

Bridging SolidWorks to CATIA: A Cognitive Transfer Approach in CAD Education

Yongmin Zhong¹, Chengfan Gu¹, Jinhe Liu^{1*}

¹ School of Engineering, RMIT University, Melbourne, Australia

*Correspondence: Jinhe Liu

Received: 12 April 2026; Accepted: 13 May 2026; Published: 21 May 2026

Abstract:

Tertiary CAD education frequently treats individual software platforms as isolated skillsets. Consequently, graduates often struggle to transition between different digital design environments. This paper presents a pedagogical approach that addresses this gap. It utilizes students' prior experience with introductory platforms (e.g., SolidWorks) to facilitate cognitive transfer to enterprise-level systems (e.g., CATIA). The intervention incorporates three specific classroom strategies. First, semantic mapping is used during lectures to emphasize universal parametric logic. Second, a paper-based workflow planning exercise requires students to outline feature sequences prior to software interaction. Third, platform-comparative tutoring explicitly links generic modeling with intent to specific interface conventions. This approach was implemented in a final-year advanced CAD course at an Australian university. Student perception data indicated improvements in perceived teaching quality. Furthermore, qualitative feedback demonstrated that students developed greater adaptability to unfamiliar CAD platforms. Although based on single-cohort descriptive data, this design offers a scalable and low-cost pedagogical framework. The primary contribution is the operationalization of cognitive-transfer techniques, shifting CAD instruction from software-specific training toward platform-portable engineering strategies.

Keywords:

Cad Education, Cognitive Transfer, Software-Agnostic Skills, Solidworks, Catia, Engineering Pedagogy

1. Introduction

Modern engineering workflows frequently require graduates to operate CAD platforms they did not formally study. For instance, a student trained in SolidWorks may encounter CATIA, Creo, or NX in industrial practice. This platform challenges tertiary CAD educators to shift instruction from software-specific training to the cultivation of adaptable digital fluency.

Research in CAD pedagogy distinguishes between declarative knowledge (software-specific syntax) and procedural knowledge (strategic reasoning regarding feature sequences and topological dependencies). Traditional CAD instruction often overemphasizes declarative execution. Consequently, learners develop a strong

interface dependency. When transitioning to advanced enterprise-CAD systems, students frequently fail to transfer their prior knowledge. Instead of adapting constraint-based strategies, many regress to trial-and-error navigation, treating the new interface as entirely disconnected from their previous CAD experience.

Classroom observations reveal that students accustomed to the "face-based" extrusion logic of SolidWorks often struggle with CATIA's modular workbench architecture. A frequent error involves novices attempting part-level construction within an assembly-oriented Product Design environment. Such behaviors indicate a fundamental lack of clarity regarding parametric design principles and the hierarchical data structures inherent in enterprise-level software.

This paper presents a pedagogical framework to address this cognitive transfer gap, implemented in a final-year advanced CAD course at an Australian university. While the course utilizes CATIA, most of the cohort enters with self-taught, heterogeneous SolidWorks experience. Instructional design actively leverages this prior knowledge as a foundational cognitive resource. This is achieved through three integrated strategies: semantic mapping, paper-based workflow planning, and platform-comparative tutoring.

This study offers a descriptive account of these interventions, supplemented by student perception data. The primary contribution is practical: providing an operational specification of cognitive-transfer techniques designed to foster platform-portable expertise in software-intensive engineering disciplines.

2. Theoretical Background

2.1. High-Road Transfer and Interface Interference

Transfer of learning frequently stalls when skills are tightly coupled to specific digital environments. Perkins and Salomon [1] differentiate between "low-road" transfer, characterized by reflexive habits formed through extensive practice in constant conditions, and "high-road" transfer, which requires the deliberate abstraction of principles for application in novel contexts. Research indicates that high-road transfer rarely occurs spontaneously; it requires explicit instructional scaffolding [2]. Commercial CAD platforms operate on a shared constraint-based parametric substrate; however, their divergent graphical user interfaces (GUIs) obscure these universal commonalities. Without explicit instructional intervention, learners conflate engineering design intent with software-specific navigation, resulting in interface interference that actively impedes high-road cognitive transfer [3].

