PriMera Scientific Engineering
Volume 8 Issue 4 April 2026
DOI: 10.56831/PSEN-08-260
ISSN: 2834-2550

WI-LUX: What If Learning UX, an Educational Conceptual Framework Integrating UX and AI (Implementation Prospects)

Type: Literature Review
Received: March 13, 2026
Published: April 02, 2026

Citation:
Robin Vivian., et al. “WI-LUX: What If Learning UX, an Educational Conceptual Framework Integrating UX and AI (Implementation Prospects)". PriMera Scientific Engineering 8.4 (2026): 03-15.

Robin Vivian[1]* and Mathilde Vosgiens[2]

[1]*Laboratoire Perseus, Universite de Lorraine Metz, France*

[2]*Freelance UX designer, France*

***Corresponding Author:** Robin Vivian, Laboratoire Perseus, Universite de Lorraine Metz, France.

Abstract

University teaching of computer science, and more specifically programming, faces persistent challenges in terms of deep understanding and abstract reasoning. Intelligent tutors based on large language models (LLMs) offer personalized support but often remain intrusive by proposing final solutions to problems. This article proposes applying the “what-if” concept from UX design—generating hypothetical scenarios to challenge assumptions—in order to encourage proactive exploration and counterfactual reasoning. A conceptual framework is presented, including prompt patterns, interactive interface, and exploration loop. A prospective discussion addresses implementation in a university context: pedagogical integration, ethical and technical challenges, and expected benefits in terms of engagement and comprehension. This work remains a theoretical contribution with practical implications for the evolution of computer science education with AI.

Keywords: AI in education; intelligent tutor; what-if scenarios; UX design; algorithms and data structures; exploratory learning

Introduction

Teaching programming and data structures is a fundamental part of university computer science education, particularly in the early years of a bachelor’s degree. Despite its importance, it remains an area where students encounter persistent difficulties: the delicate transition from implementation to conceptual understanding, insufficient mastery of invariants, limited understanding of borderline and degenerate cases, and recurring confusion around asymptotic complexities. Traditional teaching approaches—lectures, static exercises, structured tutorials—offer a proven framework, but they do little to encourage independent exploration, interactive manipulation, or hypothetical reasoning, which are essential for developing a deep understanding of algorithmic mechanisms. The recent emergence of intelligent tutors based on large language models (LLMs) opens up new possibilities for supporting this type of learning. It is clear that we are operating within a framework where AI is not a substitute for student work. The temptation to delegate everything to generative tools is becoming widespread and poses a real danger, as it will unfortunately only give students the illusion of understanding. However, several studies show that these models can provide feedback of comparable or even superior quality to that of human instructors on targeted programming or algorithmic tasks, particularly in recursion and code analysis (Prather, J., et al. 2025). Real-world deployments, such as the CodeAid study, confirm their potential to support students in their programming activities and strengthen their engagement (Kazemitabaar et al 2024). However, these systems remain largely reactive: they respond to learners' requests, but do not spontaneously encourage the exploration of variants, the formulation of hypotheses, or the questioning of initial intuitions. Yet counterfactual reasoning—at the heart of the "what if" concept—is a powerful lever for promoting conceptual understanding. A recent systematic review highlights that counterfactual explanations improve interpretability, capacity for action, and performance in various educational contexts (Jamali et al. 2025). In algorithmics, this type of reasoning is particularly relevant: "What happens if the table is almost sorted?", "What if we change the pivot strategy in quicksort?" "What if the graph becomes dense or contains negative weights?" These hypothetical scenarios allow us to explore boundary behaviors, understand the internal mechanisms of algorithms, and develop a more robust intuition.

At the same time, work in UX design and educational AI highlights the importance of interactive, manipulable, user-centered interfaces. A recent study on inductive educational platforms shows that automatic adaptation features have limited impact when they lack discoverability, while direct interactivity remains a key driver of engagement (Strielkowski et al. 2025). Human-centered approaches to educational AI also emphasize the need to involve teachers and students in the design of tools to ensure their pedagogical relevance and sustainable adoption (Munoz A 2025). Despite these advances, a gap remains in the literature: no framework explicitly proposes integrating the UX concept of "what if" into LLM tutors for university teaching of algorithms and data structures. This article aims to fill this gap by introducing a conceptual framework called "What If Learning UX" that combines counterfactual reasoning, interactive design, and the generative capabilities of LLMs. It also offers a prospective analysis of its implementation in university curricula, taking into account pedagogical, technical, ethical, and institutional issues.

The contributions of this article are as follows:

1. Proposal of a conceptual framework integrating "what if" reasoning into LLM tutors for teaching algorithms.
2. Adaptation of the UX concept of hypothetical scenarios to an exploratory learning context in computer science.
3. Prospective analysis of university implementation, including pedagogical, technical, and institutional challenges and levers.

The rest of the article is organized as follows. Section 2 presents the existing literature on LLM tutors, counterfactual reasoning, and UX approaches in educational AI. Section 3 introduces the conceptual framework "What If Learning UX." Section 4 discusses the prospects for implementation in university curricula. Finally, Section 5 offers a general discussion and avenues for future research.

Literature review

Research on algorithmic teaching draws on several disciplinary concepts: intelligent tutoring systems (ITS), adaptive learning, studies on human-machine interaction and educational UX design, and more recently, work evaluating the use of large language models (LLMs) as conversational tutors. This section summarizes this work and compares empirical results and limitations. It identifies the gaps that the What If Learning UX framework aims to fill.

