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.

Impact of CATIA-Based Parametric Design on Variant Management in Bus Interiors

Ismail Akar

ismail.akar@man.eu

MAN Türkiye A.Ş. https://orcid.org/0009-0001-4423-6627

Research Article

Keywords: Parametric design, Variant management, Bus interiors, CATIA, Knowledge-based engineering (KBE), Digital mock-up (DMU), Product Lifecycle Management (PLM)

Posted Date: March 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9092796/v1

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

Additional Declarations: The authors declare no competing interests.

Impact of CATIA-Based Parametric Design on Variant Management in Bus Interiors

Abstract

This paper presents a CATIA-based approach for managing seat layout (2+2 → 2+1) and overhead service-set (Basic → Premium) transitions under a single parametric architecture. The architecture comprises two skeletons (SeatLayout_Skeleton and ServiceSet_Skeleton), a standardized Publications interface, Excel Design Table-driven variants (pitches 750/720/950 mm; rows 10/12/8), Knowledgeware rules (min-max checks, alignment metric AlignmentDeviation_mm, user warnings), and UDF/PowerCopy attachment templates. In the case study, changes in seat pitch and row count automatically synchronize the service-set modules, while DMU validation reports alignment and clash results. Across three representative variants (750/10, 720/12, and 950/8), the proposed workflow reduced engineering update time by 80% compared with the conventional method. DMU analysis detected zero Seat–Seat and Seat–Service clashes, while alignment deviations remained within the predefined engineering targets, with average values of 0.4–0.6 mm and maximum values of 1.2–1.5 mm. The framework is reusable, scalable, and auditable. In addition, the study contributes to Knowledge-Based Engineering (KBE) research by proposing a dual-skeleton interface architecture that improves model stability and variant synchronization in complex CAD assemblies.

Keywords: Parametric design, Variant management, Bus interiors, CATIA, Knowledge-based engineering (KBE), Digital mock-up (DMU), Product Lifecycle Management (PLM)

Article Highlights:

• One shared model updates seat and service layouts together.
• Design changes are applied in one step instead of remodelling.
• The studied variants reduced engineering update time by 80%.
• Early digital checks removed detected clashes in the tested variants.
• The workflow supports repeatable management of bus interior options.

1. Introduction

Motivation. Bus interior projects entail high variant complexity due to different seating arrangements (2+2, 2+1), seat pitch options (750/720/950 mm, etc.), and overhead service set packages (Basic, Premium). In conventional practice, these variants are managed through separate files, copy-paste geometry, and local dimensional adjustments. This approach introduces operational risks such as:

- Re-modeling and redundant data generation,
- Delays in ECR/ECN cycles and traceability issues,
- High clash counts and alignment errors in early-stage DMU runs,
- The need for manual synchronization of the service set when the pitch or row count changes [1–3].

The result is prolonged design times, increased costs, and quality fluctuations.

Problem Statement. The seat layout (pitch, row count) and the overhead service set (module count, position, edge alignment) must change synchronously. However, interfaces are often embedded within the model and rely on unpublished dimensions. Furthermore, although design rules (min-max limits, regulations, access clearances) are documented, they are not actively enforceable within the CAD environment. For a scalable solution, a single parameter change must automatically propagate across all subsystems, and the results must be quantifiably validated via DMU.

Objective. This study aims to manage seat-layout transitions from 2+2 to 2+1 and service-set transitions from Basic to Premium within a single parametric architecture. It focuses on the automatic and synchronous updating of service-set modules when seat pitch or row count changes, together with early validation of alignment and clash results through DMU.

Approach Summary. The proposed method integrates the following building blocks**:**

1. **Dual skeleton architecture:** SeatLayout_Skeleton and ServiceSet_Skeleton, establishing a strict interface contract via Publications.
2. **Excel Design Table:** Driving variants such as 750/720/950 mm pitches corresponding to 10/12/8 rows.

3. **Knowledgeware rule set:** Enforcing min-max constraints and row-pitch relations, calculating the AlignmentDeviation_mm metric, and generating user Reaction warnings.
4. **UDF/PowerCopy:** Enabling the rapid attachment and repetitive instantiation of seat and service modules.
5. **DMU validation:** Measuring the alignment between the seat cushion front edge and the service set module edge, alongside reporting clash statistics.

Research Questions.

This study is guided by the following research questions:

• **(RQ1)** Can a single parametric architecture manage seat-layout and service-set variants synchronously without re-modeling?

• **(RQ2)** Can this architecture measurably reduce engineering update time and early-stage DMU clashes?

• **(RQ3)** Do Publications, Design Tables, and Knowledgeware together provide an auditable interface structure for PLM/ECR–ECN processes?

Literature Gap

Despite the extensive use of parametric design and Knowledge-Based Engineering in CAD environments, most existing studies focus on automating individual subsystems or specific design tasks. Relatively few studies investigate how multiple interior subsystems can be synchronously governed through a shared parametric interface architecture. In particular, the problem of maintaining stable references and synchronized behaviour between seat layouts and overhead service systems in option-intensive vehicle interiors has received limited attention in the literature. This study addresses this gap by proposing a dual-skeleton parametric architecture that coordinates variant management and interface stability within a unified CATIA framework.

Contributions. The primary contributions of this study are:

- **(C1)** A holistic parametric architecture and application recipe for seats and service sets.
- **(C2)** Design Table-driven variant management combined with a Publications-based interface standard.

- **(C3)** Rule enforcement via Knowledgeware and the formulation of the AlignmentDeviation_mm metric.
- **(C4)** Reusable instantiation templates utilizing UDF/PowerCopy packaging.
- **(C5)** DMU metrics and reporting templates (e.g., alignment tolerance bands, mini-reports).

2. Related Work

Parametric design and Knowledge-Based Engineering (KBE) provide mature mechanisms for governing variants through rules, reusable templates, and automated model generation. Verhagen et al. [5] and La Rocca [6] review the role of KBE across the engineering lifecycle and its integration with CAD-based design processes. At a broader design-method level, systematic engineering design principles [7, 9] and product-platform thinking [8] support the definition of stable interfaces and reusable architectures across multiple product variants.

