



## An Enhanced Model For Developing Students' Computer Modeling Skills Based On The Use Of A Web Platform

Xazratqulov Abduvohid

Jizzakh Polytechnic Institute, Uzbekistan

### Abstract

Web-based learning environments have matured into robust ecosystems for teaching computer modeling, enabling ubiquitous access, immediate feedback, and collaborative design workflows. This article presents an enhanced instructional model that integrates a browser-accessible platform for parametric and numerical modeling with structured pedagogy grounded in experiential and sociocultural learning theories. The model sequences learning from conceptual representation to validated simulation and communicative visualization, while the platform orchestrates content delivery, versioned artifacts, embedded assessments, and analytics. Implemented in an undergraduate context, the approach demonstrated measurable gains in representational fluency, accuracy of model assumptions, and transfer of skills to domain-specific tasks. Students progressed from isolated tool operation to principled reasoning about constraints, boundary conditions, meshing, and numerical stability. The discussion analyzes design choices that produced these outcomes, including formative feedback loops, interoperability with desktop tools, and reflective prompts that externalize tacit modeling decisions. The article concludes with implications for curriculum design and institutional policy to sustain equity of access, academic integrity, and analytics-supported advising.

### Keywords

Web-based learning; computer modeling; parametric design; simulation literacy; formative assessment; learning analytics; interoperability; higher education.

### Introduction

Teaching computer modeling has long been constrained by specialized software, local installations, and heterogeneous student hardware. Web platforms now consolidate modeling editors, computational backends, and rendering pipelines behind a browser interface, removing logistical barriers and supporting immediate, scalable feedback. Yet technology alone does not guarantee deeper learning. Students often learn to reproduce interface gestures without understanding the semantics of constraints or the epistemic status of simulations. The central challenge for educators is to transform a web platform from a delivery channel into a cognitive and social environment where models function as arguments about the world, and where evidence emerges from disciplined parameterization and verification rather than ad-hoc trial and error. This article proposes an enhanced model that aligns platform features with pedagogical intentions so that modeling skills develop as an integrated competence of representation, reasoning, validation, and communication.

The study aims to design and justify an instructional model that leverages a web platform to improve students' computer modeling skills, defined as the ability to construct parametrically

coherent models, formulate and justify assumptions, select and calibrate numerical methods, and communicate results for decision-making. The objectives were to raise accuracy and transferability of student work, shorten feedback cycles, and make modeling processes traceable for assessment and advising.

The model was implemented over one semester in a second-year course enrolling students from mechanical, civil, and information engineering tracks. The chosen web platform provided parametric sketching, geometry kernels, mesh generation, cloud compute for simulations, and a versioned repository with branching and commenting. Interoperability was ensured through open exchange formats for geometry and meshes, enabling optional handoff to desktop suites for advanced cases without fragmenting assessment.

Pedagogically, the course was structured into iterative design sprints anchored by authentic tasks such as optimizing a bracket under fatigue constraints or evaluating airflow in a simplified duct. Each sprint required a problem statement with measurable criteria, a modeling plan with assumptions and expected sensitivities, a sequence of model states linked to rationale, and a verification memo with residual checks or convergence studies. The platform hosted templates for these artifacts and embedded low-stakes quizzes that probed conceptual thresholds rather than interface memory.

Assessment combined analytic rubrics for problem framing, modeling coherence, verification strategy, and communicative clarity with log-based indicators from the platform. These indicators included the frequency and granularity of commits, the ratio of parametric edits to manual overrides, and evidence of convergence testing. Pre/post tasks measured representational fluency through timed translations among verbal descriptions, equations, and parametric features. Semi-structured interviews captured students' evolving views of what counts as a justified model.

Faculty development supported the implementation with workshops on designing prompts that elicit reasoning, on interpreting platform analytics as formative diagnostics, and on maintaining integrity when AI assistants are available. A transparent policy required students to disclose any generative assistance and to verify outputs through independent checks, which were graded as part of the memo.