ITS and adaptive learning: achievements and limitations

For several decades, ITS have demonstrated their ability to improve procedural learning through learner modeling and adaptive feedback (Koedinger & Corbett, 2006; Kulik & Fletcher, 2016). These systems use cognitive models to diagnose errors and propose targeted remedies, which often results in measurable gains in performance on repetitive tasks. However, the construction and maintenance of these models remain costly and specialized. Traditional ITSs tend to focus on error correction rather than encouraging conceptual exploration (VanLehn, 2011). In algorithmics, this approach limits students' exposure to edge cases and structural variations (tables, lists, trees, graphs), which are essential for developing a robust intuition about complexity and invariants.

LLMs as conversational tutors: promises and risks

The emergence of LLMs has opened up new possibilities for natural language feedback, example generation, and large-scale personalization. Recent studies show that, on targeted programming tasks, LLMs can produce explanations and feedback that are perceived as useful by students and comparable to those of human instructors (Nguyen et al., 2024; Kazemitabaar et al., 2024). These models offer continuous availability and the ability to reformulate explanations at different levels of detail, facilitating access to formative support. Nevertheless, several studies highlight significant limitations: a tendency to hallucinate on technical subjects, a lack of rigor for formal proofs, and above all a predominantly reactive stance—LLMs respond to requests but do not systematically initiate exploratory paths or conceptual challenges (Jacobsen & Weber, 2025). These limitations highlight the need to combine linguistic generation with verifiable execution.

Counterfactual reasoning and "what if" pedagogy

The literature on counterfactual explanations and hypothetical reasoning shows that presenting alternative scenarios promotes interpretability, actionability, and generalization (Bunay Guisnan et al., 2025). In an algorithmic context, "what if" scenarios (e.g., almost sorted, presence of duplicates, particular distribution of inputs) expose students to degenerate behaviors and thresholds where performance changes, thereby reinforcing intuition about invariants and amortizations. Earlier research in cognitive psychology and learning sciences (Chi et al., 1989; VanLehn et al., 1992) also showed that formulating self-explanations and comparing variants promotes deep knowledge construction rather than superficial learning.

UX design and interactivity for exploratory learning

Recent research in educational UX converges on a clear finding: the manipulability and discoverability of adaptive features are key factors in sparking and maintaining learner engagement, as well as promoting long-term knowledge retention (Jamali et al., 2025; Topali et al., 2025). When the interface offers rich and intuitive interactivity, users no longer simply consume prefabricated content; they become active participants in their cognitive exploration. Direct manipulation elements such as sliders for continuously adjusting parameters (speed, quantity, critical threshold, etc.), step-by-step animations that break down complex phenomena into visible and controllable steps, side-by-side comparators that juxtapose several scenarios or initial/final states, multidimensional cursors, drag-and-drop areas for reorganizing conceptual elements, and interactive timelines transform learning into an authentic experiential experience. These affordances encourage rapid hypothesis formulation, immediate testing, observation of direct consequences, and reflective iteration, processes at the heart of discovery learning (Bruner, 1961) and experiential learning (Kolb, 1984) models. They also reinforce the sense of agency and perceived control, two powerful psychological levers that increase intrinsic motivation, reduce anxiety about failure, and significantly improve engagement indicators (time spent, number of meaningful interactions, voluntary return to the activity). Platforms such as PhET Interactive Simulations, GeoGebra, and Desmos perfectly illustrate this principle: by making scientific or mathematical concepts tangible and modifiable at will, they allow learners to construct their own understanding rather than passively receiving it. The danger is that the cognitive model constructed may not be optimal or even correct.

Conversely, overly automatic or opaque adaptations—for example, an algorithm that silently adjusts the level of difficulty, reorganizes the course without visible justification, or hides the personalization criteria—have well-documented deleterious effects. The lack of transparency reduces confidence in the tool, as learners do not understand why or how the system made a particular decision; this opacity fuels a feeling of powerlessness and loss of control. Cognitively, it hinders the development of metacognition: users can no longer reflect on their own learning processes or identify the strategies that work for them. Ultimately, it encourages a form of blind delegation to the machine, where learners accept recommendations without questioning them, which can slow down the acquisition of self-regulation, critical thinking, and independent problem-solving skills. Several observational and experimental studies show that this phenomenon is particularly pronounced among younger or less confident learners, who are at risk of developing a dependence on adaptive AI to the detriment of their own intellectual initiative. To counter these risks, design principles must incorporate intentional transparency: making adaptation mechanisms visible (e.g., via a "Why this suggestion?" panel or a concise contextual explanation), offering explicit personalization controls ("Return to my previous choice" button, "Disable automatic adaptation," "View current settings"), and offering integrated reflection tools (assumption log, "Explain my reasoning" button, metacognitive prompts). Good UX design thus transforms the adaptive tool into a collaborative partner rather than a black box, promoting rich, intentional, and sustainable exploratory learning, where interactivity is not a gimmick but the very foundation of active knowledge construction.