In CAD/PLM practice, these principles are operationalized in CATIA through Knowledgeware functions, Design Tables, template mechanisms such as UDF/PowerCopy, and DMU-based verification workflows [1–4, 14]. Nevertheless, much of the available practice-oriented literature and documentation focuses on a single subsystem or a single automation mechanism in isolation. Fewer published studies describe how multiple interior subsystems can be synchronized through a shared parametric interface architecture.

In parallel, model-based definition and digital product-definition standards, particularly STEP AP242 and ASME Y14.41, emphasize traceable digital interfaces, model semantics, and data continuity across the product lifecycle [12, 13]. In the broader configuration domain, rule-based product-configuration frameworks also provide an important conceptual basis for representing variant logic explicitly and consistently [10].

Recent studies have further expanded the scope of parametric and knowledge-based design within the context of digital engineering and Industry 4.0. Recent research highlights the integration of parametric modeling, digital twins, and data-driven engineering workflows to support complex variant management in product development. These studies emphasize the role of shared parametric structures, rule-based automation, and model-based definition in enabling scalable design processes and reducing engineering iteration cycles in complex industrial systems.

TABLE 1 – Literature comparison

Table 1. Comparison of related studies on parametric and knowledge-based design approaches.

Study	Method	Application Domain	Main Contribution
Verhagen et al. (2012) [5]	Knowledge-Based Engineering review	General engineering design	Identification of research challenges and opportunities in KBE across the product lifecycle
La Rocca (2012) [6]	Knowledge-based CAD automation	Engineering design systems	Framework for integrating knowledge-based engineering with CAD environments
Meyer and Lehnerd (1997) [8]	Product platform design	Product family development	Platform-based strategy for managing product variants through shared structures
Ullman (2017) [9]	Systematic design methodology	Mechanical design	Structured design methodology for complex engineering products
This study	Dual-skeleton parametric architecture	Bus interior configuration	Synchronous management of seat and service-set variants using CATIA Knowledgeware and shared parameters

As summarized in Table 1, previous studies mainly focus on general KBE frameworks, design methodologies, or conceptual product platform strategies. However, limited research addresses the synchronization of multiple interior subsystems through a shared parametric architecture within a real industrial CAD environment. The architecture proposed in this study extends existing work by demonstrating how dual-skeleton structures, design tables, and Knowledgeware rules can be combined to manage option-intensive interior variants in a consistent and auditable manner.

Our Contribution

Against this background, the present study contributes an application-oriented architecture in which seat rows and overhead service-set modules are governed by the same published parameters, switched through a common Design Table, and validated with explicit DMU alignment and clash

metrics. The emphasis is not only on geometric automation, but also on interface discipline, repeatability, and auditability in an industrial bus-interior context.

3. Method

Methodology overview of the proposed dual-skeleton parametric architecture for synchronized seat and service-set variant management (see Fig. 1).

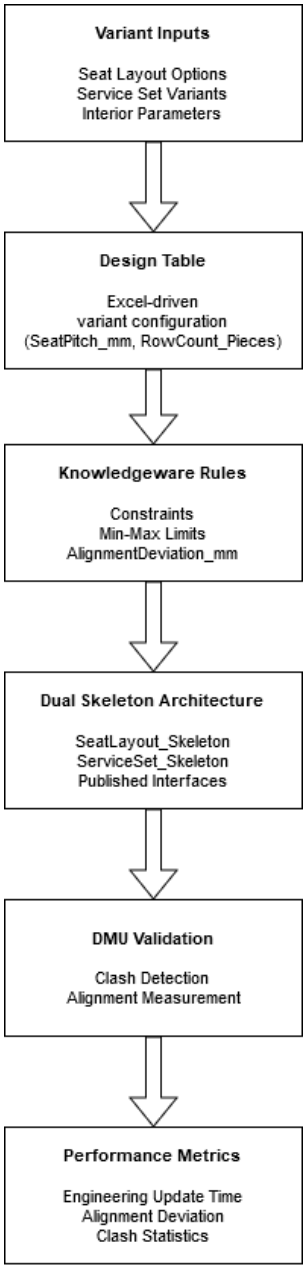


Figure 1

The following methodology integrates the simultaneous driving of the seat layout (SeatLayout_Skeleton) and the overhead service set (ServiceSet_Skeleton) subsystems via a unified skeleton architecture. It combines variant switching through a Design Table and early-stage validation using DMU metrics into a single, cohesive workflow.

