Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Keeping the reasoning with the geometry: rules, reactions and checks in production automotive CAD

Cornel Stefan Manole

Office@ocmeed.com

OCMEE Design LLC

Research Article

Keywords: parametric CAD, explicit reference modelling, knowledge-based engineering, organisational memory, automotive design, knowledge transfer

Posted Date: September 8th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10796287/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: Competing interest reported. The author is the principal of OCMEE Design LLC, which provides engineering design services in the domain described in this paper. The author has no other competing financial or non-financial interests, and no relationship with this journal in any editorial capacity.

Abstract

Engineering organisations lose the reasoning behind a design faster than they lose the geometry [1, 2]. Knowledge-Activated Design puts the rules inside the released CAD model instead, as checks that watch, rules that assign and reactions that fire, and the model tests every embedded condition on the rebuild that changed the geometry. A candidate position comes back with its own verdict, in the part, in front of whoever opens it. Nobody waits for the next review to find out. This paper reports what that does across four automotive production programmes, on a park distance control sensor network rebuilt once on generic geometry so it can be shown in full, and on a folding armrest built as a contrast. The models rebuilt and the checks reported as authored. When an engineer left mid-programme, the replacement opened the part and found the reasoning. Two boundaries are reported honestly. A number fixed by test outside the model can be checked inside it, never derived there, so that ceiling belongs to the knowledge and not the engineer. And a check reports without deciding: three agreements moved a governing dimension, the check caught it every rebuild, and the engineers who sign released it anyway. An adopting team should read that first, because a red state does not stop work. A five-level device places where the operative copy of a rule is kept, from one person's judgement to a network behind a published contract. The paper claims implementation and a service record.

1 Introduction

1.1 The knowledge-loss problem

Engineering organisations routinely know more than they can retrieve, and the part they cannot retrieve is the reasoning, not the geometry [1, 3]. Why a clearance is twelve millimetres and not ten sits with the engineers who worked the problem and, at best, in a specification somebody must open. When that engineer moves on, the organisation keeps the geometry and loses the argument [4].

However, treating this as a filing problem is why it persists. Work on organisational memory locates knowledge in reservoirs, members, tools, tasks and the networks among them, and finds that transfer succeeds or fails according to which reservoir holds what [1], while a shared representation coordinates specialists only when it carries something each can act on [4, 5, 6]. The recurring failure has a shape: rules stored alongside a model drift from the geometry they govern. The field's current direction is to externalise further, into graphs and ontologies held outside the model entirely [7, 8]. This paper reports what happened, in production automotive work, when the knowledge moved the other way.

1.2 What changes when knowledge changes location

Where knowledge sits conditions who can apply a rule and how often it is applied. A rule held in judgement is applied when someone asks. A rule in a specification beside the model is applied only when a person spends attention, which is rationed. A rule written into the model is applied whenever the model rebuilds. The rule has not changed; its location has, and with it the frequency of its evaluation.

Frequency turns a stored rule into a governing one: a violation introduced on a Tuesday surfaces that day rather than at the next gate, once three weeks of dependent geometry sits on top of it. This shift sits inside a broader change the field has named, in which design work is organised around digital representations that coordinate across functions instead of documents that describe intent [9, 10].

1.3 The industrial setting: park distance control sensor integration

United States rear visibility requirements phased in from 2016 are answered by cameras, which carry the formal compliance case [11]. That rule left ultrasonic park distance control outside its scope; such sensing spread over the same period to essentially every programme on commercial grounds. The decision is therefore recurrent rather than mandated, taken again on every vehicle against limits nobody is free to relax.

No function owns a sensor position outright. Five constrain it: advanced driver assistance systems (ADAS), studio surface, packaging, manufacturing and programme timing. A premium vehicle in this segment carries of the order of twelve such positions (practitioner interpretation), front bumper and rear, so a single revision to a styling surface can unsettle several of them together.

The programme office carries the sharp end of that risk. Before tooling is committed, moving a position costs engineering hours across the five functions; after the fascia tool is cut, the same move becomes a tool modification with a timing consequence attached, which is why late position changes are resisted rather than priced. The monetary magnitude of such a change was not recorded and none is asserted here (practitioner interpretation). A moving position travels through all five functions before it holds, about ten times on a sensing programme, an approximate count not logged at the time (practitioner interpretation) (Fig. 1), and the cost sits in the round trip [12, 13]. Each traversal took two to three weeks (practitioner interpretation), and the governing rules sat in supplier specifications, spreadsheets and individual memory, so every re-decision risked losing the reasoning behind the previous one.

1.4 The gap, the position, and the contributions

Knowledge-Activated Design is the industrial method built against those two problems: writing selected, formalisable rules into the released model as executable relations among model entities, so the model tests them on every rebuild and reports the result with the geometry. The normative source stays in the specification and release authority stays with the engineer who signs. The account is written from inside the practice it examines, by a practitioner of fifteen years in automotive design, and the method is that practitioner's later formalisation of an established capability rather than its origin [14, 15, 16, 17]. The production models cannot be published, so the components reported here are generic reconstructions.

The components of the arrangement are not new, and this paper does not claim them. In-part rule artifacts are supplied by commercial knowledge-ware [14, 15], and publishing a stable named reference set rather than an implicit parent-child chain is explicit reference modelling as already established [18,

19, 20]. What is combined here is those two, applied across models instead of within one. That combination is an increment, and this paper does not attempt to establish it, because no variant was built without the published contract and measured against one built with it.

The gap this paper does address is different, and it is a reporting gap. The literature reviewed in Section 2 establishes that executable engineering knowledge is desirable and that modelling strategy decides whether it survives reuse. What it does not contain is a record, written from inside a production programme, of how such an arrangement behaves once it is running: which rules reach the released model and which stop short of it, what it returns to the engineer on each rebuild, and what it cannot settle however well it is authored. Deployments of this kind are not rare in industry. Published accounts of what they do in service are.

Two cases carry the account, both from automotive interior and exterior work. The first is a park distance control sensor network: deployed on four production programmes, consumed by four functions, and rebuilt once as a non-proprietary reconstruction on generic geometry so that it can be shown in full. The second is a folding centre-console armrest, built as a deliberate contrast at the component level. The first case supplies the service record and the negative finding; the second fixes where the relocation stops.

The paper makes three contributions. First, Section 5.5 reports a negative finding: a governing limit was eroded by three agreements with three functions while the part flagged the breach at every regeneration and release authority accepted it, which bounds what embedding an evaluation can achieve. Second, Section 4.3 reports where the relocation stops, on one built contrast, and states what bounds how far a rule can travel. Third, Section 3 states the arrangement precisely enough for another team to place their own experience against it: four governing properties, three kinds of embedded control, and a three-tier directed acyclic network whose invariant is one-way authority. It is a description at the level of structure, not a build specification, and Section 7 says what it omits.

One claim band governs all three and is held in the abstract, the headings, Section 4 and the conclusion: this paper claims implementation of a stated method and the record of how it behaved in service, at the strength each figure's label allows, and claims no causal improvement over any alternative.

2 Related work

The move this paper reports touches several established literatures, and each is conceded in turn and placed relative to the present work in a single clause.

2.1 Knowledge-based engineering and the direction of externalisation

Knowledge-based engineering set out to make engineering rules executable and to keep them as an organisational asset, not a personal one. The foundational methodology treated knowledge capture,

formalisation and application as a managed lifecycle with named owners and documented sources [14], and it was applied directly to generative model creation in commercial CAD [15]. Later reviews of the field are candid about what has and has not held. The technology works where the product family is stable and the rules are stated, but the acquisition step is expensive, the resulting systems are hard to maintain, and the knowledge is repeatedly reported as drifting away from the geometry it governs [2, 16]. A decade-scale review of the design research literature confirms that the maintenance problem, not the encoding problem, is where deployments fail [21]. The same difficulty appears in adjacent design-automation work, where a maturity assessment of engineer-to-order delivery processes finds that automation succeeds in proportion to how well the underlying product logic has been declared and kept current [22], and where a study of CAD automation in automotive development names limits that are organisational as often as technical [23]. Knowledge-based product and process synthesis work continues to demonstrate what a well-scoped rule base can generate [24], and design-decomposition method establishes why the mapping between what is required and what is designed has to be explicit before any of it can be automated [25]. All of this works, and the present paper begins where it leaves off: it proposes no way to acquire or formalise engineering knowledge, and reports what happens after rules are formalised and the question becomes where their operative copy is kept.