Verifiability, hybridization, and pedagogical integrity

To effectively limit the risks of factual errors and hallucinations inherent in large language models (LLMs), many recent studies recommend a hybrid approach combining pure language generation with instrumented or formal execution mechanisms (Song et al., 2024; Quan 2024; Jacobsen & Weber, 2025). This hybridization aims to overcome the intrinsic limitations of probabilistic models by coupling them with more rigorous and verifiable external tools. Thus, the integration of instrumented simulators, systematic automatic verifications, and, when justified by the nature of the content, formal tools such as symbolic execution or theorem provers (such as Alt-ego, Prouver9, or Spark) not only makes it possible to cross-check and validate the explanations produced by the model, but also to detect potential inconsistencies upstream, thereby substantially strengthening the pedagogical confidence of learners and teachers. Furthermore, this technical strategy must be accompanied by rigorous ethical governance: this includes increased transparency on the limitations and sources of the responses generated, the provision of "AI-free" assessment methods to guarantee the authenticity of student work, and regular audits of cognitive and cultural biases built into the models. These measures are considered essential conditions for the responsible and sustainable deployment of these technologies in educational contexts (Topali et al., 2025).

Summary and identified gaps

In summary, the state of the art reveals three main findings: (1) LLMs offer unprecedented potential for personalization and explanation generation, (2) counterfactual reasoning is an underutilized pedagogical lever in AI environments, and (3) pedagogical success will depend on careful integration between generation, verifiable execution, and manipulable UX design. The concrete gaps are the absence of a structured framework for producing proactive what-if scenarios, the lack of longitudinal studies measuring the impact on transfer and retention, and the insufficiency of protocols for systematic verification of generative outputs. These findings directly motivate the development of the What If Learning UX framework, which aims to combine structured prompts, instrumented simulators, and interactive interfaces to make LLM tutors proactive, verifiable, and pedagogically effective.

WI-LUX conceptual framework: What-If Learning UX

Levels of assumptions and pedagogical granularity

The framework distinguishes four levels of assumptions that structure the learning experience and guide pedagogical objectives. Level 1 covers simple modifications (size, distribution) to develop empirical intuition; Level 2 focuses on changes in hypotheses (almost sorted, duplicates, switching from processing with a table to a linked list) in order to expose degenerate cases; Level 3 compares algorithmic variants (pivot strategies, heuristics) to bring out trade-offs and invariants; Level 4 proposes theoretical counterfactuals (computational models, P vs. NP hypotheses) to stimulate metatheoretical thinking. Each level combines tasks, metrics, and appropriate metacognitive questions, enabling pedagogical progression from intuition to abstraction.

Architectural components

The proposed architecture is modular and combines three complementary layers. The LLM layer orchestrates the controlled generation of scenarios, pedagogical reformulation, and the synthesis of counterfactual explanations. The verifiable execution layer runs instrumented simulations, collects metrics (time, operation comparisons, memory), and integrates verification mechanisms (unit tests, consistency checks, possibility of hybridization with symbolic tools). The interactive UX layer provides manipulable controls, dynamic visualizations, and interaction logs for teacher diagnosis. This separation facilitates traceability, reproducibility of experiments, and limitation of hallucinations by cross-checking with actual executions.

Prompt patterns and pedagogical dialogues

The framework formalizes prompt patterns aimed at guiding the LLM toward actionable and verifiable explanations. These include guided exploration (generating and comparing multiple what-if scenarios), justification constraints (requiring invariants and execution examples), adaptive diagnosis (proposing scenarios based on error profiles), and challenge response (the student formulates a hypothesis before simulation). These patterns include templates to limit ambiguity, instructions for requesting partial proofs or counterexamples, and strategies for inviting students to propose their own variations, promoting the co-construction of knowledge.

Proactive exploration loop

The loop combines suggestion, manipulation, execution, and reflection. The AI proactively proposes a relevant “what if” scenario (diagnosis based on logs), the student adjusts the parameters via the interface, the simulator executes and visualizes step by step, then the AI guides reflection through metacognitive questions and numerical comparisons. The loop incorporates educational checkpoints: requests for written justification, automatic mini-quizzes, and suggestions for more complex variants. The goal is to alternate between action and metareflection to consolidate concepts and avoid superficial learning.

Application example: quicksort

For quicksort, the interface offers sliders for size, distribution, and pivot strategy; the LLM generates three comparative value sorting scenarios (random pivot, median pivot, nearly sorted array), the simulator executes instrumented runs and displays metrics and animations (swap comparison, recursion depth, complexity). Students are invited to formulate a hypothesis before execution, then explain the differences observed in terms of invariants and amortized complexity. This case illustrates how the framework transforms a passive response into an exploratory, verifiable, and reflective sequence, while providing artifacts that can be used by the teacher (logs, frequent errors, learning trajectories).

Prospective outlook on the implementation of the “what-if” concept in university computer science education

This chapter offers a forward-looking and pragmatic vision of the deployment of the What-If Learning UX framework in university courses on algorithms and data structures. It outlines a realistic trajectory in several phases, details the methods of pedagogical integration, specifies the technical and engineering requirements, addresses ethical and equity issues, defines a rigorous empirical evaluation protocol, illustrates concrete usage scenarios at different scales, and concludes with an operational roadmap. The goal is to transform an innovative concept into robust, reproducible teaching tools that are ethically sound and acceptable to all stakeholders (students, teachers, training managers, institutional administrators). This chapter offers a forward-looking and pragmatic vision of the deployment of the What-If Learning UX framework in university courses on algorithms and data structures. It outlines a realistic, multi-phase trajectory, details the pedagogical integration methods, specifies the technical and engineering requirements, addresses ethical and equity issues, defines a rigorous empirical evaluation protocol, illustrates concrete usage scenarios at different scales, and concludes with an operational roadmap. The goal is to transform an innovative concept into robust, reproducible teaching tools that are ethically sound and acceptable to all stakeholders (students, teachers, training managers, institutional administrators).