3.1. Architecture: Skeleton and Publications

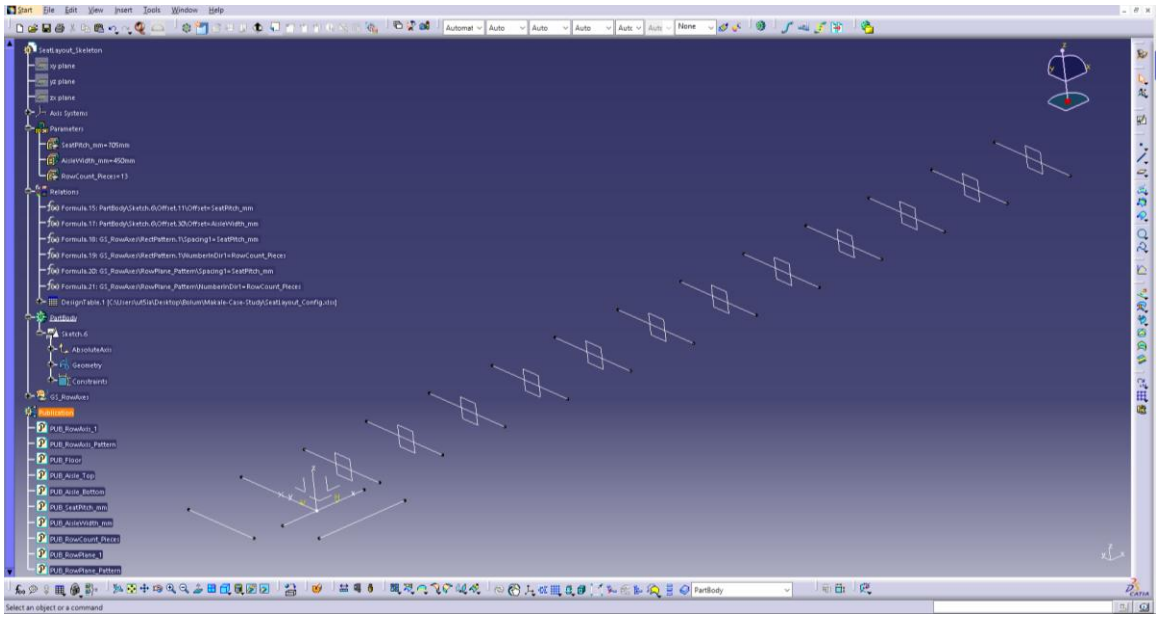


Figure 2. The hierarchical structure of the skeletons showcasing published interfaces and parameters.

Both subsystems are structured within a master product assembly (BusInterior_Master), each possessing its own skeleton CATPart (see Fig. 2).

- **SeatLayout_Skeleton:** This contains the row axes, aisle boundaries (top/bottom), the initial row plane, and essential published interfaces (Publications such as PUB_RowAxis_1, PUB_RowAxis_Pattern, PUB_Floor, PUB_SeatPitch_mm, PUB_RowCount_Pieces, and PUB_RowPlane_Pattern).
- **ServiceSet_Skeleton:** This contains the module front plane and the row-plane template. It imports external parameters from the published parameters of the seat layout (e.g., SeatPitch_mm is linked to ..\SeatLayout_Skeleton.1\PUB_SeatPitch_mm).

The fundamental principle of this architecture is that all sub-assemblies consume *only* published interfaces; direct geometric referencing is strictly prohibited. This approach significantly enhances model stability and ensures complete traceability of modifications.

3.2. Variant Driving via Excel Design Table

Variants are managed and driven through an Excel Design Table. The recommended columns include ConfigID (e.g., Left_10_750, Left_12_720, Left_8_950), SeatPitch_mm (750, 720, or 950 mm), RowCount_Pieces (10, 12, or 8), and optional dimensions such as aisle width, service gaps, and wheelchair bays. Within CATIA, the Design Table is created from this file, mapping the columns to the respective parameters in the SeatLayout_Skeleton and the ServiceSet_Skeleton. During the update workflow, selecting a configuration automatically updates the pattern parameters (e.g., RectPattern.1\Spacing1 = SeatPitch_mm and NumberInDir1 = RowCount_Pieces), ensuring the service set is perfectly synchronized with identical values.

3.3. Knowledgeware Rules (Rule/Check/Reaction)

Key constraints and relations are enforced using the Knowledge Advisor module:

- **Boundary Checks:** Rules are defined for the seat pitch (e.g., SeatPitch_mm between 650 and 950) and row count (e.g., RowCount_Pieces between 8 and 14). An additional regulatory check ensures that the aisle width meets the minimum project or regulatory threshold (e.g., AisleWidth_mm ≥ 400), depending on the applicable vehicle layout and access requirements [11].
- **Relational Rules:** A core rule enforces that the service module spacing equals the seat pitch (ServiceSet_Spacing_mm = SeatPitch_mm).
- **User Warnings:** Reactions are integrated to trigger upon parameter changes. For instance, if the AlignmentDeviation_mm exceeds 5 mm, a message box or log entry is automatically generated.

3.4. Instantiation Templates: UDF and PowerCopy

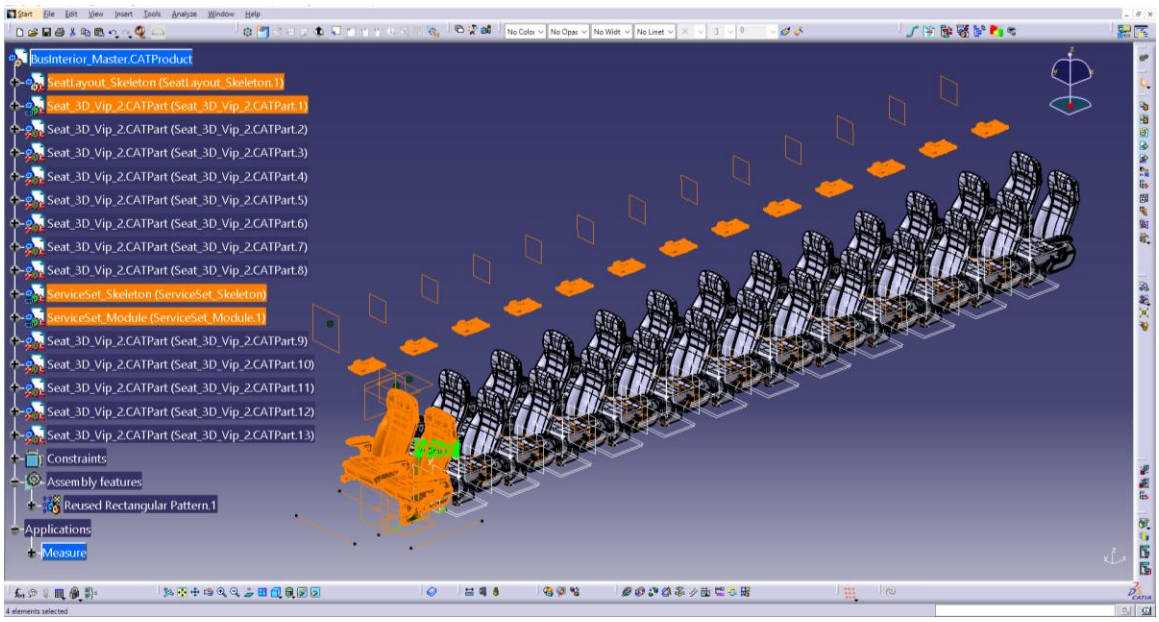


Figure 3. Reusable 3D templates for the seat layout and overhead service module driven by External Parameters.