2.2. Decoupling Procedural Logic from Declarative Syntax

Early empirical studies on CAD task performance established a critical distinction between declarative knowledge (software-specific syntax and command locations) and procedural knowledge (the strategic sequencing of features, topological dependencies, and constraint hierarchies) [4]. Expert CAD relies predominantly on procedural knowledge [5]. However, conventional instruction frequently privileges declarative execution, producing operators who stall when the interface layout changes.

A direct parallel exists in computer science education, where algorithmic logic is often conceptualized away from the keyboard to isolate computational thinking from language-specific syntax. Similarly, robust CAD pedagogy must decouple topological

logic from software influence. When workflow planning occurs entirely on-screen, the GUI exerts a commanding influence, prompting trial-and-error reactions rather than proactive design. Abstracting the planning phase to a non-digital medium forces students to engage purely with the logical architecture of the model [3], neutralizing the cognitive interference imposed by the software environment.

2.3. Cultivating Adaptive Expertise

The overarching objective of neutralizing interface dependency is the cultivation of adaptive expertise [6]. Unlike routine experts who rely on memorized procedures within familiar constraints, adaptive experts possess the epistemological flexibility to reorganize their foundational knowledge when confronted with unfamiliar conditions [7]. In the context of digital engineering tools, cultivating this adaptability requires explicit instructional mechanisms that consistently expose the platform-independent parametric substrate, rather than merely demonstrating sequential commands in a single interface.

3. Pedagogical Context and Three Bridging Strategies

3.1. Course Context

The pedagogical framework was implemented in a final-year Honours-level CAD elective at a major Australian metropolitan university, primarily enrolling students from Mechanical, Manufacturing, and Aerospace Engineering programs. The platform of instruction is CATIA. While the course has no formal CAD prerequisite beyond foundational engineering graphics, the great majority of the cohort (approximately 75 students in the implementation iteration) arrives with self-taught SolidWorks experience.

The course was delivered in four-hour weekly laboratory blocks using an agile delivery model, calibrating the balance between lecture and practice weekly based on topic complexity. Prior to the intervention, classroom observations indicated a recurring phenomenon: students with strong SolidWorks backgrounds frequently underperformed in the initial weeks. The symptom was not an absence of CAD knowledge, but a failure to recruit it. When confronted with the unfamiliar CATIA interface, learners typically regressed to trial-and-error navigation, attempting to reconstruct workflows without grasping the underlying topological differences. To mitigate this interface interference, three bridging strategies were integrated into the curriculum to explicitly facilitate cognitive transfer.

3.2. Strategy 1: Semantic Mapping in Lectures

The first strategy operates during theoretical instruction and is conversational rather than procedural. When introducing a new CATIA capability, the lecturer routinely names the equivalent capability in SolidWorks (and, where relevant, Creo or NX) and articulates the underlying parametric concept that both instantiate. For example, when introducing the CATIA Pad function or the Sketcher environment, the instructor explicitly links them to the SolidWorks Boss-Extrude and Sketch. The instruction emphasizes that what differs across platforms is largely the toolbar layout and default constraint solver behavior, not the core modeling logic (i.e., generating a volumetric solid from a fully constrained 2D profile).

This routine has two intended effects. First, it explicitly cues high-road transfer by foregrounding the abstraction that connects the students' prior interface to the present one. Second, it relativizes the new interface. Students are guided to recognize GUI differences as surface-level variations rather than conceptual barriers. Table 1 illustrates a representative subset of these semantic mappings.

Table 1. Representative semantic mappings between SolidWorks and CATIA terminology, organized by the underlying parametric concept rather than by interface.

Underlying parametric concept	SolidWorks	CATIA
Fully constrained 2D profile	Sketch	Sketcher
Extrude solid from profile	Boss-Extrude	Pad
Subtract material from profile	Extrude-Cut	Pocket
Hollow a closed body with uniform wall	Shell	Shell
Repeat features along a vector	Linear Pattern	Rectangular Pattern
Repeat feature around an axis	Circular Pattern	Circular Pattern
Reference plane through entities	Reference Plane	Plane (Reference Elements)
Assembly mate / constraint	Mate	Constraint (Coincidence, Contact, Offset)

3.3. Strategy 2: Paper-First Workflow Planning

The second strategy deliberately decouples strategic decision-making from the software environment. Before opening CATIA, students are required to produce a handwritten outline (a “workflow tree”) specifying the intended feature operations (e.g., pads, pockets, fillets, patterns) in their chronological order, alongside brief notes on reference dependencies.