Deployment path: phases, objectives, and indicators

The deployment of a tool such as What-If Learning UX cannot be instantaneous: it requires gradual progress to validate hypotheses, adjust technical and educational choices, and accumulate evidence of effectiveness before any generalization. A trajectory in three distinct phases minimizes risks, optimizes resources, and builds gradual buy-in from stakeholders.

- Pilot phase (6-12 months) This initial phase focuses on controlled and limited experiments, ideally on one or two flagship modules (advanced sorting, balanced trees, shortest path algorithms) and with one or two small groups of students. The primary objectives are to validate the ergonomics of the interfaces, measure acceptability by students and teachers, and detect early on the main sources of AI hallucinations or errors. Key indicators include: the voluntary adoption rate (percentage of students using the tool regularly), the average number of what-if explorations per student per session, the frequency of discrepancies between LLM predictions and actual instrumented executions, satisfaction scores (notably via the System Usability Scale - SUS), and qualitative feedback collected from teachers (interviews or open-ended questionnaires).

- Extension phase (12-24 months) Once the lessons learned from the pilot phase have been integrated, deployment is extended to all algorithm courses in a department or program. The tool is integrated as a plugin into the institutional LMS via the LTI[1] standard, and the module library is expanded (graphs, advanced structures, concurrent algorithms, etc.). The objectives are to evaluate the pedagogical impact on larger cohorts and to refine prompts, interaction patterns, and feedback mechanisms. Indicators focus on pre/post gains on standardized conceptual tests, student retention rates in the modules concerned, and measurable reductions in recurring conceptual errors (via analysis of logs and student work).
- System integration phase (24-48 months) At this stage, the system is intended for adoption by individual departments or across universities. Multimodal features (voice control, augmented/virtual reality for immersive visualizations) are being explored, and institutional policies are being formalized (rules for use in assessment, mandatory "AI-free" modes, etc.). The objectives are to institutionalize best practices, ensure sustainable maintenance, and establish shared governance of models and data. Success indicators include institutional adoption rates, explicit integration into learning frameworks (official programs, training models), and regular independent ethical audits.

Pedagogical integration: usage scenarios, activity design, and the role of teachers

The pedagogical effectiveness of What-If Learning UX relies on its seamless integration into existing practices, rather than abruptly replacing traditional methods. The aim is to design activities that fully exploit the tool's affordances (interactive exploration, immediate feedback, personalization) while preserving the essential dimensions of active learning: reflection, justification, transfer, and metacognition.

Educational use scenarios. The tool supports several complementary modalities:

- ***Guided learning:*** Students freely explore what-if scenarios to build conceptual intuition (e.g., impact of pivot choice on quicksort).
- ***Targeted remediation:*** Teachers use interaction logs to identify poorly mastered concepts and propose personalized, tailored learning paths.
- ***Formative assessment:*** Adaptive mini quizzes are generated based on recent explorations, with immediate feedback and longitudinal monitoring.
- ***Augmented tutorials:*** In person, the AI proposes interactive challenges; the teacher leads a group discussion based on the divergent results observed.

Activity design Each activity must be explicitly aligned with specific cognitive objectives (developing intuition, constructing evidence, promoting transfer). The expected artifacts are clearly defined: written justification of a hypothesis, saving a relevant simulation, numerical comparison of scenarios, etc. A systematic alternation between interactive manipulation and metareflection is recommended: students formulate a written hypothesis before any interaction, run the simulation, compare the observed results with their predictions, and write a critical summary. To prevent excessive cognitive delegation, the instructions include deliberate constraints (e.g., requiring a prior hypothesis, limiting access to the complete solution during assessed exercises).

Role and training of teachers Teachers become essential co-designers and mediators: they define educational prompts, validate usage scenarios, interpret interaction logs, design personalized remediation, and adjust assessment methods. Initial training (prompt engineering adapted to the educational context, interpretation of engagement metrics, ethics of AI use) is essential. Co-design workshops involving teachers and students improve the discoverability of the interface, overall acceptability, and suitability for real needs.

Technical requirements and operational architecture

The long-term success of the framework depends on a robust, secure, interoperable, and economically viable technical architecture. It must guarantee the reliability of responses, minimize the risk of hallucinations, ensure traceability, and allow for progressive scalability.

Modular and secure architecture: The system is organized into three distinct layers:

- LLM orchestration (controlled generation via structured prompt templates),
- Instrumented execution engine (dedicated simulator, sandboxed real executions, automatic checks),
- UX/LMS layer (responsive user interface, dynamic visualizations, teacher dashboards). This separation allows LLM predictions to be systematically cross-checked with actual executions, drastically reducing hallucinations and ensuring traceability and reproducibility.

Interoperability and LMS integration via open standards (LTI 1.3, SCORM) facilitates deployment in Moodle, Canvas, or other platforms. REST APIs expose essential features: scenario generation, simulation execution, log retrieval, dashboard display. Front-end components (React or Vue.js) must be responsive and comply with WCAG 2.1+ accessibility criteria.