The field's current direction of travel is worth stating plainly, because this paper travels the other way. Multi-modal knowledge graphs are proposed as the container for engineering design knowledge, holding entities, relations and rules outside any single model so that they can be queried, reasoned over and reused across tools [7]. Recent work pairs a knowledge graph with a language model to make machine-learning-driven design explainable [26]. Generative approaches now author parametric features and sizing decisions from dialogue [27], and reinforcement learning explores design spaces against surrogate models at a scale no engineer would attempt by hand [28]. Automated review of engineering drawings is being organised around a perception, cognition and collaboration architecture that sits above the model rather than inside it [29]. Each of these works, and each externalises further. The present work is upstream of them: whatever authors or queries the knowledge, something must decide where the copy that governs the rebuild is kept, the question reported here.

2.2 Parametric CAD strategy and explicit reference modelling

The second body of work is the technical ground the method stands on. Parametric CAD does not automatically produce reusable models; how a model is built decides whether it survives a change. Analysis of modelling strategies for design reusability establishes that reusability is a property of deliberate dependency structure, not of parameterisation as such [18]. A quantitative treatment of model complexity shows that the structural cost of a model is measurable and that perceived complexity tracks it, which matters because every additional published name in a contract is a cost the next programme carries [30]. Empirical work on modelling strategy and the automatic generation of geometry variants demonstrates that the same nominal parametric model succeeds or fails at variant generation according to how its references were declared [31]. Explicit reference modelling states the discipline directly:

features are built against declared, named references instead of picked geometry, so that the reference survives when the geometry behind it is replaced [19], and an improved formulation of the same methodology tightens the reference declaration and reports its effect on rebuild reliability [20]. A roadmap for parametric CAD efficiency in automotive work sets out what an organisation has to standardise before any of this repays [32], and practitioner reference works document the advanced parametric and hybrid modelling capabilities the commercial environment provides [17, 33]. This body of work is correct and the paper depends on it, sitting downstream of it in one respect only: raising explicit reference modelling from the feature to the network, so the published contract becomes the boundary across which a governed verdict travels, not only a geometric reference.

2.3 Coordination, boundary objects and organisational knowledge transfer

The third body of work explains why relocating a rule changes a meeting. Knowledge transfer between organisational units succeeds or fails according to which reservoir holds the knowledge, and moving knowledge into tools and task networks behaves differently from moving it between people [1]. A pragmatic account of knowledge boundaries establishes that a shared object coordinates specialists only when each of them can act on what it carries, and that objects which merely display a common view leave the criteria for acceptance in separate documents and memories [4]. Empirical work on objects of collaboration traces the sequences in which such objects actually span group boundaries in design practice [5], and a study of visual representations in distributed product development finds that the receiving group's reason to attend to the object decides its effect [6]. Process research supplies the other half. Design structure matrix analysis names the coupled loop that a cross-functional position decision creates [12], a current review of change propagation formalises how a declared change reaches dependent objects [13], and set-based concurrent engineering describes the alternative of carrying several checked options forward instead of committing early and reworking [34], a pattern since modelled computationally for sub-structured teams [35]. A survey of tacit knowledge management in product design confirms that the practical instruments remain thin [3]. All of this works as description, and this paper takes the boundary-object diagnosis as given. It reports what executable content inside the shared object does to the coordination loop, and what it fails to do.

2.4 Model-based systems engineering and ontology-borne rules

The fourth body of work is the nearest competing approach, because it addresses the same problem with the opposite move. Model-based systems engineering holds requirements, architecture and their relations in a system model, from which modular architectures can be developed and traced [36]. Ontology formalism with semantic rules is the current statement of that position, supporting traceability management by expressing rules and their relations in a formal external structure rather than in any engineering tool [8]. A systematic review of the mathematical underpinning of the field documents how thin the formal basis still is, which is a candid finding from within [37]. Editorial work in engineering

design has argued at field level that data alone does not carry design intent, and that grounding in knowledge, context and decisions makes a representation useful [9, 10], while methodological work in the same venue distinguishes what a study of an engineering method may and may not claim [38, 39]. This approach works and is not in competition for the same ground: it sits above the geometry, and the present work below it, reporting what happens when the operative copy of a rule sits inside the released CAD model the downstream functions consume rather than in a system model they do not open.

2.5 Method reporting in industrial engineering design

A fifth, smaller literature bears on how an account like this one may be written. Engineering design research has been pressing the distinction between demonstrating that an encoded rule executed as intended and establishing that the rule was the right one to encode, and has argued that a study should state which of these it supports before it reports an outcome [38, 39]. The companion argument is that a claim should be tied to a stated measure and method rather than to a motivation, so that a reader can see what would count as the claim failing [39]. Design automation research supplies the matching caution from the applied side, reporting that automation outcomes in industry are conditioned by product logic, organisational readiness and tooling maturity together, so that a single-site effort figure carries a heavy confound [22, 23]. This work is taken as binding, not as background: every figure in Section 4 carries its evidence class, every interpretive claim is labelled as such, and the paper claims implementation and a service record, not causal improvement.

2.6 What the five strands leave open

The gap is operational. None of these bodies of work reports, from inside a production programme, how far down a real rule stack the relocation of a rule into the released model reaches, which rules refuse to move and why, or what the arrangement does and does not settle once the rules are in place. Cost is not the gap this paper claims and no effort figure is offered anywhere in it. Knowledge-based engineering reports that maintenance is where deployments fail, without reporting what the maintained arrangement then does in service. Parametric CAD method reports what makes a model reusable, without reporting how a network of such models behaves when it is governing releases on a live programme. Coordination research reports what a shared object must carry, without reporting an object built to carry it. Model-based systems engineering reports rules held outside the tool, without reporting the alternative of rules held inside it. This paper reports that alternative, with what it returns on each rebuild and with its stated ceiling. The components exist in industry and the reviews above describe them in aggregate. What is not set down is the boundary: which rules reach the released model, which stop short of it, and what the arrangement fails to do once the rules are inside. The gap is not that the field lacks accounts of executable knowledge, which it has in quantity [15, 24], nor that it lacks accounts of reusable parametric structure [30, 32]; it is that no published account joins the two and states how far the joined arrangement reaches into a working rule set, which rules it cannot reach at all, and how it behaves in service.

3 The method: rules inside the model

3.1 What moves and what does not

Knowledge-Activated Design rests on one move: the rule stops sitting beside the model and starts sitting inside it, so opening the part displays what the part is supposed to do. What that move must not disturb is held apart from it: the normative source, which does not move; the operative copy inside the released model, evaluated on rebuild; and decision authority, which stays with the engineer who signs. Only the site of evaluation moves.

3.2 Four governing properties

A model governs in this sense only when all of the following are present: it must carry Knowledge that is Embedded, made Parametric, and given Intelligence that acts. Absent any one of them, the work has fallen back to ordinary parametric CAD or to scripted automation. They are claimed as jointly necessary and argued by elimination, each removal producing a distinct failure (Table 1), of which the Embedding failure is the one the knowledge-based engineering reviews report most often [2, 21]. No uniqueness or completeness is claimed.

Table 1
The four governing properties and their release questions

Property	What fails without it	Release question
Knowledge	Geometry with nothing to enforce	Can the condition and its source be identified from the model?
Embedded	Rules drift from the geometry	Which parameter, check or rule does the model read?
Parametric	A change stops where it was made	Do dependent objects rebuild through declared references?
Intelligence	Silent regeneration into a bad state	Driven bad, then good, does it report each transition?

3.3 The three control types

The production work was authored in CATIA V5 R21 and the non-proprietary reconstruction in 3DEXPERIENCE R2023x. In this modelling environment the properties are carried by three kinds of embedded control. Checks are read-only conditions that watch geometry and parameter state and warn when violated, the canonical example firing if a bezel intrudes into a detection cone. Rules read parameter values and check results and assign values, colours and visibility on every update. Reactions fire on a value change without a manual rebuild. Three authoring disciplines recur: one condition per check, so a failure names what failed; every rule written total, so no stale state survives; and every control authored against published reference names.

3.4 The network: one direction of authority

A single governed part is not enough, because a rule embedded in one model governs nothing if a consumer can rewrite the input it depends on. The arrangement used here, rather than required in general, is a directed acyclic network of three tiers. An intake tier normalises upstream geometry and publishes a contract of stable named references. A variable tier holds the parameters, the embedded rules and the geometry construction (Fig. 2).

Above the variable tier sits the third. A controlled output tier issues the agreed result to the functions that consume it, and it can carry with that geometry a record of which checks were evaluated on the release and what was decided about each. Separating the issued result from the model that computed it is what allows a consumer to subscribe to a decision without subscribing to the construction behind it. Whether the record travelling with the result is complete enough to serve afterwards as an audit trail is a separate question, and Section 4.2.3 reports one display for which it was not (Fig. 3).