Repetitive assembly constraints are managed via templates. A PowerCopy (RowAxisBundle) is utilized to duplicate the row plane, row axis, alignment constraints, and necessary publications in a single package. Additionally, a User-Defined Feature (UDF) named ServiceModule_IF acts as a parametric template for the service module, incorporating its base, front edge, and mid-planes while importing SeatPitch_mm externally (as shown in Fig. 3). These templates reference only Publications, enabling rapid setup and minimizing maintenance costs during instantiation.

3.5. DMU Metrics: Alignment Deviation and Clash Sets

Digital Mock-Up (DMU) validation is distilled into two primary measurements:

1. **Alignment Deviation (AlignmentDeviation_mm):** This metric is defined as the signed distance from the seat-cushion front edge (projection line) to the service-module front-edge plane for each row. It is measured using the Measure Between tool (Point/Line to Plane), yielding a distance d_i for each row. For each variant, the average deviation, mean(|d_i|), and the maximum deviation, max(|d_i|), are evaluated against predefined engineering targets: ≤ 3 mm for average deviation and ≤ 5 mm for maximum deviation.

2. **Clashes:** Interferences are checked using DMU Space Analysis between selected sets (Seats vs. ServiceSet and Seats vs. Seats). This outputs specific clash counts

(Clash_SeatService_Count and Clash_SeatSeat_Count), which can be categorized into hard clashes and contacts.

3.6. End-to-End Workflow

The comprehensive parametric workflow proceeds as follows:

1. Select the desired configuration from the Design Table.
2. Update the skeletons, which automatically adjusts the pattern spacing and instance counts.
3. Allow the service set to synchronize automatically via External Parameters.
4. Knowledgeware rules fire to verify range checks, enforce service spacing, and trigger Reaction warnings.
5. Compute DMU metrics, including alignment deviations and clash counts.
6. Generate a mini-report logging the variant ID, measurements, and remarks.

3.7. Bill of Interfaces (Summary)

Interface (Publication)	*Owner Skeleton*	*Type*	*Purpose*
PUB_SeatPitch_mm	SeatLayout_Skeleton	Length	Row spacing (drives both patterns)
PUB_RowCount_Pieces	SeatLayout_Skeleton	Integer	Number of rows
PUB_RowAxis_Pattern	SeatLayout_Skeleton	Geometry set	Instance axes for seats
PUB_RowPlane_Pattern	SeatLayout_Skeleton	Plane set	Front planes per row
PUB_Floor	SeatLayout_Skeleton	Plane	Ground constraint for seats
PUB_ServiceRowPlane_Pattern	ServiceSet_Skeleton	Plane set	Service module front planes (for DMU)

3.8. Formula Summary

- RectPattern.SeatRows.Spacing1 = SeatPitch_mm
- RectPattern.SeatRows.NumberInDir1 = RowCount_Pieces
- RectPattern.ServiceRows.Spacing1 = SeatPitch_mm
- RectPattern.ServiceRows.NumberInDir1 = RowCount_Pieces
- ServiceSet_Spacing_mm = SeatPitch_mm
- AlignmentDeviation_mm(avg) = mean(|d_i|)
- AlignmentDeviation_mm(max) = max(|d_i|)

3.9. Model Governance

- Naming Conventions: Suffixes are strictly applied (see Appendix A), such as _mm for lengths (SeatPitch_mm, AisleWidth_mm) and _Pieces for integers (RowCount_Pieces).
- Version Control: The Design Table file is systematically revised with a ConfigID, date, and author metadata.
- Performance Optimization: For highly populated variants, using the *Deactivate Pattern* function is recommended during editing, followed by a manual *Update*.
- Traceability: A change log is maintained for Publications, and DMU measurements are exported directly as DMU_Report_[ConfigID].xlsx.

3.10. Assumptions and Scope

- The front edge geometry of the seats and the front plane of the service module are treated as published, immutable interfaces.
- Wheelbase variations and body-in-white (BIW) modifications are out of scope for this specific study, though they can be seamlessly integrated into the same architecture if required.

- DMU tolerance bands are adaptable on a per-project basis (recommended bands are provided in Appendix D).

4. Case Study

4.1. Testbed Setup

The study was conducted within a master product assembly named BusInterior_Master.CATProduct, utilizing two distinct skeletons:

- **SeatLayout_Skeleton.CATPart:**
  - **Parameters:** SeatPitch_mm, RowCount_Pieces, and optional parameters like AisleWidth_mm.
  - **Publications:** PUB_Floor, PUB_RowPlane_1, PUB_RowPlane_Pattern, PUB_RowAxis_Pattern, PUB_SeatPitch_mm, and PUB_RowCount_Pieces.
  - **Patterning:** Driven by RectPattern.SeatRows where Spacing = SeatPitch_mm and Instances = RowCount_Pieces.
- **ServiceSet_Skeleton.CATPart:**
  - **External Parameters**: SeatPitch_mm is linked to ..\SeatLayout_Skeleton.1\PUB_SeatPitch_mm, and RowCount_Pieces is linked to ..\SeatLayout_Skeleton.1\PUB_RowCount_Pieces.
  - **Publications:** PUB_ServiceRowPlane_Pattern and optional PUB_ServiceFrontEdgePlane.
  - **Patterning:** Driven by RectPattern.ServiceRows where Spacing = SeatPitch_mm and Instances = RowCount_Pieces.

Assembly Constraints (Summary): The seat component is constrained using PUB_Floor and the first row plane (PUB_RowPlane_1). The service module is aligned to the PUB_ServiceRowPlane_Pattern planes. 3D cabling is excluded from this scope; alignments are performed strictly based on front edges.