Verifiability and hybridization. For critical content (formal proofs, algorithmic invariants), hybridization is recommended: combining LLM with formal tools (theorem provers such as Lean or Coq, symbolic execution). Discrepancies between LLM prediction and actual execution trigger automated audit workflows (flagging for human review, request for teacher intervention).

Scalability and costs. A cloud-native architecture (Docker containers, Kubernetes orchestration) with autoscaling manages usage peaks (lab sessions, exams). To control costs, powerful remote models are combined for initial generation and local/open-source models (e.g., Llama, Mistral) for repetitive tasks. Quotas per user/institution limit abuse.

Security and confidentiality. All code executions are sandboxed (strict isolation), and user inputs are filtered to prevent injections. Logs are encrypted, training data is anonymized, and the system complies with the GDPR (opt-in consent, right to export, minimization of collected data).

Ethical issues, fairness, and governance

The integration of generative AI tools into higher education raises structural ethical questions that cannot be relegated to secondary consideration. Proactive governance is essential to manage the risks of over-reliance, bias, unequal access, and accountability.

The main risk is over-reliance and loss of cognitive autonomy: instant access to explanations or visualizations can encourage excessive delegation of reasoning. Mitigation strategies include: requiring explicit assumptions before interaction, restricting access to complete solutions in assessment, and implementing decreasing scaffolding[2] modes (assistance that gradually diminishes).

The biases and representativeness of LLM responses (often reflecting dominant paradigms or majority cultural examples) require regular audits, diversification of training corpora (inclusion of varied languages and paradigms), and the ability for teachers to correct or supplement outputs in real time.

Equitable access requires lightweight, mobile-compatible versions that work partially offline, and institutional measures (funding, equipment loans) for students who are digitally disadvantaged.

Finally, transparency and accountability require clear information about the limitations of AI (risk of error, lack of real understanding, reduction in knowledge or critical thinking), secure storage of interaction records, explicit definition of responsibilities (validation, correction), and the establishment of local ethics committees to oversee deployments.

Empirical evaluation protocol and impact indicators

To move from a promising intuition to scientific validation, a rigorous empirical evaluation protocol is essential. It combines controlled experimental design, mixed measures (quantitative and qualitative) and complementary analyses to measure the real impact on learning and identify the conditions for effectiveness.

The recommended experimental design is based on randomised controlled studies over one semester: control groups (traditional teaching + static resources) vs. intervention groups (integrated What-If UX). Repeated measurements: pre-test, immediate post-test, retention test at 3-6 months.

Quantitative measures include standardised conceptual tests (complexity, invariants), transfer tasks on new problems, engagement metrics (number of explorations, active time), quality of hypotheses (predictive accuracy), and rate of detected hallucinations (LLM divergence vs. execution).

Qualitative measures are based on semi-structured interviews, analysis of written justifications (metacognitive rubrics), satisfaction questionnaires (SUS), and perceived usefulness.

Additional analyses use logs to map learning trajectories, cluster usage profiles, and study correlations between exploration and gains. The open publication of protocols and anonymised data promotes reproducibility.

Concrete usage scenarios and scale cases

To illustrate practical implementation, here are three representative scenarios at different scales, showing how the tool can be integrated into everyday teaching.

Use in augmented tutorials: During the session, the teacher launches a quicksort challenge. Students, in small groups, manipulate sliders to test different pivots, formulate hypotheses, and observe the visualisations. The teacher leads the discussion based on the differences (e.g., worst-case degeneration) and guides the students towards a formal proof or complexity analysis.

Asynchronous remediation: Via the LMS, a student in difficulty receives a personalised course (three targeted what-if scenarios, justification exercises, mini-quizzes). Logs trigger a teacher alert if progress stagnates (few varied hypotheses, repetition of errors).

Hybrid formative assessment: Exams combine 'non-AI' tasks (closed coding) and 'AI' tasks where students document their exploratory process (hypotheses, captures, conclusions). AI provides immediate formative feedback, but grading remains human.

Operational roadmap and final recommendations

Implementation requires realistic and iterative planning.

Immediate priorities (0-12 months): Co-design with teachers, minimum viable prototype (quicksort), thorough security and verifiability testing.

Medium term (12-36 months): LMS integration, extension to other key modules, ongoing teacher training.

Long term (36 months+): Standardisation of practices, regular ethical audits, integration into accreditation frameworks (CTI, HCERES).

In conclusion, the thoughtful deployment of What-If Learning UX can profoundly enrich the teaching of algorithms, provided that student autonomy is placed at the centre, fairness is guaranteed, continuous empirical evaluation is relied upon, and shared governance is established. This interdisciplinary approach (computer science, didactics, ethics) paves the way for more active, inclusive and in-depth learning.

Key recommendations. (1) Adopt an iterative and human-centred approach; (2) combine LLM and verifiable executions to limit hallucinations; (3) formalise usage policies and 'AI-free' modes for assessment; (4) invest in teacher training; (5) systematically measure the educational impact and publish the results.

In summary, the implementation of the What If Learning UX concept is technically and pedagogically feasible, but it requires a coordinated approach: rigorous prototyping, co-design with teachers, verification mechanisms and ethical governance. If properly conducted, this trajectory can transform reactive LLM tutors into proactive, manipulable learning environments focused on the development of deep algorithmic reasoning.