One invariant carries the governance property and is the only part of the architecture the argument depends on: authority flows one way, and a consumer reaching past the contract to raw upstream geometry is detectable on inspection of the binding as a contract violation rather than as an ordinary geometry error, though nothing forces that inspection to happen while the control is being authored (Fig. 4). The intent is that a new programme changes the surfaces behind the contract and not the contract itself, because the published names stay stable. How far that held across the four deployments is not characterised here (see Section 7). This is explicit reference modelling raised from the feature to the network, established practice put to work rather than proposed [18, 19, 20].

3.5 A practitioner reading device for where a rule is kept

The four properties report whether a model governs, not how far its governing reaches: one governed model may be consulted by a single engineer while another settles a decision for four. Each arrangement below is therefore read against five ordered levels, L1 to L5, naming where a rule's operative copy is kept (Table 2). It is a practitioner reading device secondary to the cases, and a reader who rejects it loses nothing from them.

Table 2
Where the operative copy of a rule is kept. Each property is placed independently; the arrangement takes the lowest placement

Level	Where the operative copy is kept	Who applies it, and how often	How far a change travels
L1	In a person's judgement	A person who knows it, when asked	No further than that person
L2	In a document or sheet beside the model	A person, when prompted	Within the part, if at all
L3	Inside the part, as a check or rule	The model, on every rebuild	Within the part, with a verdict
L4	Inside the part, with a published contract	The model and its subscribers, on every rebuild	Across a boundary, to named references
L5	Across a directed acyclic network behind a contract	Every model, on every accepted change	Across functions, with decision evidence

One clarification prevents the device from being applied two ways. The level names where the operative copy is kept, meaning the copy the model evaluates. A limit whose authority is owned outside the model does not by itself pull the placement down to L2, because the copy that runs on every rebuild still sits in the part. What an external limit does is block the rise to L4 and L5, since those levels require the limit itself to travel with the geometry, and it creates a maintenance obligation the model cannot discharge. Taking the lowest placement rather than an average is what makes the reading useful. A model may report pass and fail while the values it tests against are kept by hand in a spreadsheet, high on Intelligence and low on Embedding; an average would score it well, and the lowest placement names the property holding it back.

4 Evidence for implementation and use in service: a production deployment and a built contrast

4.1 What the evidence is, and how each claim is labelled

Both cases named in Section 1.4 are authored components the author built. Neither is a study of someone else's work, and no third party's records are used. Every claim carries one of three evidence classes, and the comparison in Section 4.4 reports the class of each arrangement. **Recorded** is the author's own contemporaneous work record. **Reconstruction** is the author's later reconstruction of the same work done an earlier way, with no organisation's records used. **Practitioner interpretation** is what a participant observed, not what a study measured. Where a paragraph carries no parenthetical label, it takes the class of the section it sits in.

4.2 Case 1: the park distance control sensor network, deployed in production

4.2.1 What was built

The first case is the anchor: deployed in production, consumed by four functions, the only one read as reaching the network level. It holds the supplier detection envelopes, spacing and clearance limits, and bracket construction logic behind one published contract of stable named references, and the network comprises eleven linked models (recorded). Detection envelopes follow the supplier performance specification, which implements the low-speed manoeuvring aid standard [40, 41] and stands as a proxy for sensing behaviour, so the model screens geometry against it and establishes no detection performance.

Effort figures recorded on these deployments are not restated here. What this paper takes from the case is architectural: what was embedded, where it was published, and which functions consumed it. No effort, productivity or time-saving claim is made anywhere in this paper, and Section 2.6 states why cost is not the question asked.

4.2.2 The non-proprietary reconstruction, run live

On a generic sport utility body the network takes a candidate position and returns the consequence: detection cone against the ground planes, clearance behind the fascia, implied bracket, and a pass or warn verdict per supplier condition (Fig. 5). Candidate positions are held as columns in a design table, each a complete state of the skeleton already passing the embedded checks. The table stores states and not verdicts, so a restored column is re-checked instead of trusted, and it works only because the skeleton exists [18, 31]. The rebuild crossed platform generations with the rules re-authored to the same logic, which reports only that nothing depended on a file or history that could not be re-expressed. These are capabilities, not measured outcomes, and this reconstruction is not counted among the four deployments.

4.2.3 What changed in the review, in the record, and at a departure

The library of checked configurations changed the review dynamic (practitioner interpretation). One session is described here. Rows spanning two holder types, three bezel decisions and several weld-face orientations had been prepared, each passing its checks. When the design lead objected to a bezel choice, the engineer switched rows live, Packaging walked the clearance on screen, Programme management priced the alternatives, and the choice was settled against alternatives that had already been checked rather than against alternatives still to be studied: the set-based pattern in practice [34, 35]. The session was not timed, nothing was logged about how it closed, and no comparison of elapsed time against the conventional route is offered here (practitioner interpretation).

The front evaluation rendered the result as a four-zone compliance graph with the detection cones projected onto it, each zone coloured by whether it met the threshold (Fig. 6). That display retains no

history, so it is not an audit trail, and it reports conformity with the encoded condition rather than with the specification behind it.

Midway through a programme, one of the ADAS engineers left. Conventionally that triggers weeks of handover, since which positions were acceptable, and why, sat partly in documents and largely in that engineer's memory. Instead the replacement opened the part and found the checks, the supplier rules embedded in them, the graph of the current state, and the record of what had been chosen and why. One departure was observed and no general claim about turnover is made (practitioner interpretation). Rules that had sat in spreadsheets and in memory were in the part because the work required them there, which is the durable asset the foundational methodology set out to produce [14].

4.2.4 How it was handed over, and where the case sits

The asset was built to be handed over. A user guide is reissued whenever the network is, and every embedded limit carries a named owner, a source revision and a review trigger, without which a model answers an obsolete question precisely [6].

The reading device of Section 3.5 is descriptive vocabulary applied by the author, not an independently assessed instrument, and Section 7 records that no independent placement has been made. Read by the author, all four properties reach L5: the limits are encoded across the linked models, upstream references reach consumers only through the published contract, one accepted change travels the network, and verdicts arrive at four functions with decision evidence.

A second assessor could reasonably place the same network at L3, and the reason is not a weak one. The detection envelope the checks test against is issued by the supplier and maintained outside the part, which is the first of the two conditions holding the armrest at L3 in Section 4.3. On that reading the network's Embedding property is capped for the same reason the armrest's Embedding is, and the L5 placement rests entirely on the contract and network properties. The reading offered here rests on a different emphasis: what L4 and L5 name is whether the verdict travels behind a published contract, and it does, whereas the supplier envelope is a source revision rather than a live calculation. Which emphasis is right has been settled by nobody, no independent scoring of these arrangements has been carried out, and the placement of the anchor case is therefore open.

4.3 Case 2: the multi-layer armrest, a contrast at the component level

The sensor network is read at L5, on the placement offered in Section 4.2.4 and not on the competing one. Either reading leaves open whether that depth belongs to the component class or to the conditions the reading device names. To separate them a folding centre-console armrest was built and rebuilt generically. No hours were recorded and no baseline set, so it carries no effort claim. Holding author, tooling and consuming function constant while varying only the knowledge makes it a contrast, not independent confirmation.

The armrest is a layered assembly driven from a styling top surface, cover over foam over substrate and mesh lamination, so that changing the surface rebuilds every layer, edge treatment and stitch feature. Sixteen named parameters form the declared input boundary, one of them the top-surface source through which styling enters (Fig. 7).

Four of those sixteen parameters set layer thicknesses, and the four stack into one through-thickness dimension. Figure 8 shows that stack drawn against a 50 mm allowance. That allowance is not derived by the model and is not measured here: it is the through-thickness budget assumed on this component when the occupant protection case is argued on pad thickness rather than on trajectory or on energy absorption, and it enters the part as a number set elsewhere (practitioner interpretation). The foam layer is the largest single term in that stack, and the control watching it is the one discussed next.

What the component contributes to this paper is its placement, not the derivation behind it. It is read at L3, held there by two properties: embedding is capped by one of its four controls, the foam compression band, whose governing number is established by physical test outside the model, and intelligence is capped because publishing geometry alone does not reach L4, whose property is a contract carrying the verdict. Only one function consumes the armrest decision. Both caps are conditions of the problem rather than of the author.

The four controls and the reason the fourth cannot be fully embedded are not derived here. This paper uses only the resulting placement, as one row of the comparison in Table 3.

4.4 Cross-case reading

Both arrangements rest on built components: one production deployment across four programmes, and one authored contrast built for the purpose. Nothing else is offered, and no claim in Section 8 rests on anything that was not built or recorded.

The two arrangements and their placements are collected in Table 3.

