A2GO · FOUNDER ESSAY

Why SCOR Still Matters in the Agentic Era

AI agents can read anything — so why would a process framework born in 1996 matter more now, not less? Because agents raise the price of ambiguity. A founder's argument for SCOR DS as the shared language between humans, systems, and the agents that now sit between them.

Stephen Hutson & Michael Romeri, co-founders, A2go

July 2026 · a2go.ai

A2go

The obvious objection

Here is the argument against this essay, stated as strongly as we can: large language models can read anything. They ingest your ERP exports, your contracts, your emails, your tribal knowledge in whatever dialect your company speaks it. If the machine can meet your supply chain where it is, why would anyone spend effort conforming to a reference model written before most of today's planners had email?

It's a fair question, and for a while we wondered ourselves. We now believe the opposite is true: **agents raise the price of ambiguity**, and a shared reference model is how you stop paying it.

What SCOR is, for the record

The Supply Chain Operations Reference model — SCOR — was developed in 1996 by the management consulting firm PRTM together with AMR Research, and was endorsed by the Supply Chain Council as the cross-industry standard for supply chain strategy, performance management, and process improvement. The Council later became part of what is today the Association for Supply Chain Management (ASCM), which stewards the model.

We know this model from practice. Both of us worked at PRTM as management-consultant SCOR practitioners, and we contributed to SCOR there — in engagement rooms, against real client supply chains, where the model earned its keep or didn't.

The current edition, the SCOR Digital Standard (SCOR DS), is open-access under Creative Commons and restructures the model around seven processes: **Orchestrate** at Level 0 — connecting the chain to suppliers, customers, and internal stakeholders — and six Level-1 processes: **Plan, Order, Source, Transform, Fulfill, Return**. Gone is the old linear pipeline; SCOR DS draws the supply chain as a network loop, because that is what supply chains actually are. Performance is expressed through eight attributes — reliability, responsiveness, agility, cost, profit, asset management, environmental, and social — each decomposing into a hierarchy of defined metrics.

Keep that word, *defined*. It's the whole essay.

Language precedes coordination

Every supply chain organization runs on a private dialect. "On time" means the customer-request date in one division and the last-committed date in another. "Fill rate" is measured in units here, lines there, orders somewhere else. Humans absorb these inconsistencies through years of hallway calibration; that's what "experience at this company" partly is.

Now put an agent in the middle. An AI agent negotiating between your demand plan and your supplier commitments doesn't get years of hallway calibration. It gets whatever definitions it can find — and if those definitions conflict, it will do something worse than fail: it will proceed, confidently, on one of them.

This is the lesson of the last few years of enterprise AI. The failure mode isn't usually the model being weak; it's the model being **ungrounded** — fluent about entities that mean different things in different systems. The remedy isn't more intelligence. It's shared, versioned, externally-stewarded definitions the model can be *held to*. An ontology.

SCOR is that ontology for supply chains, and it comes with three properties an internal glossary never has: it's **industry-neutral**, so it survives your reorgs and your acquisitions; it's **complete enough**, covering process, metric, practice, and skill; and it's **not yours**, which means no internal faction can quietly redefine a metric to look better in the Monday deck.

The metrics hierarchy is a contract

The part of SCOR we lean on hardest at A2go is the part most people skim: the metric hierarchy. Perfect customer order fulfillment is a Level-1 reliability metric; it decomposes into defined Level-2 components — delivered in full, delivered on time to commit, documentation accurate, condition perfect — and SCOR treats each level as the diagnostics of the level above. For a metric like this, the descent is arithmetic: you can compute your way from the headline number to the transactions beneath it with no interpretive step.

Once agents participate in decisions, that structure does two jobs.