Methodology for back-propagating problems in the what-if approach to AI-assisted teaching

Transposing the 'what-if' approach from UX design to generative AI-assisted teaching encourages students to generate multiple hypothetical scenarios to find creative solutions. For instance, a university computer science student might ask, "What if we altered the data structure of this algorithm?" This often results in a branching decision tree, where some branches are promising, some require corrections for logical or factual inconsistencies, and some must be abandoned due to inefficiency or impracticability. These failures are not dead ends, however; they represent educational opportunities. To exploit these opportunities, we propose a problem back-propagation methodology inspired by backpropagation in machine learning (Rumelhart, Hinton & Williams, 1986). This technique propagates errors from the output to the inputs in order to adjust the parameters and is adapted here to promote metacognitive understanding. This methodology enables students to identify the root causes of errors, rectify them, and apply the lessons learned to future contexts, whether similar or distinct. As LLMs evolve towards greater robustness, with features such as self-verification being integrated into models like Claude 3.5 and GPT-4o, this methodology will become increasingly relevant in fields such as computer science, where counterfactual reasoning is paramount. It aligns with theories of learning from error (Metcalfe, 2017), whereby missteps strengthen memory and cognitive resilience. It also aligns with recent studies in AIED showing that iterative feedback loops can improve retention by 15-35% (Koedinger et al., 2024).

The methodology comprises five iterative steps and is designed for a hybrid human-AI environment in which AI (e.g. a LangChain-based tutor) provides assistance without taking control. It can be implemented via platforms such as Moodle with AI extensions or dedicated tools such as Streamlit, which can be used to visualise 'what if' trees.

Step 1: Identify local errors. This phase begins with the proactive detection of errors at branch and sub-variant level. Guided by AI, the student examines the generated outputs, which may include hallucinations (factually incorrect answers), inconsistencies (e.g. violations of invariants in algorithms) and unexpected results (e.g. a what-if simulation predicting O(1) complexity for an NP-complete problem). Automated tools facilitate this process. These include AI verification prompts ('Analyse this branch for potential errors') and interactive logs that capture metrics (e.g. a consistency score via cosine similarity between the AI response and verified references). In the context of teaching algorithms, if a branch such as 'What if we used a heap instead of an array?' leads to an overflow error, the student can flag it via a simple interface ('Error detected' button). This early identification of errors prevents them from accumulating and encourages active vigilance, in line with the principles of self-regulation in learning (Zimmerman, 2000).

Step 2: Backward propagation. Once the error has been identified, it is fed back to the initial assumptions, mimicking the process of backpropagation where gradients adjust the previous layers. The student asks causal questions such as: 'Which initial assumption caused this inconsistency?' The AI provides assistance in the form of thought-prompting chains (e.g. 'Retrace the logical chain step by step and isolate the source'). This transforms the 'what if' tree into a dynamic graph where local errors impact parent nodes. For instance, if a sub-branch of biology (e.g. 'What if we modified a gene?') produces an incorrect prediction using BioPython, which is integrated into the AI, the propagation process reveals a bias in the initial hypothesis (e.g. ignorance of epigenetic effects). Visualisations (e.g. D3.js diagrams) illustrate this process, helping students to understand how a local error affects the whole and encouraging them to take a more systemic view.

Step 3: Correction and adjustment. This step involves refining the branches through iterations. The student rephrases the flawed 'what if' scenarios by applying the lessons learned (e.g. by adding constraints such as 'What if we took degenerate edge cases into account?'). The AI provides immediate feedback, validating the adjustments through revised simulations. For a branch to be abandoned, it is labelled 'Abandoned for violation of physical constraints - lesson: check theoretical feasibility upstream'. This approach is inspired by the idea of learning from error, whereby active corrections reinforce neural connections (Metcalfe, 2017). In an academic setting, a group of students could collaborate using a shared tool where the AI suggests corrected variants, thereby increasing collective engagement.

Step 4: Improving overall understanding. Backward propagation culminates in metacognitive synthesis, whereby students connect errors to broader concepts. Through reflective journals or AI prompts (e.g. 'Summarise how this error illustrates a fundamental principle'), they can identify recurring patterns (e.g. underestimating systemic interactions). Research in AI education indicates that such mechanisms enhance deep understanding, resulting in measurable improvements in post-test scores (Koedinger et al., 2024). For instance, an error in an algorithmic branch might reveal a knowledge gap in complexity theory, prompting an AI-guided review of the relevant concepts.

Step 5: Transferring knowledge to avoid future errors. Finally, lessons are generalised to similar (e.g. another sorting algorithm) or dissimilar (e.g. application to AI ethics) situations. AI simulates cross-scenarios (e.g. 'Apply this correction to a problem in economics'), promoting transfer learning (Perkins & Salomon, 1992). Post-hoc workshops with peer reviews reinforce this process, helping students to internalise strategies such as 'Always challenge initial assumptions with extreme cases'.

Despite its strengths, this methodology presents two challenges: the need for reliable AI to avoid error amplification and the requirement for teacher training. Prospects include integrating multimodality (e.g. using VR to visualise propagations) and conducting empirical assessments to quantify impacts. In conclusion, backpropagation transforms the limitations of hypothetical scenarios into a powerful educational tool, preparing students for resilient and adaptable learning in an AI-dominated world.

Discussion and prospects

This chapter summarises the contributions of the What If Learning UX framework, discusses its methodological and technical limitations, identifies priority areas for research, and proposes operational recommendations for research and practice. The aim is to situate the conceptual contribution within a broader research horizon, anticipate obstacles to adoption, and open up concrete directions for validating, extending, and regulating the use of AI-assisted exploratory learning devices in computer science.