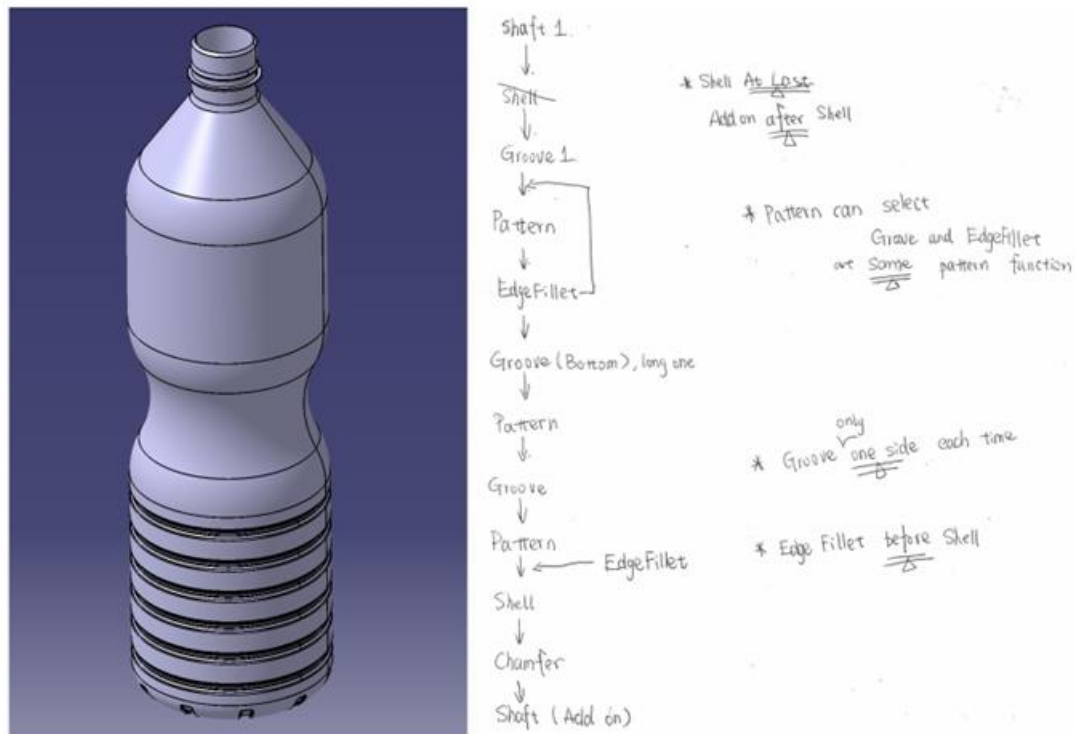


Figure 1. A representative student-produced “workflow tree” and geometric sketch of a bottle-like industrial component. The right-hand annotations document the iterative refinement of modeling logic: initial procedural steps are supplemented with critical heuristics and strategic modifications recorded during a tutor-led debriefing. This practice exemplifies the use of “modeling pseudocode” to decouple strategic intent from software-specific syntax.

This exercise neutralizes a specific cognitive interference: once a student opens the software, the graphical interface presents an immediate procedural temptation to click visually prominent tools and react to the outcomes. By requiring a handwritten outline first, strategic decisions are forced into a representational medium—pencil and paper—that has no platform-specific affordances.

A second function of this strategy emerges through assessment. The handwritten outline is collected alongside the digital model and graded for internal coherence. This communicates a crucial instructional message: multiple feature sequences can yield identical geometry, and the assessment values robust strategic reasoning over the imitation of a single canonical path.

Initially, this strategy encountered significant student resistance. Many learners perceived manual sketching as an obsolete hurdle. To overcome this, the teaching team framed the exercise as a form of “modeling pseudocode,” explicitly comparing CAD planning to algorithmic design in computer science. Instructors demonstrated how a robust logical structure prevents the “topological collapse” often triggered by late-stage parametric modifications. Eventually, this strategy transitioned into a self-regulated habit, accelerating overall modeling sessions by eliminating mid-build procedural hesitation.