4.2. Configurations (Design Table)

Variants were driven via an Excel Design Table. Three representative scenarios were selected for this study (see Fig. 3):

ConfigID	*SeatPitch_mm*	*RowCount_Pieces*	*Notes*
Left_10_750	750	10	Standard / Comfort
Left_12_720	720	12	Standard
Left_8_950	950	8	Extended pitch / VIP

4.3. Step-by-Step Implementation

The implementation workflow proceeds as follows:

1. **Configuration Selection:** In CATIA, insert a Design Table from the file, map the SeatPitch_mm and RowCount_Pieces columns to the model parameters, and select the active row.
2. **Seat Rows (SeatLayout_Skeleton):**
    - RectPattern.SeatRows\Spacing1 = SeatPitch_mm
    - RectPattern.SeatRows\NumberInDir1 = RowCount_Pieces
3. **Service Set (ServiceSet_Skeleton):**
    - Import SeatPitch_mm and RowCount_Pieces via External Parameters.
    - RectPattern.ServiceRows\Spacing1 = SeatPitch_mm
    - RectPattern.ServiceRows\NumberInDir1 = RowCount_Pieces
4. **Assembly Constraints:** The seat part is linked to PUB_Floor (via Coincidence/Offset) and PUB_RowPlane_1. The service module is aligned to the PUB_ServiceRowPlane_Pattern.
5. **Knowledgeware Rules:**
    - Rule: $650 \leq$ SeatPitch_mm ≤ 950; $8 \leq$ RowCount_Pieces ≤ 14
    - Rule: ServiceSet_Spacing_mm = SeatPitch_mm
    - Check: AisleWidth_mm ≥ 400 (optional)
    - Reaction: If AlignmentDeviation_mm exceeds 5 mm, trigger a warning and create a log entry.
6. **DMU Alignment Measurement:** Calculate the signed distance (d_i) between the seat-cushion front edge and the service-module front-edge plane for each row.
    - AlignmentDeviation_mm(avg) = mean(|d_i|)
    - AlignmentDeviation_mm(max) = max(|d_i|)
7. **DMU Clash Detection:** Perform Clash Analysis between the Seats and ServiceSet groups, and within the Seats group itself.
8. **Logging:** Record measurements and capture screenshots. Repeat steps 1–7 for all variants.

4.4. Reproducibility

- **Inputs:** SeatPitch_mm values of 750, 720, and 950 mm; RowCount_Pieces values of 10, 12, and 8; and the interfaces PUB_Floor, PUB_RowPlane_Pattern, and PUB_ServiceRowPlane_Pattern.
- **Outputs:** AlignmentDeviation_mm (avg/max), Clash_SeatSeat, Clash_SeatService, along with corresponding screenshots (e.g., Fig-02…05).

5. Results

The following results present representative values obtained for three distinct configuration variants (750/10, 720/12, and 950/8).