Theoretical and pedagogical contributions

The proposed framework makes three major theoretical contributions. First, it formalises the integration of counterfactual reasoning into the teaching of algorithms by defining levels of pedagogical hypotheses that link simple empirical variations to theoretical counterfactuals. This granularity facilitates the design of progressive activities ranging from experimental intuition to metatheoretical reflection. Second, it explicitly articulates UX principles (manipulability, discoverability, immediate feedback) with prompt engineering patterns, creating an operational bridge between interface design and controlled linguistic generation. Third, it proposes a modular architecture that combines LLM generation and verifiable execution, offering a pragmatic compromise between generative creativity and technical rigour. Pedagogically, these contributions aim to transform reactive interactions into proactive exploration loops, promoting metacognition, the ability to formulate hypotheses, and conceptual robustness in the face of degenerate cases.

Methodological and technical limitations

Several limitations must be recognised and addressed. Empirical validity limitation: the framework is conceptual and requires rigorous empirical evaluations on varied cohorts to establish its actual effectiveness on comprehension, transfer and retention. LLM-related limitation: dependence on generative models exposes the system to hallucinations and inaccuracies, particularly for formal proofs or complex analyses; the proposed hybridisation strategy (instrumented execution, automatic checks, integration of provers) reduces but does not eliminate this risk. Scale and cost limitation: instrumented simulations and large-scale LLM calls can generate significant costs and infrastructure constraints for institutions. Pedagogical limitations: without careful didactic design, the system may encourage superficial learning if students simply observe the simulations without producing explanations or arguments. Finally, ethical limitations: risks of over-reliance, bias in responses and inequalities of access require institutional safeguards.

Priority areas for research

To transform the conceptual framework into a validated and robust system, several areas of research are a priority.

Axis 1: Longitudinal empirical validation Conduct randomised controlled studies over several semesters and institutions to measure the impact on conceptual understanding, transfer and retention. These studies should include behavioural (exploration logs), cognitive (conceptual tests) and qualitative (interviews, self-explanation analyses) measures.

Axis 2: Verifiability and formal hybridisation Develop methods for integrating LLM and formal tools (symbolic execution, theorem provers) to provide partial proofs or automatic checks. Research 'double validation' protocols where generative output is systematically compared with instrumented execution.

Axis 3: Learning-centred UX design Study how interface elements (sliders, animations, comparators) influence the quality of hypotheses formulated and the depth of reflection. Experiment with discoverability and scaffolding variants to identify configurations that maximise engagement without encouraging delegation.

Axis 4: Personalisation and pedagogical adaptability Explore adaptation algorithms that automatically select relevant what-if scenarios based on error profiles and learning trajectories, while ensuring transparency and teacher control.

Axis 5: Governance, ethics and equity Develop institutional frameworks for the governance of systems (usage policies, bias audits, 'AI-free' modes for assessment) and study the impacts on equity of access, particularly for students in digitally precarious situations.

Practical recommendations for implementation

Based on the above analyses, several operational recommendations are necessary for teaching and technical teams wishing to deploy the framework.

Recommendation 1. Systematic co-design Involve teachers, students and educational engineers from the design phase onwards to ensure the educational relevance and discoverability of features.

Recommendation 2. Iterative deployment and controlled pilot Start with targeted modules (e.g. quicksort, Dijkstra) and iterate based on mixed (quantitative and qualitative) evaluations before generalising.

Recommendation 3. Integration of verification mechanisms Always cross-check LLM explanations with instrumented executions and clearly display the limitations and confidence levels of responses.

Recommendation 4. Training and resources for teachers Train teachers in educational prompt engineering, log interpretation and ethical issues so that they can fully play their role as mediators.

Recommendation 5. Assessment policies and 'AI-free' modes Define clear rules for the use of AI in summative assessment and offer 'AI-free' alternatives to preserve academic integrity and cognitive resilience.

Expected impacts and long-term vision

If technical and pedagogical challenges are addressed in a coordinated manner, What If Learning UX can have transformative effects. In the short term, we can expect to see improved engagement and a reduction in recurring conceptual errors through the active exploration of edge cases. In the medium term, the systematic integration of counterfactual loops could increase students' ability to transfer knowledge between contexts and think critically about algorithmic assumptions. In the long term, these mechanisms could help redefine the skills expected in computer science — by emphasising exploratory reasoning and the ability to question complex systems — and integrate these skills into accreditation frameworks.

What If Learning UX offers a pragmatic and ambitious path to making AI tutors not only reactive, but proactive and exploratory. Its success will depend on a delicate balance between technological innovation, pedagogical rigour, and ethical governance. Future research and experimentation will need to empirically validate the pedagogical benefits, refine verification mechanisms, and ensure that these tools reinforce, rather than replace, students' ability to think and justify their reasoning. In this sense, what-if is not just a feature: it is an invitation to rethink how AI can augment human algorithmic thinking.

Conclusion

This article presented the WI-LUX framework, which aims to integrate counterfactual reasoning from UX design into intelligent tutors based on large language models for teaching algorithms and data structures. We have formalised a pedagogical granularity into four levels of hypotheses, described a modular architecture combining LLM generation and verifiable execution, proposed prompt patterns and a proactive exploration loop, and outlined implementation and evaluation scenarios in a university context. The main contribution is to transform reactive interactions into manipulable exploratory sequences, promoting hypothesis formulation, comparison of variants, and metacognitive reflection.