Table 3
The two arrangements and where each sits

Arrangement	Expressibility	Consumers	Placement	Evidence class
Sensor integration network	High, spatial	Four functions	L5	Recorded
Multi-layer armrest	Three spatial checks, one physical	One function	L3	Reconstruction, no effort claim

Depth has a hard precondition: on rebuild the model must compute each quantity the rule relates, a distance between its own solids or an angle against its own datums. The kind of knowledge that fails that test is physical behaviour established elsewhere, as with the armrest's foam band: the part can store the resulting number but cannot derive it. Such a ceiling belongs to the rule and to the representation on hand for expressing it, not to the person authoring the model. That reading is drawn

here from one built contrast, and a controlled single-component derivation of the same point, holding author, toolset and consuming function constant, would be the way to test it.

Whether relocation repays appears to turn on how many functions consume the decision (practitioner interpretation). With one consumer, a contract governing a decision nobody outside the component needs is overhead, which is why the armrest stops at L3 and loses nothing by it. With several consumers the serial loop dominates, and a network removes part of it by letting one change reach everyone without a handoff. Reuse across programmes is a second term that ought to matter, and this evidence cannot separate it from the first: the sensor network is high on both and the armrest low on both. Neither term is measured, no threshold is claimed, and the reuse term is reported rather than argued.

5 What production use taught

What the deployment and the practice around it taught is set out in Sections 5.1 to 5.6. One of them, placed fifth, is a negative finding, and it bounds the rest.

5.1 The durable unit is larger than the model

A relocation account invites the reading that knowledge leaves people and settles in models. Practice suggests otherwise. The durable unit was the model together with the specification behind its limits, the user guide released with it, named owners, a group able to repair it, and a release process versioning model and guide together. A model becomes organisational memory only when ownership, documentation, maintenance and release authority surround it (practitioner interpretation).

5.2 Redistribution, not transfer

Product rules move from specifications and specialist judgement into controlled executable copies, and routine application moves from specialists to the model, while authoring, debugging, exception handling and ownership stay with people. Embedding moves a rule from the member reservoir into a tool the task network exercises on every rebuild [1], which is the reading offered for the handoff described in Section 4.2.3 (practitioner interpretation).

5.3 Split use from authoring

Adoption separated using a model from authoring one. Most engineers were trained to instantiate a governed network, supply its declared inputs and interpret its checks, while a smaller advanced-modelling group authored, repaired and reissued it. Neither custodial effort nor capability transfer was measured (practitioner interpretation).

5.4 Embedding scales errors as efficiently as good practice

A wrong limit in a document misleads whoever reads it. A wrong limit inside a reusable network is applied on every rebuild with the authority of something the model appears to have checked.

5.5 Negative finding: a reporting part does not hold the decision

This lesson came from an episode in which nothing malfunctioned. Every control fired as written, and the outcome was still wrong by the encoded limit.

Three agreements, struck at different moments with three different functions, each moved the same governing dimension. Each answered a demand unrelated to the others, and the three were negotiated separately rather than as a package, so no combined displacement is claimed here. The check evaluates on every rebuild, so from the first concession onward the part reported the breach on every regeneration and never came back green. Engineers holding release authority looked at the flag and released the geometry. What the finding needs is the sequence and not the sizes, and the sizes are in any case not available. The per-concession magnitudes and the programme context are operational detail of one deployment and are not reported here; the total departure from the limit at release was never recorded, and the count of regenerations across which the flagged state persisted was never logged. Nothing is estimated in their place. What is recorded is the number of concessions, three, and the number of functions party to them, three.

The finding concerns entitlement rather than software. Relocating a rule into the part alters how often a question is asked and how visible its answer is; it does nothing to alter who may overrule that answer. Making a state legible to four functions is not the same as making it binding on any of them [4, 6]. Authority remained with people, deliberately, and a team adopting the method on the assumption that a red state stops work will learn otherwise at its first negotiation. The guardrail adopted is procedural rather than geometric. The dimension was made a controlled parameter, so moving it takes an explicit request instead of passing unremarked inside a larger change, which turns a run of small unnoticed losses into one visible decision. Beyond that, the retained record keeps the two states apart afterwards, so a release meeting its encoded conditions is never confused with one waved through in spite of them, the second carrying a named owner and a stated rationale.

5.6 Adoption stalls on contract discipline, not skill

Authoring skill is not the scarce thing. Teams pick up checks, rules and reactions quickly, and onboarding appeared to run quickly because the knowledge is already in the part, though no onboarding time was measured (practitioner interpretation): a newcomer reads the checks in an existing model, then authors one check, one rule and one reaction against the published contract. What takes longer is treating the published contract as binding when it is faster not to. That is a matter of discipline rather than of modelling ability, and it is where training has to be aimed.

6 Discussion: returning to the literature that framed the gap

The knowledge-based engineering reviews name maintenance and drift as the recurring cause of deployment failure [2, 21]. The arrangement reported here relocates that burden rather than solving it, so

drift becomes a contract violation with a definite place to be caught, and the maintenance obligation of Section 3.5 is where that burden now sits. The parametric CAD literature is confirmed and not extended: the failures observed here are the ones it predicts when references are picked rather than declared [18, 19, 20], and the only addition is that a published contract can carry a verdict as well as a reference, which is what separates a network of reusable models from one that governs.

The review is consistent with what the coordination literature would predict, on one observed session and with no comparison (practitioner interpretation). A model displaying a common view while leaving acceptance criteria elsewhere yields action items, not decisions [4, 5], and a representation has effect only where the receiving group has reason to attend to it [6]. Section 4.2.3 is one instance of what changes when the criteria travel inside the object, and Section 5.5 is the limit case in the same terms: the object carried the criterion, displayed the breach and retained the state, and the organisation decided otherwise. Coupling research explains where this matters, since the cost of the sensor decision sits in the loop rather than in any single judgement [12, 13].

The design-automation literature supplies the caution this account has to accept. Automation outcomes in industry are conditioned by product logic, organisational readiness and tooling maturity together, so a single-site effort figure carries a heavy confound [22, 23], and the decomposition literature is explicit that a mapping between what is required and what is designed must be declared before any of it can be automated [25]. The architectural account in Section 4.2.1 is reported inside that caution rather than against it. Organisational readiness is precisely the term this evidence cannot supply, since one practitioner's record of one setting cannot separate it from the method, and Section 7 states that confound instead of working around it.

Against the model-based systems engineering and ontology position the difference is placement, not correctness. Externalising rules into an ontology gives traceability and cross-tool reasoning an embedded check cannot offer [7, 8], and the formal underpinning of that direction is still being assembled [37]. An embedded copy offers instead that the rule is evaluated by the model downstream functions actually open. Neither settles what Section 5.5 reports, and the distinction between verifying that an encoded rule executed and establishing that it was the right rule to encode is the one design research methodology has been pressing [38].

7 Limits of the evidence

The evidence is a practitioner's record of the practitioner's own deployment. Inspection shows that rules execute inside the model, that changes propagate and that the encoded checks pass; every claim about review behaviour and organisational memory is practitioner interpretation. Physical performance is established nowhere.

Single author, confounded contributions. The outcomes of Section 4.2 have at least three sources this evidence cannot separate: the method itself; the author's own modelling discipline and the decision to

standardise, confounded with the relocation throughout; and a favourable setting, none of whose conditions was varied.

Section 3 **is a structural description, not a build specification.** It states the tiers, the invariant, the three control types and the four governing properties. It does not give a reference-naming convention, a schema for the published contract, an example control as authored, or a mapping of the deployed models onto the tiers. A team could place its own experience against what is given here. A team could not rebuild the arrangement from it, and no claim to the contrary is made.

No benchmark is available. A benchmark compares methods against a shared problem statement, a shared task set and an agreed measure. The problem formulated here, where the operative copy of a governing rule is kept and how far that placement reaches, has no reference task set and no agreed measure against which such an arrangement could be scored. A score against external rule bases, or against model-based systems engineering, would be neither obtainable nor meaningful, and is not the missing piece of this paper. Evidence should instead be proportional to the decisions a method supports and to the consequences of being wrong [38, 39]. Offered in that proportion are four things: one governed network deployed on four production programmes, reported as an architecture and not programme by programme; one non-proprietary reconstruction shown in full; one built component reported as a contrast against the deployment, placed at a lower level and stated as such; and one release that went out against a flagged condition, reported in full. No superiority is claimed and the claim band of Section 1.4 requires none.

The reading device is a proposal, and the anchor placement is contested. Every placement in Table 3 was made by the person who built both the components and the device. Section 4.2.4 sets out a competing reading of the anchor case, at L3 rather than L5, following this paper's own convention, and leaves it standing. A worse difficulty faces any assessor: a placement can be checked only by opening the models, which cannot be published, so the reading rests on the description given here. Independent placement by engineers who did not build the models has not been carried out.

