetration requirements span structural, fire protection, outfit, and production planning. No single discipline owns the full picture. With MBC, each discipline authors its portion of the penetration model directly. The standardized



Welding is where the geometry ends and the physical steel begins.

framework connects them. When a new program starts, the penetration knowledge from previous programs is available across all disciplines, not just in the memory of engineers who worked the last hull.

THREE: Eight cognitive functions in coordination. A senior naval architect is more valuable than a junior one not because they know more facts, but because their cognitive processes are better integrated. They perceive the significance of a stability margin faster, reason across more interdependencies at once, and draw on richer contextual memory when evaluating a trade-off. MBC maps each cognitive function to a corresponding computational capability and ensures they work together.

No single function works in isolation. When perception surfaces a design change, attention determines which requirements apply. Reasoning evaluates compliance, problem-solving proposes interventions where conflicts arise, and memory records the decision for the next iteration. For fire-main piping, when a route change is made in the 3D model,

the system identifies which class rules and owner specifications apply, evaluates compliance, and flags any conflict at the moment the change is made. The piping engineer sees what is wrong and why, with the reasoning traced back to the specific clause.

FOUR: Models that combine and recombine. In the design spiral, context shifts constantly. A vessel that begins concept design as a diesel-electric platform may, before contract design closes, take on a hydrogen fuel cell option, an ice class requirement, and a revised cargo deck spec that changes the topside weight budget. Each change does not just modify a parameter. It changes which models are relevant, which constraints are active, and which trade-offs need reopening.

A conventional requirements management system handles this through manual update and re-review. MBC handles it through dynamic recombination. Each knowledge model carries its own applicability logic, encoding the conditions under which it is relevant, the parameters it depends on, and the other with which models it interacts. When context



Compliance can be made visible in real time instead of discovered at drawing submittal.

changes, the system automatically reassembles the relevant set. When a classification society releases an updated rule, the knowledge model updates and every design check that depends on that rule re-evaluates against the new logic. No engineer needs to identify which calculation sheets reference the changed clause and rebuild them.

FIVE: Living system, not a static black box. A knowledge model authored at the start of a program should become more valuable over time, not less. The stability logic for one vessel class should be reusable and refinable for the next. A conflict discovered during contract design should propagate into the knowledge base so it cannot recur silently on a follow-on program.

Parametric models, analysis results, and trained AI models are all snapshots. They capture the state of a specific configuration at a specific moment. None of them grow from the accumulated expertise of the people using them. An MBC system does. Every time an expert refines a model, every time an engineer accepts or rejects a proposal, the system gets more accurate, across hundreds of users simultaneously. Hull penetration rework is the clearest example. The same geometry errors, fire boundary misclassifications, and structural compensation oversights recur across programs because the knowledge of what causes them lives in individual engineers. When a penetration conflict is resolved on one program, the resolution refines the model, enabling every future program to benefit from that knowledge instead of repeating the same mistake.

WHEN THE THREAD STARTS TO THINK

The geometry still travels through the thread. What changes

is what travels with it. Requirements become active models that evaluate the design state as geometry is created and modified, surface compliance conflicts at the moment a decision is being made and carry traceability back to the specific class rule clause, while the structural engineer is still working. First-time quality improves at the source, not at the review gate.

Trade-off decisions stop living in meeting minutes and become structured knowledge objects that are persistent, traceable, and available to the next team that faces the same decision. When a quality assurance engineer reviews a subcontractor weld procedure against military standards, he works against a structured knowledge base that maps each section of the procedure to applicable standard clauses, flags conflicts, and generates a traceable compliance record. No manual search through printed documents is required.

The experts who carry the most critical knowledge author it directly into the digital environment, with no IT intermediary in the loop and no translation lost along the way. Parametric models and system architecture describe what a ship is and how it is built. The cognitive layer carries the explicit knowledge and intelligence that drives the work forward, including the requirements, reasoning, and governance that determine whether what is being built is right.

Automotive programs have run this way for years. The ships being designed over the next two decades—autonomous, hybrid-energy, and subject to accelerating regulatory change—will be built on digital threads. What those threads need to carry is not just data. **MT**

STEVE BOISVERT is chief solutions consultant at AuroslQ. **TERRA MULLIS** is vice president of innovation and technology and senior aerospace consultant at AuroslQ.