The enhanced model improved learning outcomes by tightening the coupling between representation and reasoning. Students initially constructed models as static drawings and treated simulation as a black box. As sprints progressed, the requirement to articulate assumptions before modeling and to defend parameter choices during oral reviews shifted attention to dependency structures and sensitivity. The web platform amplified this shift because every change created a persistent, commentable state. Instructors could point to a dimension edit and ask why the related constraint was not updated, or why a boundary condition remained fixed after materials changed. This visibility transformed feedback from generic to surgical and made revision a normal part of learning.

Evidence from rubrics showed the largest gains in verification strategy and communicative clarity. Students learned to report mesh independence with succinct plots and to bound error with simple analytical estimates, replacing vague assurances with quantitative checks. Representational fluency improved as learners moved more confidently among sketches, feature trees, and governing relations. Platform analytics corroborated these patterns: commit messages became more descriptive, clustering around conceptual milestones rather than



arbitrary save points; parametric edits displaced manual tweaks; and branch merges included linked comments that justified design choices. Interviews indicated that students began to perceive modeling as an argumentative process, where assumptions, not screenshots, carried the epistemic load.

The model also enhanced inclusivity and resilience. Browser access and cloud compute reduced disparities in local hardware, while time-boxed compute tokens prevented a few teams from monopolizing resources. Interoperability preserved student agency: those with access to specialized desktop solvers could export and re-import results, but assessment focused on reasoning and verification, not on tool prestige. Academic integrity concerns were addressed by foregrounding verification over production value. When AI tools proposed features or scripts, students were required to test them against known cases and to explain failure modes, turning potential shortcuts into teachable moments about robustness.

Several design choices were pivotal. Embedding micro-assessments at natural friction points kept cognitive load manageable without diluting rigor. Requiring a pre-simulation checklist with unit, constraint, and boundary audits reduced meaningless runs and refocused effort on model fitness. Maintaining a consistent rhetorical structure for memos—claim, evidence, warrant—helped students avoid the common pitfall of reporting pictures without argument. Finally, aligning platform analytics with rubric dimensions allowed instructors to triage support, directing coaching to teams whose logs suggested unproductive iteration or brittle parameterization.

The approach had limits. Highly nonlinear or multiphysics problems occasionally exceeded the web backend's capacity, and students needed guidance to scope problems to platform capabilities. A minority remained tool-centric, equating more detailed meshes with better science; targeted mini-lectures on numerical error and diminishing returns partially corrected this misconception. Sustainability required institutional commitments to license management, privacy-aware analytics, and scheduled maintenance that respected assessment windows.

A web platform can be more than a convenient host for modeling software when its affordances are interwoven with a pedagogy that treats models as disciplined arguments. The enhanced model described here produced reliable improvements in students' capacity to structure parameters and constraints, to justify assumptions with verification, and to communicate results for decision-making. Its strength lies in closing the loop between intention, representation, computation, and reflection, with analytics that render learning processes visible and coachable. Programs adopting this approach should invest in interoperability, formative feedback design, equity of access, and integrity policies that harness rather than deny AI assistance. Future work should refine analytics for detecting misconception patterns, extend the model to cross-disciplinary projects that couple physical and data-driven models, and study long-term transfer into internships and capstone research.

## References

1. Kolb D. A. *Experiential Learning: Experience as the Source of Learning and Development*. — Englewood Cliffs: Prentice Hall, 1984.
2. Hestenes D. Modeling theory for physics instruction // *American Journal of Physics*. — 1987. — Vol. 55, No. 5. — P. 440–454.
3. Pahl G., Beitz W., Feldhusen J., Grote K.-H. *Engineering Design: A Systematic Approach*. — London: Springer, 2007.

4. Prince M. Does active learning work? A review of the research // Journal of Engineering Education. — 2004. — Vol. 93, No. 3. — P. 223–231.
5. De Jong T., van Joolingen W. R. Scientific discovery learning with computer simulations of conceptual domains // Review of Educational Research. — 1998. — Vol. 68, No. 2. — P. 179–201.
6. Freeman S. et al. Active learning increases student performance in science, engineering, and mathematics // PNAS. — 2014. — Vol. 111, No. 23. — P. 8410–8415.
7. Wieman C., Perkins K. Transforming physics education // Physics Today. — 2005. — Vol. 58, No. 11. — P. 36–41.
8. Ullman D. G. The Mechanical Design Process. — New York: McGraw-Hill, 2017.

