WHITE PAPER

Expert systems *didn't go away.*

An old idea with relevance. Before the current wave, we captured how a **thirty-year veteran** made a judgement call and ran it at scale. That pattern is still the **highest-return AI** most operations can deploy.

Riverton & Associates, Inc. · 19 May 2026

Executive summary

Around twenty years ago we built an expert system with IBM for AK Steel. It encoded the judgement of a thirty-year veteran operator — decisions that had never been written down — and applied them on every shift rather than on the shifts he happened to work. It saved roughly $10 million a year.

It is still running. It is still saving money.

That fact deserves more attention than it usually gets. In an industry where a production model is doing well to survive eighteen months before drift forces a retrain, a system built two decades ago is still making the same decisions correctly. This paper argues that the pattern behind it — elicit what a skilled practitioner knows, encode it, run it everywhere — remains the highest-return application of AI available to most operations, and explains where it wins, where it loses, and why the historic barrier to building one has largely fallen away.

What the pattern actually is

An expert system captures how a person makes a decision and executes that decision on every case, consistently, without fatigue.

It is worth being precise about the distinction from machine learning, because the two get confused. A learned model infers rules from labelled examples. An expert system takes the rules from a person who already has them. There is no training set, no labelling exercise, and no requirement that the past contains enough examples of the situation you care about.

That difference matters most where it is least convenient: rare events, new processes, and decisions that were never logged in a form a model could learn from.

The steel case

A thirty-year veteran operator on a production line accumulates a body of knowledge that is almost entirely undocumented. He knows which combinations of conditions precede a problem. He knows when the specification is a hard limit and when it is a guideline. He knows which readings to trust and which sensor drifts. Most of this he cannot readily articulate — asked why he made a call, the honest answer is often *because that is what it needed*.

> The value of encoding that was not that the system out-thought him. It was that it held the line when he was not there.

An operation does not run at the level of its best operator. It runs at the average of everyone on every shift, including the new hire on nights and the stand-in during holidays. The gap between the best operator's judgement and the median decision made across a year is enormous, and it shows up as scrap, rework, downtime and lost throughput. Closing that gap — not exceeding the expert, just reaching him consistently — was worth about $10 million a year.

Why it lasted twenty years

The longevity is the part that should change how people think about this.

- **The knowledge does not drift.** The system encodes process behaviour and material physics. The relationship between conditions and outcome is the same today as it was then. Contrast this with a model trained on consumer behaviour or fraud patterns, where the world moves underneath the model and yesterday's accuracy is no guarantee of today's.

- **It is inspectable.** Twenty years of engineers have been able to open it, read what it does, and understand why. A rule states its own reasoning. A change can be reviewed before it ships. Nobody has ever had to take its output on trust.
- **It fails loudly.** Given input outside anything anticipated, a rule simply does not fire and the system says so. It does not produce a confident, plausible, wrong answer — the characteristic failure mode of the systems currently receiving all the attention, and a genuinely dangerous property in an operation where someone acts on the output.
- **It has no infrastructure treadmill.** No retraining schedule, no feature store to maintain, no accelerator to budget for, no dependency that goes end-of-life every eighteen months.

Why the pattern was abandoned anyway

It is worth being honest about this rather than nostalgic. Expert systems fell out of use for real reasons, not only fashionable ones.

- **The knowledge acquisition bottleneck.** Getting rules out of an expert's head was slow and expensive — a skilled interviewer, many sessions, and an expert willing to spend weeks on it. Worse, experts genuinely cannot articulate much of what they know, so the elicitation had to infer the rule from watching decisions rather than from asking.
- **Brittleness at the edges.** Systems handled the anticipated cases well and the unanticipated ones not at all.
- **Maintenance load.** Rule sets grow, interact, and eventually nobody is confident about what changing one will do.
- **Over-promise.** The 1980s commercial wave claimed far more than it delivered, the funding collapsed, and *expert system* became a phrase that made you sound twenty years out of date.