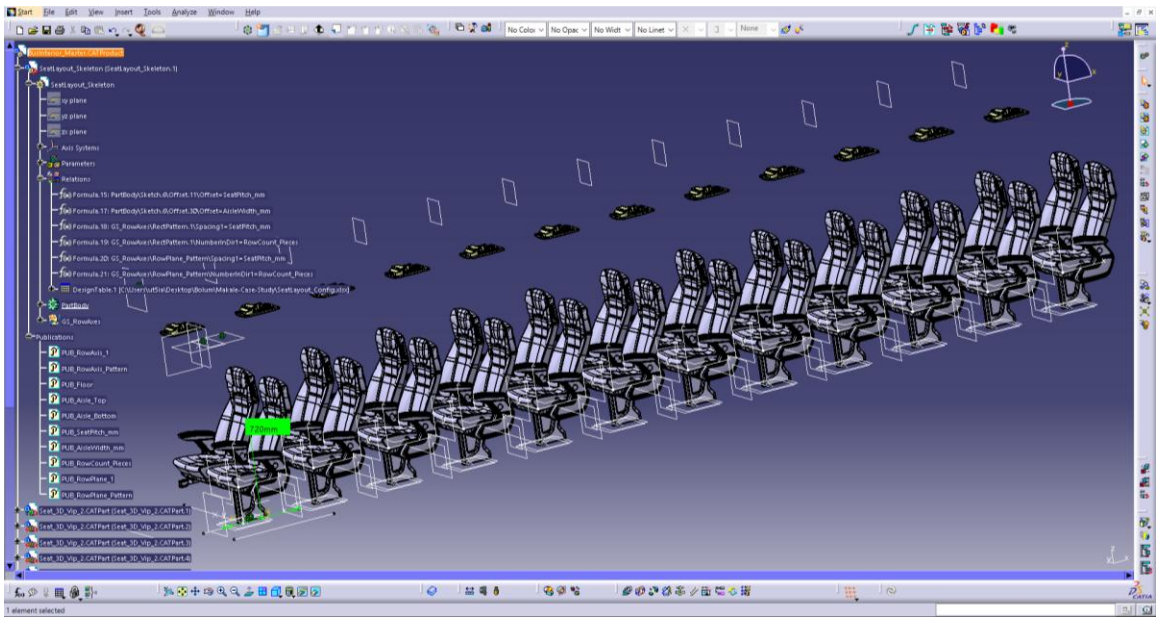
5.1. Time Savings

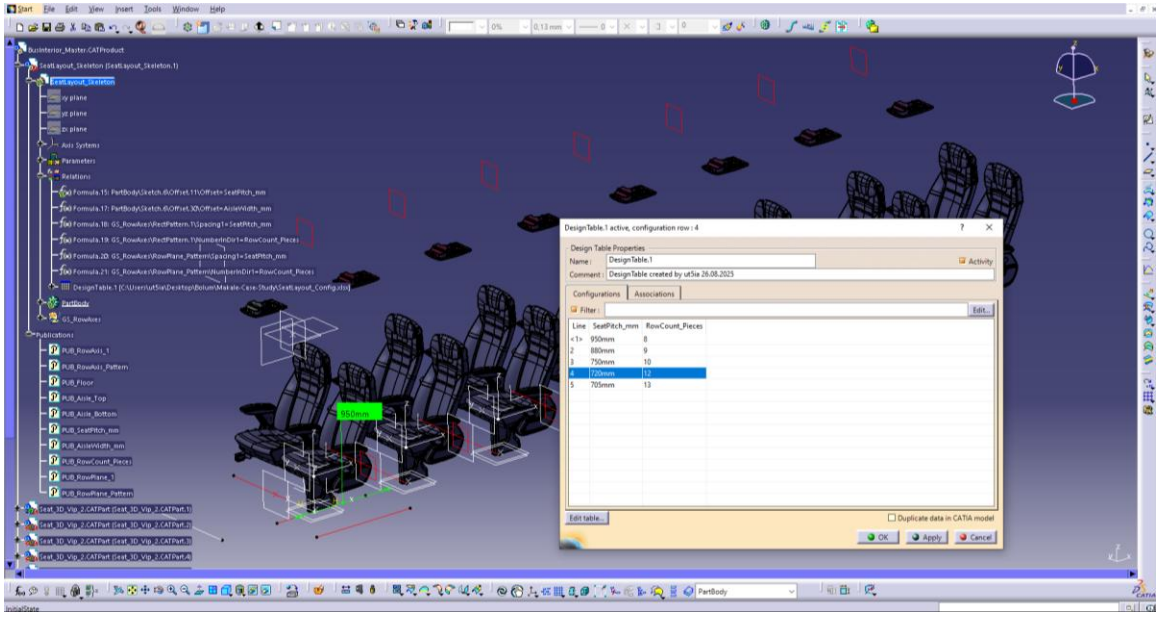
- **Definition:** The engineering time required to completely update the seat and service set layouts following a variant change (i.e., modifying the pitch and row count).
- **Calculation:** Savings (%) = [(Conventional – Parametric) / Conventional] × 100

Table 1. Seat layout design time evaluation.

Variant (Pitch / Rows)	*Conventional (h)*	*Parametric (h)*	*Savings (%)*
750 / 10	**10.0**	**2.0**	**80**
720 / 12	**9.0**	**1.8**	**80**
950 / 8	**7.5**	**1.5**	**80**

- **Observation:** For all evaluated variants, the parametric workflow executes the update with a single click, eliminates the need for re-modeling, and reduces the engineering effort by 80%.




DesignTable.1 active, configuration row : 4
Design Table Properties
Name : DesignTable.1
Comment : DesignTable created by utSia 26.08.2025
Activity
Configurations
Associations
Filter :
Edit...
Line SeatPitch_mm RowCount_Pieces
<1> 950mm 8
2 880mm 9
3 750mm 10
4 720mm 12
5 705mm 13
Edit table...
Duplicate data in CATIA model
OK
Apply
Cancel

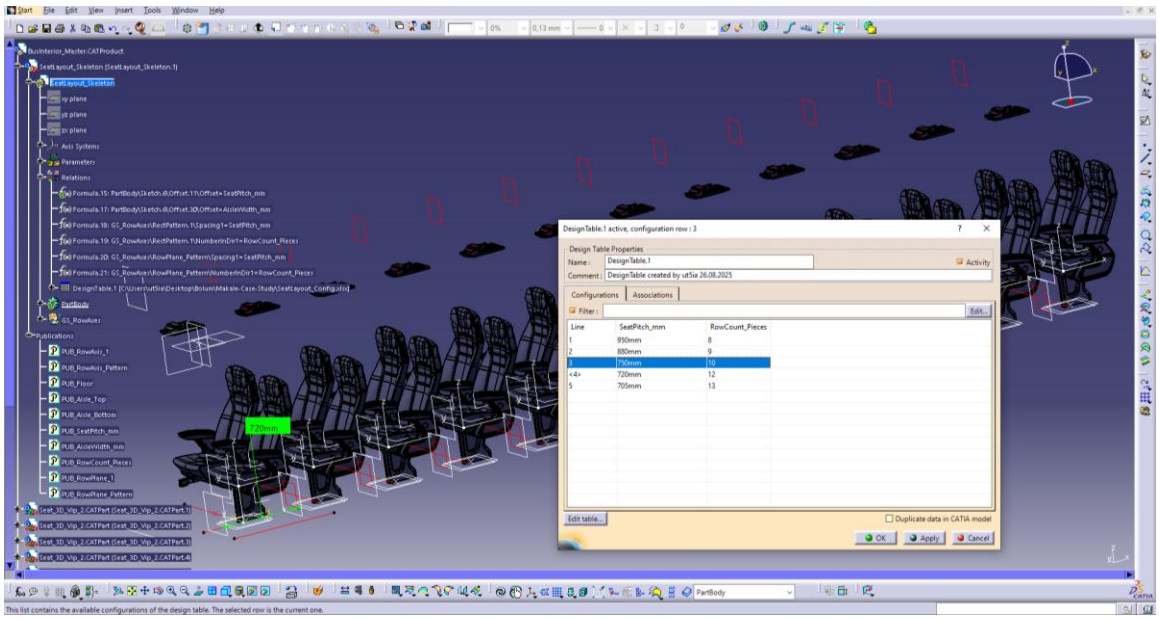


Figure 4. Automated variant generation via Design Table: (a) 950 mm / 8 rows, (b) 720 mm / 12 rows, (c) 750 mm / 10 rows.

5.2. DMU: Alignment and Clashes

- **Alignment Metric:** AlignmentDeviation_mm represents the average (mean) and peak (max) absolute values of the signed distances between the seat cushion front edge and the service module front edge plane for each row.
- **Clash Sets:** Interferences were calculated specifically for Seat–Seat and Seat–Service combinations.

Table 2. DMU alignment and clash analysis results.

