

Interoperability Standards for Industrial Asset Models: a Field Guide

H. Havel and I. Ismail

2021 (synthetic sample)

Synthetic sample paper. Written solely as a sample source for the chi-tragupta documentation's worked examples; the standards landscape it sketches is simplified for illustration. Public domain (CC0).

Why standards decide twin projects

Digital-twin pilots rarely die of modelling problems; they die of integration. Every asset on a real site speaks the protocol its vendor shipped a decade ago, and the cost of a twin is dominated by the adapters between those dialects and the twin's own model. Standards are how that cost is paid once instead of per project.

Three layers of agreement

Interoperability requires agreement at three distinct layers, and conflating them is the commonest failure we observed. At the **transport layer**, systems agree how bytes move – a publish-subscribe broker, a request-response service, a file drop. At the **syntax layer**, they agree how a message is structured – a schema language and a serialisation. At the **semantics layer**, they agree what the fields mean – that two systems' "temperature" is the same sensor, unit and sampling discipline. Transport and syntax standards are mature and plentiful; semantic agreement is where projects stall, because it cannot be bought, only negotiated.

The asset-model registries

The practical instrument of semantic agreement is a shared asset-model registry: a versioned catalogue mapping every signal to a definition, a unit and an owner. In our synthetic survey of twelve multi-vendor sites, the eight that maintained a registry integrated a new asset in a median of six days; the four that negotiated semantics per project took seven weeks for the same task. The registry's value came less from the catalogue itself than from the ownership column: a signal with a named owner had someone to answer the questions the schema could not.

Versioning discipline

An asset model is a contract, and contracts change. The sites that survived model evolution treated versions as append-only: a field is never repurposed, only deprecated and replaced, and every message carries the model version it was written against. The one site that allowed in-place field redefinition spent, by its own estimate, a quarter of one engineer's year attributing a single mis-unit incident.

Conclusion

Buy transport and syntax; budget for semantics. A registry with named owners and append-only versioning converts the semantic negotiation from a per-project tax into a one-time investment, and in our sample it was the strongest single predictor of a twin project reaching production.