3.4. Strategy 3: Platform-Comparative Tutoring

The third strategy governs one-to-one laboratory interactions. When a student reaches a procedural impasse, the standard reaction in conventional tutoring is for the tutor to take the keyboard, perform the required action, and return control. While efficient, this actively obstructs cognitive transfer.

The present teaching team adopted an explicit alternative protocol. Tutors were instructed to avoid immediate declarative correction. Instead, the tutor asks what geometric or topological result the student intends to achieve, and how they would accomplish it in SolidWorks. Only then does the tutor bridge to the CATIA-specific path.

For instance, when a student struggles with a pattern operation, the tutor directs attention to specific feature-definition options, such as CATIA’s Keep Specification constraint. Rather than dictating the setting, the tutor prompts the student to observe the topological consequences through guided trial-and-error. This process forces the student to recognize that CATIA requires a more rigorous definition of parent-child dependencies compared to the relatively automated behaviors of SolidWorks.

This protocol requires deliberate preparation and weekly tutor debriefings. While it is slower than direct intervention in absolute terms—a single interaction might take three to four minutes rather than thirty seconds—it produces a fundamentally different learning dynamic. By the second half of the semester, students routinely initiated these comparisons themselves (e.g., “I think this is the same as a SolidWorks Loft, isn’t it?”), indicating that the cognitive bridge was actively functioning.

4. Student Response

To evaluate the efficacy of the cognitive-transfer strategies, student response data was collected via the university’s standardized Course Experience Survey (CES) at the conclusion of the implementation iteration. While the aggregated institutional metrics indicated a marked improvement in overall student satisfaction and perceived

teaching quality compared to prior iterations, the primary focus of this evaluation rests on the open-ended qualitative feedback. Thematic analysis of these responses revealed a distinct shift in how students perceived their own digital fluency, extending beyond simple CAD operation.

4.1. Theme 1: Confidence with Unfamiliar Platforms

A dominant theme across the free-text responses was a significant reduction in interface anxiety. Students consistently reported that they no longer regarded an unfamiliar CAD package as an obstacle requiring a fresh learning curve; instead, they viewed it as a different visual interface overlaying a logic they already understood. One student characterized the course as having taught them “how to think in CAD” rather than “how to use CATIA.” This framing directly mirrors the intended outcome of semantic mapping and paper-first strategies: separating logical design from software-specific execution.

4.2. Theme 2: Cross-Domain Transfer to CAE Environments

Beyond CAD platform migration, the most notable qualitative signal was evidence of cross-domain cognitive transfer. Multiple students reported that the procedural logic cultivated in the CAD course provided a distinct advantage when they subsequently engaged with Finite Element Method (FEM) software, such as ANSYS and Abaqus. Students highlighted that the rigorous approach to defining topological dependencies and managing parent-child constraints—reinforced through platform-comparative tutoring—directly facilitated their comprehension of geometry preparation and mesh logic in CAE environments.

4.3. Theme 3: Formative Feedback on Resources

A minority theme concerned gaps in the available supplementary resources, particularly the consistency of short, component-specific instructional videos. This feedback was treated as formative input by the teaching team and acted upon in the subsequent course iteration. It is reported here for transparency and demonstrates the iterative nature of the course design, rather than contradicting the dominant themes of enhanced cognitive transfer.

5. Discussion and Implications

5.1. The Shared Logic of Cognitive Decoupling

The three strategies described in Section 3 differ in delivery mode—lecture, individual desk work, and one-to-one tutoring—but share a unified cognitive logic. Each strategy explicitly exposes the platform-independent substrate of parametric modeling. Whether by naming equivalents directly (semantic mapping), externalizing intent in a non-digital medium (paper-first workflow planning), or routing tutor interventions through prior experiences (platform-comparative tutoring), the interventions restructure how existing knowledge is accessed.

Read against the transfer-of-learning literature, these strategies collectively operationalize the deliberate abstraction step required for high-road cognitive transfer [1,2]. Furthermore, the unexpected qualitative evidence of cross-domain transfer to CAE/FEM environments suggests that decoupling procedural logic from declarative syntax does more than simply ease CAD platform migration. It cultivates an algorithmic approach to engineering design—a core tenet of adaptive expertise [7].