First, **it separates what may be generative from what must be deterministic.** An agent's reasoning about a trade-off — protect this contract, re-promise that spot order — is judgment, and generative models are genuinely good at it when grounded. But the numbers the judgment rests on must be computed, not composed. When a CFO asks why penalty exposure is $11,000 and not $56,667, "the model estimated it" is not an answer. "It is the defined SCOR reliability decomposition over these six orders, computed from these ERP rows" is. Our architecture pairs non-deterministic reasoning with deterministic compute for exactly this reason; SCOR's hierarchy is what makes the deterministic half *specifiable*.

Second, **the hierarchy is the explanation path.** When an agent recommends an allocation and a human asks why, the answer should descend the same tree an analyst would: attribute, to metric, to component, to transaction. Explainability isn't a feature you bolt onto a model; it's a property of having agreed, in advance, what the numbers mean and how they roll up.

SCOR DS is quietly agent-shaped

Here is what surprised us on re-reading the Digital Standard with 2026 eyes: the revision reads as if it anticipated agents, though it predates the current generation of them.

The old SCOR drew a linear thread — source, then make, then deliver, with Plan sitting above it — which matched how monolithic planning systems batch-processed the world. SCOR DS draws the supply chain as a **continuous, connected network, always in motion**. Read that

picture with agent eyes and it describes concurrent, asynchronous work: an order-planning agent re-allocating while a sourcing agent re-routes inbound while a logistics agent re-cuts promises, all against the same disruption, none waiting for a meeting. That is not how a monthly S&OP cycle behaves. It is exactly how a set of agents behaves.

And the DS made **Orchestrate** a process in its own right — the Level-0 layer where strategy, business rules, risk posture, and performance management connect the others. Read that with agents in mind and it describes the coordination layer every multi-agent system needs and few actually specify: who sets the constraint envelope, who sequences the loops, where the budget cap lives, what gets archived and why. When we say ADIP's agents are "coordinated through a Judgment Layer," we are describing an implementation of what SCOR DS calls Orchestrate — one operating under more constraints than cost and speed, now that environmental and social sit among the performance attributes.

What changed, and what didn't

When we practiced SCOR at PRTM, a diagnostic worked like this: weeks of data gathering, a metrics baseline assembled by hand from extracts, a benchmark comparison, a gap analysis, and a roadmap — delivered as a document, decaying from the day it was printed. The model was never the bottleneck. Data access was. Compute was. The refresh cycle was.

All three bottlenecks are now engineering problems rather than facts of life. Systems are queryable in place; the metrics baseline that took weeks is a view that refreshes continuously; the "diagnostic" can run every morning against last night's transactions. What the agentic era actually changes is the **cadence** at which SCOR can be applied — from an engagement every few years to a standing instrument.

What didn't change is the questions. Which commitments are at risk? Where is the trade-off between serving this customer and protecting that margin? What is the cheapest reliable recovery? We asked those questions with SCOR in consulting rooms decades ago. Our agents ask them now — with the same vocabulary, against the same metric definitions, at a cadence no engagement team could sustain.

What to do Monday

If you run a supply chain and any of this lands, three moves, in order:

1. 1. **Adopt the vocabulary deliberately.** SCOR DS is open-access. Map your internal terms to its processes and metrics and write the mapping down. Every ambiguity you resolve on paper is one an agent won't resolve for you, silently, later.
2. 2. **Instrument the Level-1 metrics before automating anything.** If perfect customer order fulfillment, customer order fulfillment cycle time, and total supply chain management cost aren't computed deterministically from your systems today, fix that first. Agents amplify whatever measurement discipline they find — including its absence.

3. 3. **Pick one decision loop, not one department.** A single recurring decision — an allocation under shortage, a re-plan trigger, a disposition call — instrumented end to end with its rationale archived, teaches you more about agentic operations than any platform evaluation.

The framework was never the point; the shared understanding was. For thirty years SCOR let people who had never met walk into the same room and argue productively about the same supply chain. The agentic era doesn't retire that idea. It just adds new participants to the room — and they, more than anyone, need the language.

SCOR is developed and stewarded by ASCM; the SCOR Digital Standard is available open-access at ascm.org. A2go's founders contributed to SCOR as management-consultant practitioners at PRTM.