Variant	*Alignment Avg (mm)*	*Alignment Max (mm)*	*Seat–Seat Clash (count)*	*Seat–Service Clash (count)*
750 / 10	0.4	1.2	0	0
720 / 12	0.5	1.3	0	0
950 / 8	0.6	1.5	0	0

Targets and Observation: The predefined engineering targets were ≤ 5 mm for maximum alignment deviation and ≤ 3 mm for average alignment deviation. All evaluated variants satisfied

these thresholds: average deviations ranged from 0.4 to 0.6 mm, maximum deviations ranged from 1.2 to 1.5 mm, and no physical clashes were detected (see Fig. 5).

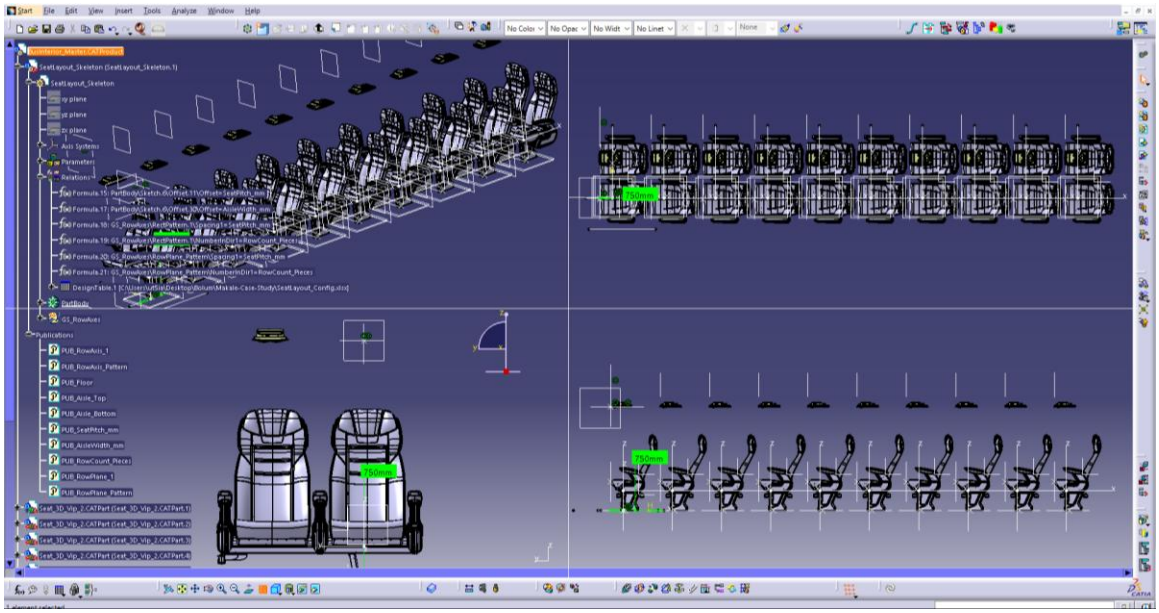


Figure 5. Visual verification of seat-to-service-set alignment during the DMU validation phase.

6. Discussion

6.1. Generalizability

The concept of managing product variants through shared parameters and reusable interface structures is consistent with established principles in Knowledge-Based Engineering and product platform design [5,8]. In such approaches, a stable set of governing parameters acts as the backbone of variant generation while maintaining interface consistency across product configurations. The architecture proposed in this study follows the same principle by using shared parameters such as SeatPitch_mm and RowCount_Pieces to synchronously control both seat layouts and overhead service modules.

- **Single Parameter Source:** Sharing fundamental parameters like SeatPitch_mm and RowCount_Pieces via Publications is not only suitable for seats and service sets, but is equally applicable for driving other overhead modules—such as lighting, ventilation nozzles, luggage racks, and call buttons—at the exact same pitch.

- **Variant Scaling:** When a new configuration row (e.g., 760/11) is added to the Design Table, the workflow functions seamlessly without requiring any modifications to the underlying rules or pattern links.
- **Supplier Changes:** A change in the seat or service module supplier can be efficiently managed by simply re-mapping and re-instantiating the existing UDF/PowerCopy templates.

6.2. Limitations

- **Excluded E/E Scope:** Cable routing and Electrical/Electronic (E/E) mounting components were excluded from this study. If these are to be integrated in the future, additional rules must be defined to manage bending radii, minimum separation distances, and center-of-gravity impacts.
- **High-Level Body Impacts:** Variations in the chassis or body-in-white (BIW)—such as rear wall geometry or roof channel cross-sections—may necessitate supplementary interfaces (e.g., establishing PUB_Body_RefPlanes).
- **Performance Constraints:** Pattern updates may experience performance slowdowns when computing exceptionally high row counts. This issue should be optimized during the editing phase by utilizing the *Lightweight* or *Deactivate Pattern* functions.

6.3. Risks and Mitigations

- **Parameter Deviations:** The risk of entering incorrect pitch or row intervals is mitigated by enforcing min-max constraints via Rule and Check functions. Any violation triggers immediate user warnings and logs through the Reaction mechanism.
- **Unpublished Interface References:** Creating direct geometric links without using Publications inevitably leads to model breakages. This risk is mitigated by implementing an Interface Audit checklist and strictly enforcing a "consume only PUB" modeling discipline across the design team.
- **Broken Excel Links:** If the Design Table file path changes, the parametric link breaks. This is prevented by establishing a standardized version-path protocol under the PLM system and enacting a mandatory "link check" procedure prior to release.

7. Conclusion and Future Work

7.1. Conclusion

- **Unified Control:** Seat and service-set layouts were successfully synchronized through a single shared parameter structure.
- **Performance:** The case study demonstrated an 80% reduction in engineering update time, zero detected physical clashes, and alignment deviations fully within the predefined engineering targets, with a maximum observed deviation of 1.5 mm.
- **Reusability and Stability:** Publications, Knowledgeware, and UDF/PowerCopy supported interface stability and reusable modeling templates.
- **Verification:** The selected DMU metrics provided a practical and repeatable basis for alignment and clash assessment across variants.