5.2. Practical Implications for Course Designers

For engineering educators seeking to cultivate platform-portable fluency, three practical implications emerge:

- Prior Knowledge as a Cognitive Asset:** The heterogeneous CAD experience students bring to advanced courses must be treated as a foundational resource rather than an inconvenience. Instructional paradigms that actively suppress prior habits (e.g., “forget what you learned in SolidWorks”) inadvertently create the exact interface interference they subsequently must solve.

- The Efficacy of Linguistic Bridging:** The most consequential instructional shift reported here is largely linguistic. By consistently articulating the equivalent terminology and shared parametric concepts across platforms, educators can construct a cognitive bridge that requires preparation but zero infrastructure investment.

- Analog Planning for Digital Execution:** Externalizing strategic workflows onto paper prior to software interaction is a near-zero-cost intervention with disproportionate educational returns. Treating this analog step as “modeling pseudocode” forces students to confront topological dependencies without the distraction of GUI navigation.

5.3. Limitations

The present account is bound by several limitations. The study describes a single course at a single institution with one implementation cohort, precluding claims of universal generalizability across other disciplines or CAD platforms. Furthermore, the analysis relies on descriptive, self-reported perception data (CES and qualitative feedback). The study lacks externally marked, independent outcome measures of CAD competence, and no longitudinal post-graduation tracking was conducted to verify if this perceived adaptability translates into sustained workplace capability. Consequently, this paper is positioned as a descriptive pedagogical design specification rather than an evaluative causal study.

5.4. Directions for Future Work

Future investigations should systematically address these limitations. A multi-site implementation across different pairs of CAD platforms would clarify which elements of the instructional design generalize. Crucially, future studies must incorporate independently administered platform-migration tasks—where students are required to replicate a complex model in an entirely unfamiliar CAD environment—to provide objective outcome measures. Finally, longitudinal tracking of graduates into their early careers is necessary to validate the durability of these platform-agnostic heuristics in dynamic industrial settings.

6. Conclusions

Tertiary CAD education must confront the volatility of modern industrial tooling by shifting its focus from software-specific training to the cultivation of adaptable digital fluency. This paper described a pedagogical framework that treats students' prior, self-taught CAD knowledge as a cognitive bridge to enterprise-level platforms rather than a source of interference.

Through semantic mapping, paper-first workflow planning, and platform-comparative tutoring, the instructional design successfully decoupled procedural

engineering logic from declarative interface execution. The qualitative response strongly indicates that these low-cost strategies not only reduce immediate interface anxiety but also foster the robust, algorithmic thinking necessary for cross-domain transfer to broader digital engineering environments. While further empirical validation is required, the operational blueprint provided here offers a scalable path forward. The ultimate objective is clear: CAD pedagogy must transcend the training of interface operators to educate engineers equipped to master any digital design environment the industrial sector presents.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

- [1] D.N. Perkins and G. Salomon, Transfer of Learning. International Encyclopedia of Education (2nd ed.): Pergamon Press, 1992.
- [2] J.M. Royer. Theories of transfer of learning, Educational Psychologist., 1978.
- [3] E.N. Wiebe, Transfer of Learning Between 3D Modeling Systems, The Engineering Design Graphics Journal, 2009, 67(3), DOI: 10.18260/edgj.v67i3.25.
- [4] G.T. Lang.; R.E. Eberts.; M.G. Gabel.; M.M. Barash. Extracting and Using Procedural Knowledge in a Cad Task. (in English), IEEE Transactions on Engineering Management, 1991, 38(3), 257-268, DOI: 10.1109/17.83758.
- [5] H. Besharati-Foumani.; M. Lohtander.; J. Varis, Fundamentals and new achievements in feature-based modeling, a review. (in English), Procedia Manuf, 2020, 51, 998-1004, DOI: 10.1016/j.promfg.2020.10.140.
- [6] O. Pierrakos. measuring adaptive expertise in engineering education. ASEE Annual Conference, 2016.
- [7] J. Kua.; W.S. Lim.; W. Teo.; R.A. Edwards. A scoping review of adaptive expertise in education. Med Teach, 2021, 43(3), 347-355, DOI: 10.1080/0142159X.2020.1851020.



© 2026 by the author(s); licensee International Technology and Science Publications (ITS), this work for open access publication is under the Creative Commons Attribution International License (CC BY 4.0). (<http://creativecommons.org/licenses/by/4.0/>)