The four deployments are stated, not characterised. The sensor network ran on four production programmes and that is a recorded fact of the deployment, but the programmes are not described and nothing distinguishes one from another here. A reader cannot therefore judge how much of the arrangement carried across unchanged and how much was re-authored each time. Cross-programme reuse is accordingly reported and not demonstrated, and no conclusion in Section 8 rests on it.

Scope. Everything here concerns geometry-centred work organised around parametric CAD, in one industry and largely in one organisation's production context.

8 Conclusion

Engineering knowledge has a location, and that location decides who can apply a rule, how often it is evaluated and what an organisation retains when individuals leave. What this account adds is what

happens next, once the rules are inside and the model is governing releases. The two built arrangements report both halves of the location question. Expressibility bounds how far a rule can move: the armrest's foam band stops short of full embedding because its governing number is fixed outside the model, and the reading offered here is that authoring effort does not move that boundary (practitioner interpretation). Consuming structure bounds whether moving a rule repays: one consumer left the armrest at L3 with nothing lost, while four consuming functions made the network level worth carrying for the sensor case. Reuse across programmes is reported as a fact of the deployment and is not offered as a demonstrated property, because the four programmes are not characterised here and nothing separates reuse from consuming structure in this evidence. The claim band is unchanged at the close: implementation of Knowledge-Activated Design and the record of how it behaved in service, at the evidence strength each labelled figure carries. No causal improvement is claimed, no benchmark applies to a problem formulated this way, and what stands in place of one is set out in Section 7.

The model is not self-sufficient. Specifications remain the normative source, engineers retain decision authority, and a model carries organisational memory only when rules, documentation, ownership and release governance are maintained together. The negative finding states the boundary exactly: three agreements with three functions eroded a governing limit; the part flagged the breach at every regeneration from the first concession onward, and an authorised organisation accepted the deviation. What the organisation gains is not primarily speed. It is that the reasoning behind its geometry stops leaving the building.

Declarations

Competing Interests

The author is the principal of OCMEE Design LLC, which provides engineering design services in the domain described in this paper. The author has no other competing financial or non-financial interests, and no relationship with this journal in any editorial capacity.

Ethics approval.

Not applicable. The work reports engineering practice and model artefacts. No research involving human participants, animals, their data or their biological materials was carried out.

Consent to participate.

Not applicable. No human participants were enrolled.

Consent to publish.

Not applicable. The manuscript contains no individual person's data in any form.

Right to publish.

The industrial material reported here is presented without employer, site, programme, supplier or individual identifiers, in aggregate and descriptive form, and no proprietary file, model or specification is reproduced or made available.

Use of generative AI.

A large language model was used as an editing aid for language, structure and consistency checking. All engineering content, method definitions, case facts, figures and claims originate with the author, who reviewed and edited every output and takes full responsibility for the manuscript.

Funding.

This work received no funding from any public, commercial or not-for-profit body.

Author Contribution

Cornel Stefan Manole conceived the work, defined the method, authored and deployed the models, carried out the analysis, prepared all figures and tables, and wrote the manuscript. Cornel Stefan Manole reviewed and approved the final manuscript.

Acknowledgements.

None.

Data availability.

No dataset underlies the findings of this paper. The production models and supplier specifications described in Section 4.2 are the property of the organisation at which the work was carried out, are not reported here as data, and cannot be shared. The non-proprietary reconstruction described in Section 4.2.2 was built by the author on generic geometry, contains no material belonging to any third party, and can be made available from the author on reasonable request. The findings reported in Sections 4 and 5 are observational and are stated in full in the text, so no separate data file is required to read or check them.

Code availability.

The embedded controls reported here are rule and check artefacts inside CAD parts rather than standalone source code, and no software application or custom code was produced. The structural form of each control is described in the text so that it can be re-authored in any parametric modeller with a rule language.

References

1. Argote L, Ingram P. Knowledge Transfer: A Basis for Competitive Advantage in Firms. Organ Behav Hum Decis Process. 2000;82(1):150–69. https://doi.org/10.1006/obhd.2000.2893.
2. Verhagen WJC, Bermell-Garcia P, van Dijk REC, Curran R. A Critical Review of Knowledge-Based Engineering: An Identification of Research Challenges. Adv Eng Inf. 2012;26(1):5–15. https://doi.org/10.1016/j.aei.2011.06.004.
3. Sun X, Huang R, Jiang Z, Lu J, Yang S. On Tacit Knowledge Management in Product Design: Status, Challenges, and Trends. J Eng Des. 2025;36(10):1673–710. https://doi.org/10.1080/09544828.2023.2301232.
4. Carlile PR. A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development. Organ Sci. 2002;13(4):442–55. https://doi.org/10.1287/orsc.13.4.442.2953.
5. Brubaker ER, Sheppard SD, Hinds PJ, Yang MC. Objects of Collaboration: Roles and Sequences of Objects in Spanning Knowledge Group Boundaries in Design. J Mech Des. 2023;145(3):031404. https://doi.org/10.1115/1.4056798.
6. Wlazlak P, Eriksson Y, Johansson G, Ahlin P. Visual Representations for Communication in Geographically Distributed New Product Development Projects. J Eng Des. 2019;30(8–9):385–403. https://doi.org/10.1080/09544828.2019.1661362.
7. Pan X, Li X, Li Q, Hu Z, Bao J. Evolving to Multi-Modal Knowledge Graphs for Engineering Design: State-of-the-Art and Future Challenges. J Eng Des. 2025;36(7–9):1156–95. https://doi.org/10.1080/09544828.2023.2301230.
8. Dong M, Wang G, Lu J, Wu S, Ma J. Ontology Formalism and Semantic Rules Supporting Traceability Management in Model-Based Systems Engineering. J Eng Des. 2026;37(4):1255–80. https://doi.org/10.1080/09544828.2025.2532291.
9. Jiao R, Luo J, Malmqvist J, Summers J. New Design: Opportunities for Engineering Design in an Era of Digital Transformation. J Eng Des. 2022;33(10):685–90. https://doi.org/10.1080/09544828.2022.2147270.
10. Jiao R. From 'Data-Driven' to 'Data-Informed' Design: Grounding AI for Design in Knowledge, Context and Decisions. J Eng Des. 2026;37(5):1753–75. https://doi.org/10.1080/09544828.2026.2629760.
11. National Highway Traffic Safety Administration. Fed Regist. 2014;79(66):19177–250. https://www.federalregister.gov/documents/2014/04/07/2014-07469/federal-motor-vehicle-safety-standards-rear-visibility. Federal Motor Vehicle Safety Standards; Rear Visibility. Final Rule, 49 CFR Part 571.

12. Browning TR. Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. IEEE Trans Eng Manage. 2001;48(3):292–306. https://doi.org/10.1109/17.946528.
13. Brahma A, Wynn DC. Concepts of Change Propagation Analysis in Engineering Design. Res Eng Des. 2023;34(1):117–51. https://doi.org/10.1007/s00163-022-00395-y.
14. Stokes M, editor. Managing Engineering Knowledge: MOKA, Methodology for Knowledge Based Engineering Applications. London: Professional Engineering Publishing; 2001. ISBN 978-1-86058-295-0.
15. Skarka W. Application of MOKA Methodology in Generative Model Creation Using CATIA. Eng Appl Artif Intell. 2007;20(5):677–90. https://doi.org/10.1016/j.engappai.2006.11.019.
16. La Rocca G. Knowledge Based Engineering: Between AI and CAD. Review of a Language Based Technology to Support Engineering Design. Adv Eng Inf. 2012;26(2):159–79. https://doi.org/10.1016/j.aei.2012.02.002
17. Ghionea IG, Tarba CI, Cukovic S. CATIA v5: Advanced Parametric and Hybrid 3D Design. Boca Raton, FL: CRC; 2022. https://doi.org/10.1201/9781003281153.
18. Camba JD, Contero M, Company P, Parametric CAD, Modeling. An Analysis of Strategies for Design Reusability. Comput-Aided Des. 2016;74:18–31. https://doi.org/10.1016/j.cad.2016.01.003.
19. Bodein Y, Rose B, Caillaud E. Explicit Reference Modeling Methodology in Parametric CAD System. Comput Ind. 2014;65(1):136–47. https://doi.org/10.1016/j.compind.2013.08.004.
20. Aranburu A, Camba JD, Justel D, Contero M. An Improved Explicit Reference Modeling Methodology for Parametric Design. Comput-Aided Des. 2023;161:103541. https://doi.org/10.1016/j.cad.2023.103541.
21. Kugler P, Dworschak F, Schleich B, Wartzack S. The Evolution of Knowledge-Based Engineering from a Design Research Perspective: Literature Review 2012–2021. Adv Eng Inf. 2023;55:101892. https://doi.org/10.1016/j.aei.2023.101892.
22. Willner O, Gosling J, Schonsleben P. Establishing a Maturity Model for Design Automation in Sales-Delivery Processes of ETO Products. Comput Ind. 2016;82:57–68. https://doi.org/10.1016/j.compind.2016.05.003.
23. Kreis A, Hirz M, Rossbacher P. CAD-Automation in Automotive Development: Potentials, Limits and Challenges. Comput-Aided Des Appl. 2021;18(4):849–63. https://doi.org/10.14733/cadaps.2021.849-863.
24. Schaechtl P, Goetz S, Schleich B, Wartzack S. Knowledge-Based Product and Process Design Synthesis of Additively Manufactured Non-Assembly Mechanisms. J Eng Des. 2024;35(7):818–48. https://doi.org/10.1080/09544828.2024.2339162.
25. Albano LD, Suh NP. Axiomatic Design and Concurrent Engineering. Comput-Aided Des. 1994;26(7):499–504. https://doi.org/10.1016/0010-4485(94)90081-7.
26. Hu X, Liu A, Dai Y. Combining ChatGPT and Knowledge Graph for Explainable Machine Learning-Driven Design: A Case Study. J Eng Des. 2025;36(7–9):1479–501.