The pedagogical implications are twofold: on the one hand, the system promises to improve conceptual understanding and transferability by exposing students to edge cases and algorithmic trade-offs; on the other hand, it redefines the role of the teacher as co-designer and mediator, responsible for the quality of prompts, the interpretation of logs, and the regulation of usage. On a technical level, the combination of LLM + instrumented simulator offers a pragmatic compromise between generative creativity and verifiable rigour, but it requires robust mechanisms for verification, traceability, and cost management.

We recognise the limitations and risks: the need for longitudinal empirical evaluations, vulnerability to model hallucinations, infrastructure costs, and ethical issues related to over-reliance and equity of access. Future research should prioritise experimental validation, hybridisation with formal tools, UX optimisation for discoverability, and the design of institutional policies that guarantee transparency and accountability. Rigorous evaluation protocols and ethical audits will be essential to measure real impact and guide deployment.

In conclusion, the what-if concept applied to AI tutors opens up a promising avenue for refocusing computer science education on exploratory reasoning and critical thinking. Its successful implementation will depend on an iterative, human-centred and interdisciplinary approach, bringing together educators, engineers and institutional decision-makers. If these conditions are met, What If Learning UX can help train computer scientists who are not only capable of writing code, but also of questioning, explaining and reasoning about the systems they design.

References

1. Bruner JS. "The act of discovery". Harvard Educational Review 31.1 (1961): 21-32.
2. Bunay Guisnan P, Lara JA and Romero C. "Counterfactual explanations in education: A systematic review". Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery (2025).
3. Chi MTH., et al. "Self explanations: How students study and use examples in learning to solve problems". Cognitive Science 13.2 (1989): 145-182.
4. Jacobsen LJ and Weber KE. "The promises and pitfalls of large language models as feedback providers: A study of prompt engineering and the quality of AI driven feedback". AI (2025).
5. Jamali H., et al. "AI-powered adaptive learning interfaces: a user experience study in higher education". Front. Comput. Sci 7

(2025).

6. Jamali H., et al. “AI powered adaptive learning interfaces: A user experience study in education platforms”. Frontiers in Computer Science (2025).
7. Kazemitabaar M., et al. “CodeAid: Evaluating a classroom deployment of an LLM based programming assistant that balances student and educator needs”. ACM (2024).
8. Kolb DA. “Experiential learning: Experience as the source of learning and development”. Englewood Cliffs, NJ : Prentice-Hall (1984).
9. Kulik JA and Fletcher JD. “Effectiveness of intelligent tutoring systems: A meta analytic review”. Review of Educational Research 86.1 (2016): 42-78.
10. Metcalfe J. “Learning from errors”. Annual Review of Psychology 68 (2017): 465-489.
11. Munoz A. “Human-centered AI for student engagement and academic efficiency”. Issues in Information Systems 26.1 (2025): 324-337.
12. Nguyen H, Stott N and Allan V. “Comparing feedback from large language models and instructors: Teaching computer science at scale”. Proceedings of the Eleventh ACM Conference on Learning @ Scale (2024): 335-339.
13. Perkins DN and Salomon G. “Transfer of learning”. In International Encyclopedia of Education 2nd ed 2 (1992): 6452-6457.
14. Pinkwart N. “Another 25 years of AIED? Challenges and opportunities for intelligent educational technologies of the future”. International Journal of Artificial Intelligence in Education 26 (2016): 771-783.
15. Prather J., et al. “Leveraging LLMs for Personalized Support in Programming Projects”. Proceedings of the ACM on Human-Computer Interaction (2025).
16. Quan X., et al. “Verification and refinement of natural language explanations through LLM-symbolic theorem proving”. arXiv preprint arXiv:2405.01379 (2024).
17. Rumelhart DE, Hinton GE and Williams RJ. “Learning representations by back-propagating errors”. Nature 323.6088 (1986): 533-536.
18. Song P, Yang K and Anandkumar A. “Lean Copilot: Large language models as copilots for theorem proving in Lean”. arXiv (2024).
19. Strielkowski W., et al. “AI-driven adaptive learning for sustainable educational transformation”. Sustain. Dev 33 (2025): 1921-1947.
20. Topali P., et al. “Pedagogical considerations in the automation era: A systematic literature review of AIEd in K-12 authentic settings” (2025).
21. Topali P., et al. “Designing human centered learning analytics and artificial intelligence in education solutions: A systematic literature review”. Behaviour & Information Technology (2025).
22. VanLehn K, Jones RM and Chi MTH. “A model of the self explanation effect”. The Journal of the Learning Sciences 2.1 (1992): 1-59.
23. Zimmerman BJ. “Attaining self-regulation: A social cognitive perspective”. In M. Boekaerts, P. R. Pintrich, & M. Zeidner (Eds.), Handbook of Self-Regulation. Academic Press (2000): 13-39.

End Notes

1. The LTI standard enables secure information exchange between your LMS and an external learning tool: https://moodle.com/fr/nouvelles/quest-ce-que-lti-et-comment-il-peut-ameliorer-votre-ecosysteme-dapprentissage/.
2. Scaffolding involves teaching a concept in a different subject, such as teaching math in English, as is done in European classrooms.