7.2. Future Work

1. **E/E Integration:** Developing rule sets for wiring harnesses and lighting systems (e.g., minimum bending radii, separation distances) and conducting DMU clearance analyses.
2. **Automated Reporting:** Creating a macro that generates PDF/Excel DMU outputs—including dimension tables and automated screenshots—with a single click upon Design Table selection.
3. **Optimization Loop:** Implementing pitch and row optimization algorithms (e.g., parameter sweeping) based on targeted aisle widths, seating capacities, and ergonomic metrics.
4. **UDF/PowerCopy Catalog:** Expanding supplier-independent seat and service templates to establish a comprehensive corporate library.
5. **PLM Automation:** Mapping specific Design Table rows directly to ECR/ECN workflows and automatically monitoring the first-pass yield KPI.
6. **Regulation Verification:** Integrating additional Rules and Checks to automatically verify compliance with legal regulations regarding seat pitch and access routes, such as minimum aisle widths and emergency exit clearances.

8. Nomenclature

(A) Acronyms and Abbreviations

Acronym	Term	Description

AP242	STEP Application Protocol 242	Managed model-based 3D engineering data exchange standard.
BIW	Body-In-White	The vehicle's structural frame or sheet metal structure before painting and sub-assembly.
CAD	Computer-Aided Design	Software used for creating, modifying, and analyzing digital product models.
DMU	Digital Mock-Up	3D simulation of a product used for checking clearances, alignments, and interferences (clashes).
DT	Design Table	Spreadsheet-based tool in CATIA used for managing and driving multiple configuration variants.
E/E	Electrical/Electronic	Systems pertaining to wiring harnesses, lighting, and electronic modules.
ECR/ECN	Engineering Change Request / Notice	Formal workflows within PLM for documenting and authorizing design modifications.
KBE	Knowledge-Based Engineering	Engineering methodology utilizing rules and parameters to automate repetitive design tasks.
KPI	Key Performance Indicator	A quantifiable measure used to evaluate the success of an organization or specific process (e.g., first-pass yield).
MBD	Model-Based Definition	The practice of using 3D CAD models to define individual components and product assemblies completely.
PLM	Product Lifecycle Management	System used for managing product data, versions, revisions, and release cycles.

PUB	Publication	A published geometric element or parameter in CATIA serving as a stable interface contract for external assemblies.
UDF	User-Defined Feature	A reusable parametric feature template encapsulated within CATIA Knowledgeware.
UNECE R107	United Nations Economic Commission for Europe Regulation No. 107	Uniform provisions concerning the general construction of category M2 or M3 vehicles (buses/coaches).

(B) Symbols and Parameters

Symbol / Parameter	Type	Unit	Description
SeatPitch_mm	Length	mm	Distance between consecutive seat rows; primary driver for pattern spacing.
RowCount_Pieces	Integer	--	Total number of seat rows; primary driver for pattern instances.
AisleWidth_mm	Length	mm	Minimum clearance for the center aisle (e.g., regulatory constraint).
ServiceSet_Spacing_mm	Length	mm	Longitudinal spacing between overhead service modules.
d_i	Length	mm	Signed distance between the seat front edge and service module front plane at the i-th row.
AlignmentDeviation_mm(avg)	Length	mm	Mean absolute value of d_i across all evaluated rows.
AlignmentDeviation_mm(max)	Length	mm	Maximum absolute value of d_i across all evaluated rows.
Clash_SeatSeat_Count	Integer	--	Total number of physical interferences detected between seat instances.

Clash_SeatService_Count	Integer	--	Total number of physical interferences between seats and the overhead service set.
ConfigID	String	--	Unique identifier representing a specific variant row in the Design Table.
VentCount	Integer	--	Number of ventilation nozzles per service module (extended parameter example).
WheelchairBay	Boolean	--	Flag indicating the requirement for a wheelchair access space (True/False).
AnchorPattern	Enum	--	Hole pattern configuration type instantiated via UDF (e.g., TypeA / TypeB).

References

[1] Dassault Systèmes (2023) CATIA V5 Knowledgeware (Knowledge Advisor) User's Guide. Dassault Systèmes Documentation.

[2] Dassault Systèmes (2022) CATIA V5 DMU Space Analysis & Clash User's Guide. Dassault Systèmes Documentation.

[3] Dassault Systèmes (2023) CATIA V5 Product Knowledge Template (PKT): UDF and PowerCopy. Dassault Systèmes Documentation.

[4] Dassault Systèmes (2021) Design Tables – Linking CATIA with Excel. Dassault Systèmes Documentation.

[5] Verhagen WJC, Bermell-García P, van Dijk REC, Curran R (2012) A critical review of knowledge-based engineering research in product lifecycle. Comput Aided Des 44(1):1–19. https://doi.org/10.1016/j.cad.2011.10.011

[6] La Rocca G (2012) Knowledge-based engineering: Between AI and CAD. Comput Aided Des 44(3):166–179. https://doi.org/10.1016/j.cad.2011.11.002

[7] Pahl G, Beitz W, Feldhusen J, Grote KH (2007) Engineering Design: A Systematic Approach, 3rd edn. Springer, Berlin.

[8] Meyer MH, Lehnerd AP (1997) The Power of Product Platforms. Free Press, New York.

[9] Ullman DG (2017) The Mechanical Design Process, 6th edn. McGraw-Hill, New York.

[10] Sabin D, Weigel R (1998) Product configuration frameworks: A survey. Artif Intell Eng Des Anal Manuf 12(2):79–93.

[11] UNECE (2018) Regulation No. 107: Uniform provisions concerning the approval of category M2 or M3 vehicles.

[12] ISO (2020) ISO 10303-242: Managed model-based 3D engineering (STEP AP242). International Organization for Standardization.

[13] ASME (2019) ASME Y14.41: Digital Product Definition Data Practices. American Society of Mechanical Engineers.

[14] Dassault Systèmes (2021) Assembly Design – Constraints and Publications Best Practices. Dassault Systèmes Documentation.