https://doi.org/10.1080/09544828.2024.2355758.

27. Rosnitschek T, Walschewski J, Grohmann P, Eckardt S, Stonis M, Alber-Laukant B, et al. From Dialogue to Design: GenAI-Based Automation of Parametric Modeling and Sizing Tasks in CAD Workflows. Comput-Aided Des. 2026;196:104065. https://doi.org/10.1016/j.cad.2026.104065.
28. Li H, Wang R, Wang Z, Li G, Wang G, Yan Y. Leveraging Deep Reinforcement Learning for Design Space Exploration with Multi-Fidelity Surrogate Model. J Eng Des. 2025;36(5–6):880–919. https://doi.org/10.1080/09544828.2024.2366686.
29. Chen X, Cheng K. AI-Assisted Auditing Review of Mechanical Engineering Drawings: A Perception-Cognition-Collaboration Reference Architecture and Its Implementation Perspectives. Int J Comput Integr Manuf. 2026. Advance online publication, 1–22. https://doi.org/10.1080/0951192X.2026.2714117
30. Contero M, Perez-Lopez D, Company P, Camba JD. A Quantitative Analysis of Parametric CAD Model Complexity and Its Relationship to Perceived Modeling Complexity. Adv Eng Inf. 2023;56:101970. https://doi.org/10.1016/j.aei.2023.101970.
31. Aranburu A, Cotillas J, Justel D, Contero M, Camba JD. How Does the Modeling Strategy Influence Design Optimization and the Automatic Generation of Parametric Geometry Variations? Comput-Aided Des. 2022;151:103364. https://doi.org/10.1016/j.cad.2022.103364.
32. Bodein Y, Rose B, Caillaud E. A Roadmap for Parametric CAD Efficiency in the Automotive Industry. Comput-Aided Des. 2013;45(10):1198–214. https://doi.org/10.1016/j.cad.2013.05.006.
33. Ghionea IG. CATIA v5: Practical Studies Using Finite Element Analysis. Boca Raton, FL: CRC; 2024. https://doi.org/10.1201/9781003426813
34. Sobek DK, Ward II, Liker AC. Toyota's Principles of Set-Based Concurrent Engineering. MIT Sloan Manage Rev. 1999;40(2):67–83. https://sloanreview.mit.edu/article/toyotas-principles-of-setbased-concurrent-engineering/.
35. Rismiller S, Cagan J, McComb C. Understanding Collaboration in Sub-Structured Teams through a Computational Model of Set-Based Concurrent Engineering. J Eng Des. 2024;35(5):543–69. https://doi.org/10.1080/09544828.2024.2324402.
36. Stirgwolt BW, Mazzuchi TA, Sarkani SA, Model-Based. Systems Engineering Approach for Developing Modular System Architectures. J Eng Des. 2022;33(2):95–119. https://doi.org/10.1080/09544828.2021.1980203.
37. Wach P, Topcu TG, Jung S, Sandman B, Kulkarni AU, Salado A. A Systematic Literature Review on the Mathematical Underpinning of Model-Based. Syst Eng Syst Eng. 2025;28(1):134–53. https://doi.org/10.1002/sys.21781.
38. Jiao R. Demystifying Verification and Validation in Design Research Methodology: From Underdeveloped Practice to Methodological Rigour. J Eng Des. 2026;37(4):1143–62. https://doi.org/10.1080/09544828.2026.2624355.
39. Jiao R. Towards Rigorous Problem Formulation for Engineering Design Research: From Motivations to Measurable Claims via Metric-Measure-Method. J Eng Des. 2026;37(6):2145–71.

https://doi.org/10.1080/09544828.2026.2633289.

40. International Organization for Standardization. Transport Information and Control Systems: Manoeuvring Aids for Low Speed Operation (MALSO): Performance Requirements and Test Procedures. ISO 17386:2010. Geneva: ISO; 2010. https://www.iso.org/standard/51448.html.
41. International Organization for Standardization. Intelligent Transport Systems: Manoeuvring Aids for Low-Speed Operation (MALSO): Performance Requirements and Test Procedures. ISO 17386:2023. Geneva: ISO; 2023. https://www.iso.org/standard/82952.html.

Figures

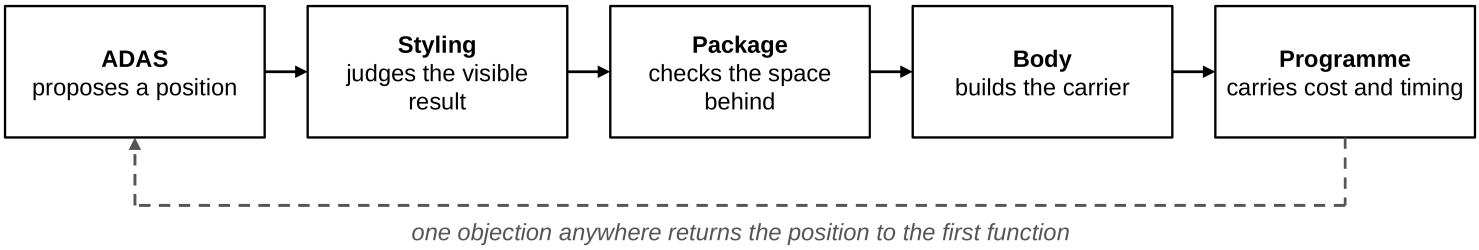


Figure 1

The cross-functional position loop, in which an objection at any of five functions returns the position to the first function. Every traversal is a full pass across five functions before the position holds

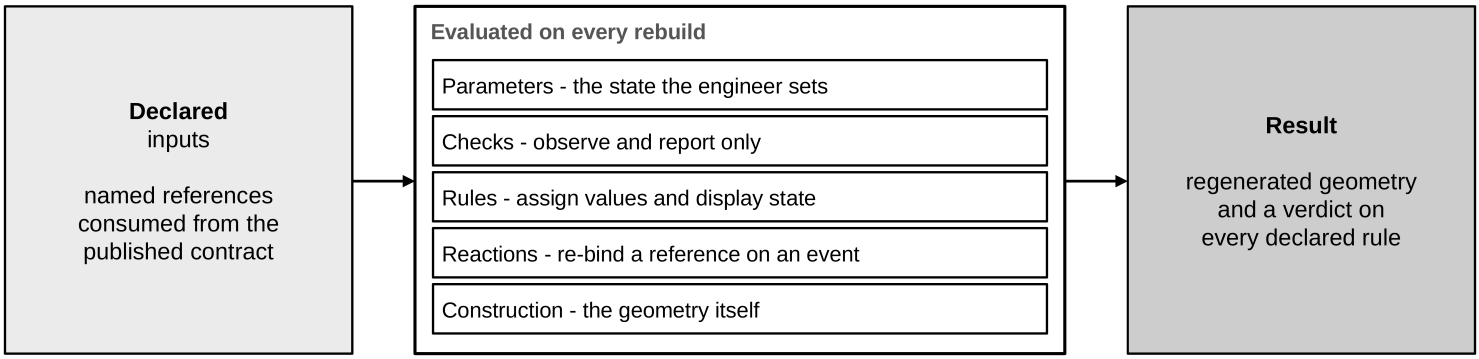


Figure 2

Inside one variable-tier model: declared inputs consumed from the published contract, the controls evaluated on every rebuild, and the regenerated result. The tier holds the engineering, nothing above it evaluates and nothing below it decides