Then machine learning arrived, then deep learning, then language models, and each absorbed the attention and the budget. The pattern was not displaced because something better was found for the same problem. It was displaced because the industry's attention moved — and a technique that had stopped being interesting stopped being considered, while the systems already built carried on quietly working.

Where it still beats a learned model

- **No labelled history exists.** A new line, a rare failure mode, a decision nobody recorded. A model cannot learn what the data does not contain; a person can still tell you.
- **The decision has to be explained.** Regulated processes, safety cases, anything a customer or auditor will challenge. "The model scored it 0.83" is not an explanation. A fired rule is.
- **Errors are expensive and asymmetric.** When being confidently wrong costs far more than declining to answer, a system that refuses to fire outside its competence is worth more than one that always produces something.
- **The input is already structured.** Sensor readings, ERP fields, process telemetry. Most of the machinery of modern AI exists to cope with unstructured input; if yours is structured, you are paying for capability you do not need.
- **The expert exists and is available.** Which brings us to the reason this is urgent.

Where it loses

Fairness requires stating the other half. Expert systems are the wrong choice for perception — images, audio, free text — and for high-dimensional pattern recognition where the signal genuinely is a statistical regularity nobody could articulate. They are wrong where the domain really does drift, such as fraud and pricing, and they are wrong when nobody actually knows the rules. If your best practitioner cannot outperform chance, there is nothing to elicit and you need a model that learns.

The failure to avoid is treating this as an ideological choice. It is a question about the problem: does the knowledge exist in a person, and is the domain stable? If yes, encode it. If no, learn it.

The bottleneck has moved

Here is what has actually changed, and why the pattern deserves reconsideration now rather than merely respect.

The historic cost of an expert system was elicitation — the weeks of interviews, the analyst translating them into rules, the slow round trips to validate. That cost has fallen sharply. A language model can conduct the structured interview, work through decision logs and shift notes to propose candidate rules, spot contradictions between what an expert says and what the records show, draft the first implementation, and generate the test cases that prove it.

That suggests a specific architecture, and it is the one we would build today:

- **A language model at the edges** — reading unstructured input, eliciting and maintaining the rules, explaining outcomes in plain language.
- **A deterministic engine at the core** — making the decision, recording which rule fired and on what evidence.

The model handles what it is genuinely good at: language, ambiguity, and interpretation. The decision itself stays in a component that is inspectable, testable, and cannot hallucinate. This gets the elicitation speed of the current wave with the durability and auditability of the old pattern — and it is a far better answer for an operations decision than putting a language model in the decision path and hoping.

Why this is the highest-return AI most operations can deploy

Three reasons, and none of them are technical.

- **The gap is consistency, not capability.** Most operations already contain someone who makes the right call. The loss is not that nobody knows — it is that the knowledge is present on some shifts and absent on others. Capturing the best judgement and applying it everywhere is a large, immediate gain that requires no new science.
- **The expertise is leaving.** The thirty-year veteran is, by definition, near the end of a thirty-year career. When they retire the knowledge goes with them, and no amount of subsequent investment recovers it. There is a window, it is closing in most industries right now, and it does not reopen.
- **The cost profile is unusually favourable.** No labelling programme, no accelerator budget, no retraining schedule. The dominant cost is the expert's time, which is finite for reasons that have nothing to do with the project.

Conclusion

Expert systems did not fail. They went out of fashion, which is a different thing, and the industry stopped distinguishing between the two.

The system we built for AK Steel has been running for two decades and is still saving money — a claim very few modern AI deployments will be able to make in 2045. It did not out-think anyone. It took what one experienced person knew, and made it available on every shift.

That is still the most valuable thing most operations can do with AI. The tools for building it have improved considerably. The people whose judgement is worth capturing are retiring. Both of those facts point the same way.

Riverton & Associates researches, designs, develops and implements production software and AI systems, including the expert system described here.

`Riverton & Associates, Inc. · Austin, Texas · riverton-assoc.com`