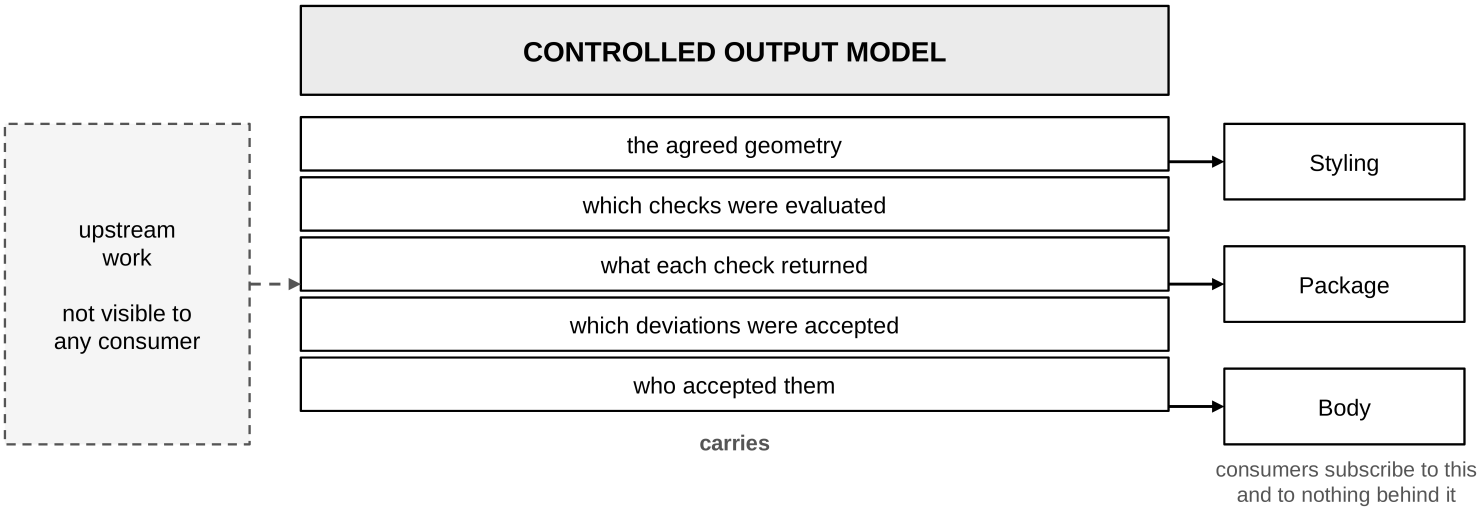


Figure 3

A controlled output model, carrying the agreed geometry together with the record of which checks ran and what was decided, so that a consumer receives a decision with its evidence attached rather than a folder of geometry to interpret

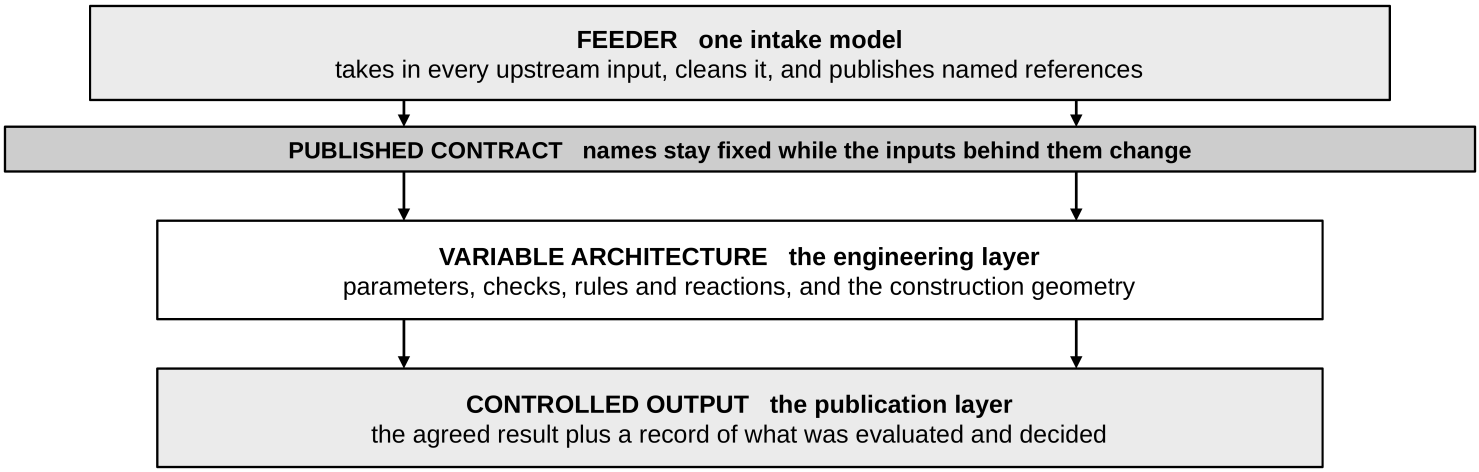


Figure 4

One-way authority across the three tiers, with the published contract separating the intake model from the engineering layer. No edge runs upward, so downstream work cannot alter what the tier above it published, which is what lets a layer be replaced without renegotiating the layer below

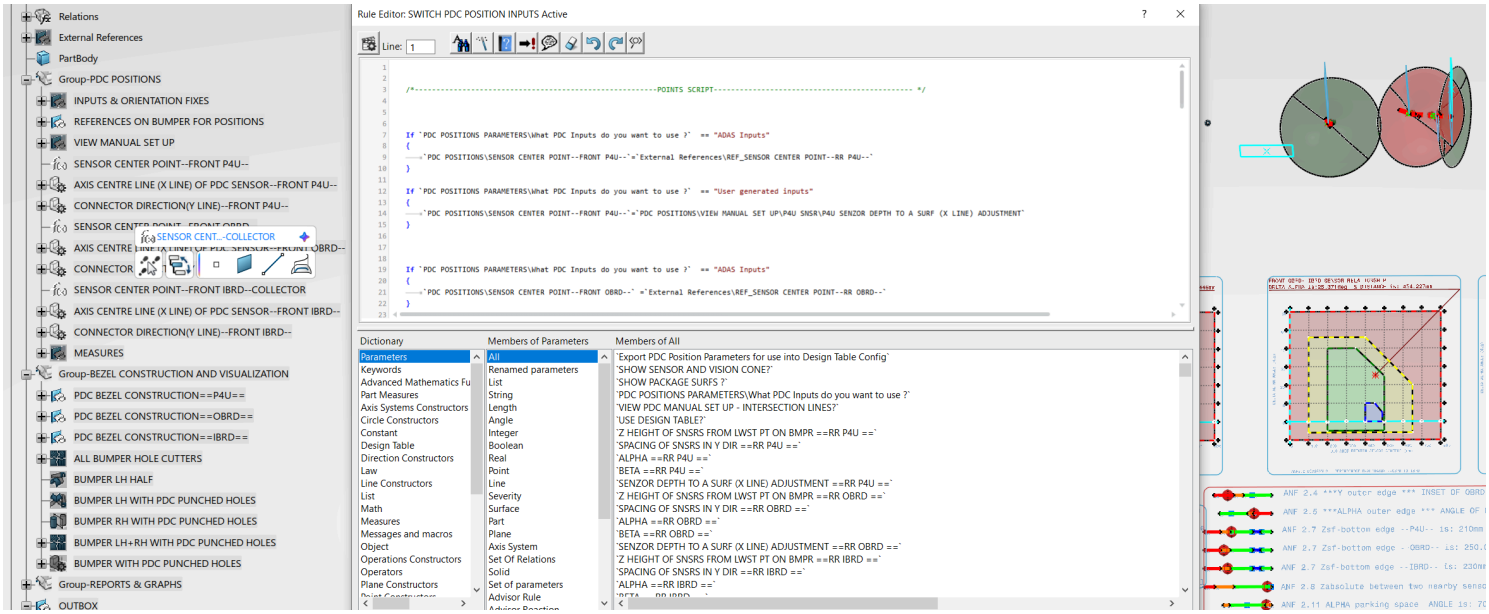


Figure 5

A candidate sensor position evaluated in the authoring environment, with the position parameters and the rule editor open

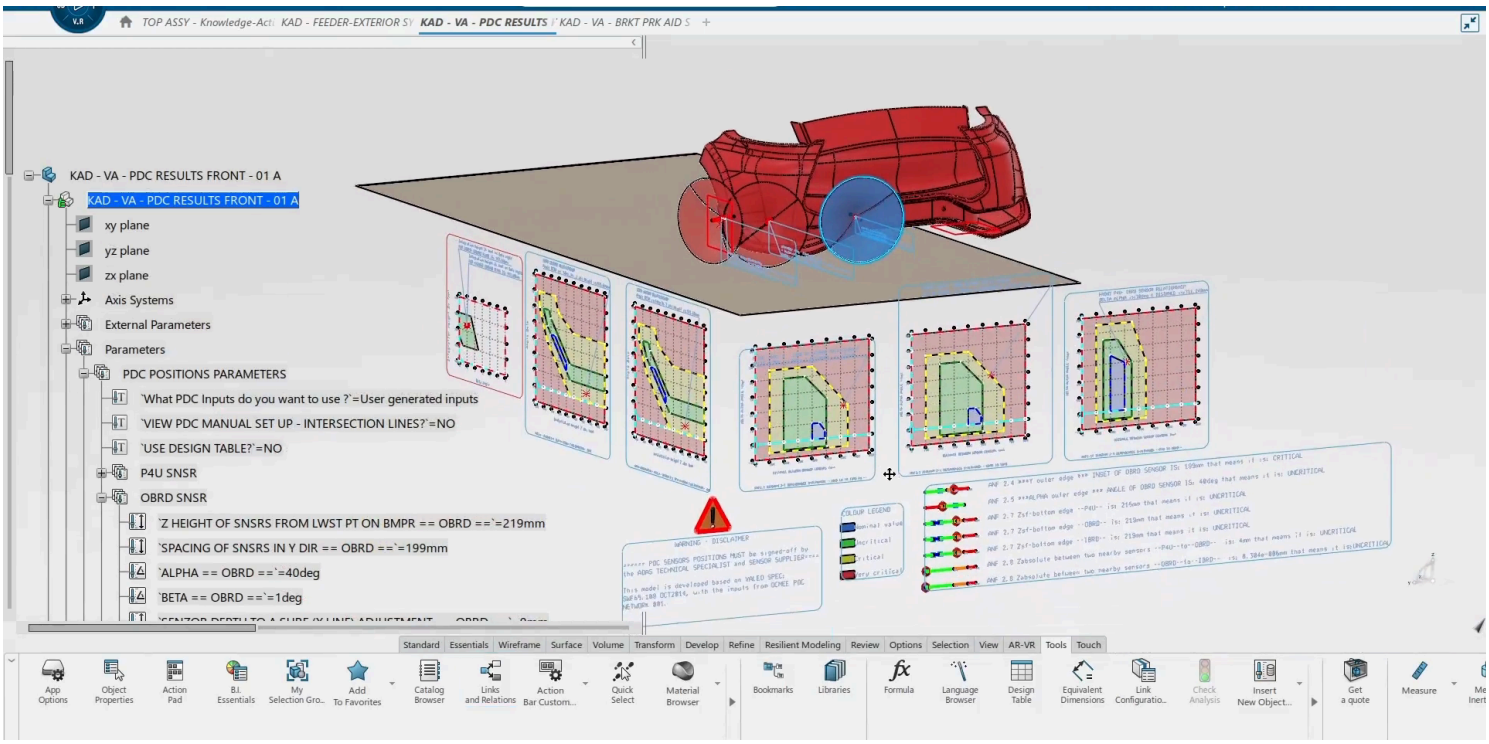


Figure 6

The four-zone compliance graph in a failed state, with the position parameters that drive it shown in the specification tree

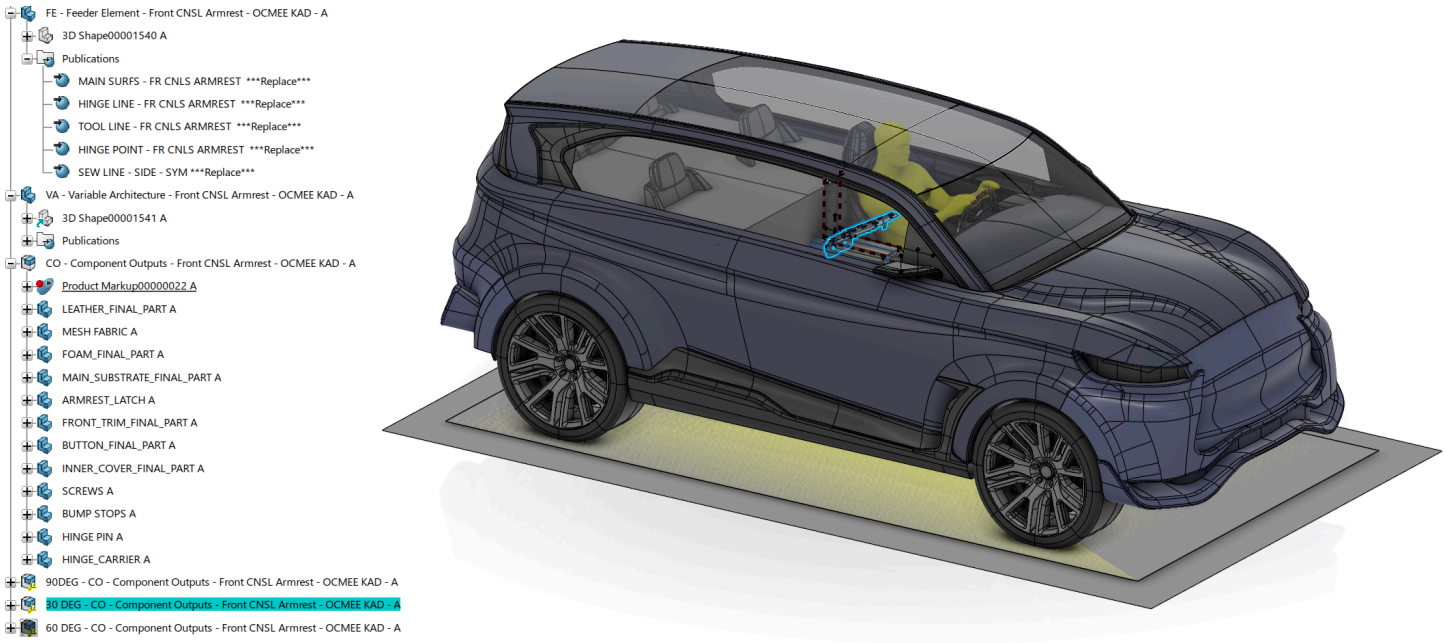


Figure 7

The armrest specification tree, showing the component held as a governed model with its declared inputs, variable-tier architecture and controlled outputs named, positioned in a generic vehicle package for context

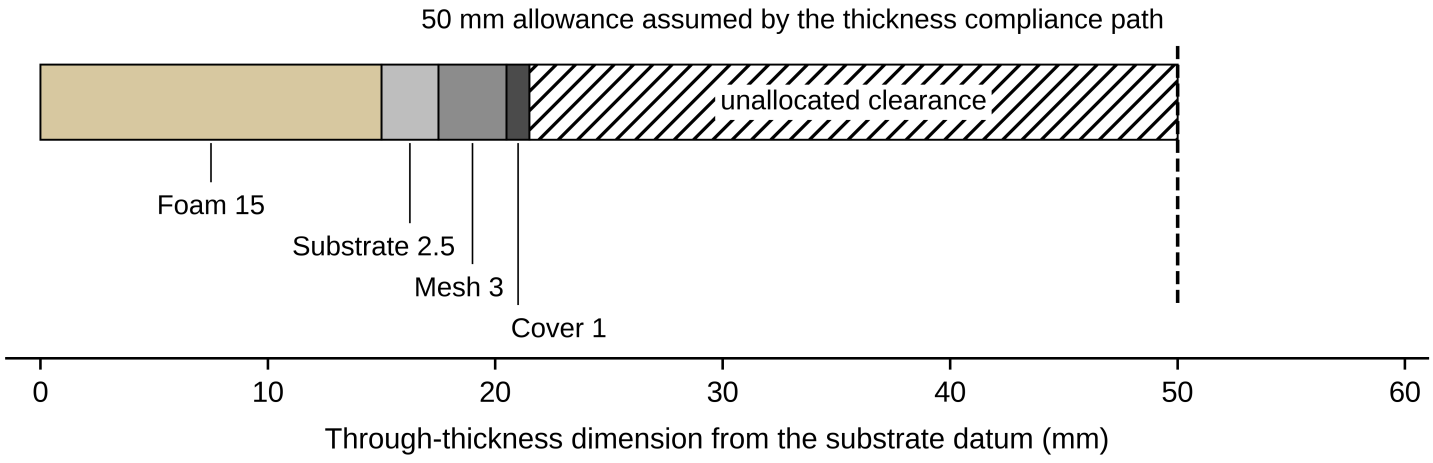


Figure 8

The armrest through-thickness allowance, showing the layer boundaries set by the declared parameters against the 50 mm allowance assumed by the thickness compliance path