

Implementing Concept Instruction via MCP Server

Anderson, T. A.

AI-powered learning

Concept teaching

critical attributes

Educational Technology

Instructional Design

Large Language Models

learning engineering

Model Context Protocol (MCP)

This applied learning engineering research demonstrates the operationalization of Merrill & Tennyson's (1977) concept teaching framework through AI-powered tools. An MCP (Model Context Protocol) server was developed implementing five sequential tools that decompose concept lesson creation into theory-aligned stages: concept definition, attribute analysis, example generation, practice activity creation, and lesson publication. The system encodes M&T's research-based prescriptions as structured prompts, validation algorithms, and schema-driven workflows. Key findings reveal structural constraints with input/output schemas are essential for maintaining theoretical fidelity—unconstrained LLM generation produced inconsistent, theoretically misaligned outputs despite natural language descriptions of instructional principles. The architecture combining LLM semantic capabilities with seven algorithmic validators suggests a promising pattern for educational AI: leverage LLMs for content generation

while enforcing learning science principles through external validation. Future work will embed these concept lessons within problem-based learning environments for just-in-time instruction.

Introduction

The proliferation of AI code generation tools has fundamentally disrupted traditional assessments in programming education—when students can prompt ChatGPT or GitHub Copilot to produce syntactically correct code, the ability to write code no longer serves as a reliable proxy for conceptual understanding. This shift elevates the importance of concept learning: educators must now focus on whether learners truly comprehend “why” code works, not merely whether they can produce it. Merrill and Tennyson’s seminal work “Teaching Concepts: An Instructional Design Guide” (1977) synthesized six empirical studies into prescriptive principles for concept instruction. Despite years of research establishing clear instructional prescriptions, these principles have remained challenging to systematically implement at scale in contemporary digital learning environments.

The emergence of large language models (LLMs) and the Model Context Protocol (MCP) presents both opportunities and challenges for learning engineering. While LLMs demonstrate impressive capacity for semantic analysis and code generation, their application to instructional design often produces inconsistent results when left unconstrained. As Merrill et al. (1996) emphasize, merely generating content about concepts does not constitute pedagogically sound instruction. Efforts to use AI for generating “online textbooks” frequently conflate information delivery with effective concept teaching, ignoring research-based prescriptions about example selection, attribute identification, and boundary case representation.

Recent developments in structured AI tool orchestration through MCP suggest a novel approach: encoding Merrill & Tennyson’s research-based prescriptions as formal validation algorithms, structured prompts, and composable tools that leverage LLM capabilities while maintaining theoretical fidelity. This approach imposes external constraints on LLM outputs through input/output schemas and automated quality checking against M&T’s empirically derived instructional principles.

Methods

An MCP server was developed as a web application, implementing five sequential tools that decompose concept lesson creation into discrete, theory-aligned stages:

Define concept	Analyze concept	Generate examples	Create practice activities	Publish lesson
----------------	-----------------	-------------------	----------------------------	----------------

The system operates in two modes to support different deployment contexts: Prompt Mode returns structured prompts for execution by Claude Desktop’s selected model. LLM Mode makes direct API calls to Ollama (local model) or OpenAI (remote) for autonomous operation.

This hybrid architecture was essential for practical deployment, as early attempts to allow unconstrained LLM generation produced highly inconsistent outputs. The structured input/output schemas enforced by MCP tools proved critical for

maintaining theoretical fidelity. Merrill & Tennyson's research-derived prescriptions were operationalized through three complementary mechanisms:

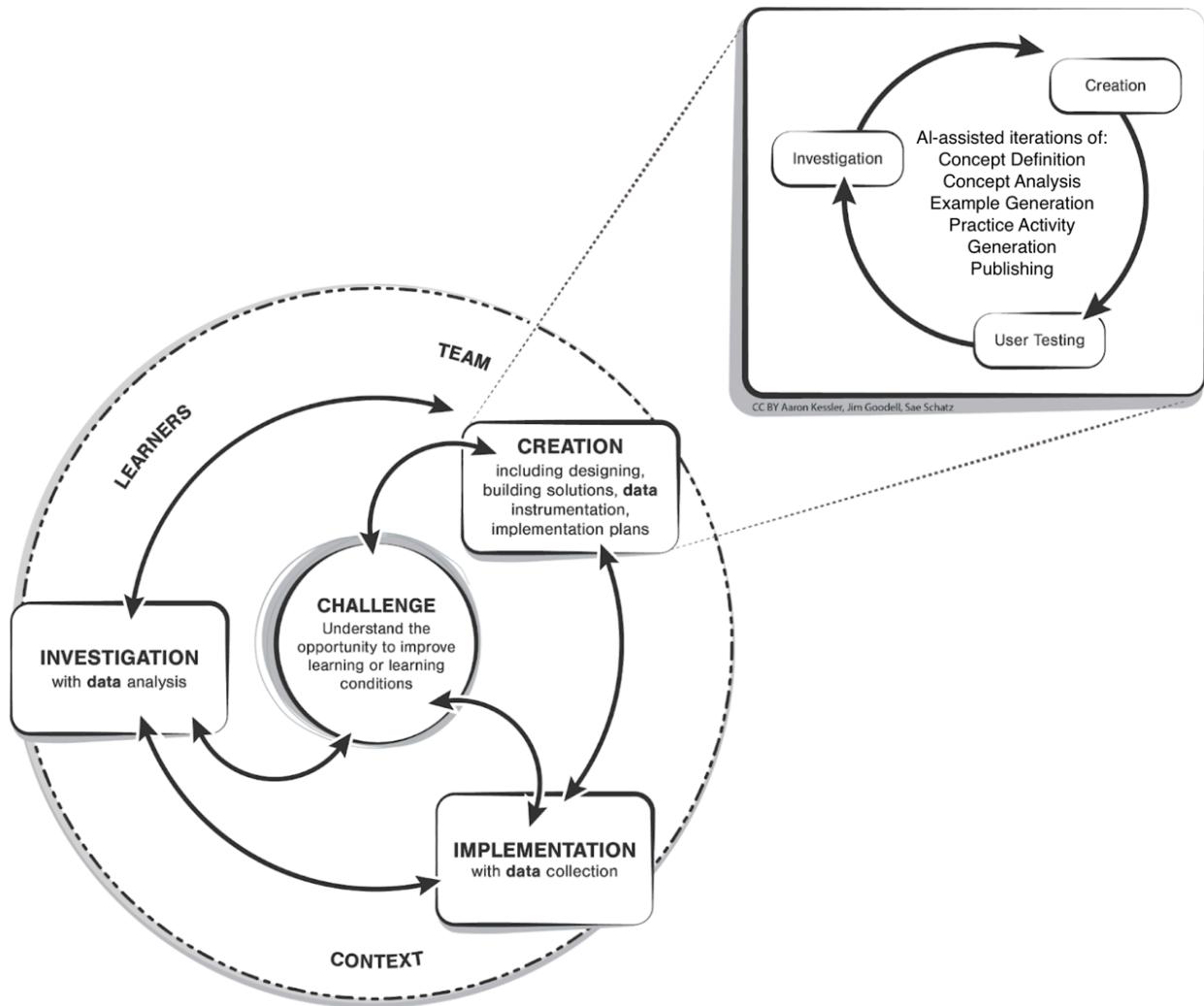
1. Structured System Prompts. Each tool receives domain-specific prompts encoding M&T principles as explicit constraints.
2. Validation Algorithms. Algorithmic validators ensure outputs satisfy theoretical requirements: they verify that all positive examples contain ALL critical attributes and that negative examples are missing at least one. Diversity scores (0-1) are computed measuring variation on non-critical features to prevent stereotype bias. Balanced example ratios (approximately 50-70% positive examples to negative examples) are computed. "Near-miss" negative examples (missing exactly one critical attribute) are validated. Theory-based teaching warnings (overgeneralization risks, misconceptions, boundary ambiguities) are generated. Quality scores with actionable improvement recommendations are provided and insufficient variable attribute variations are flagged.
3. Schema-Driven Outputs. Zod (a TypeScript validation library) schemas enforce structured data flow between tools to prevent LLMs from inventing their own interpretation of what constitutes "good" concept instruction.

This implementation follows a nested Learning Engineering approach (Craig et al., 2025), where the concept lesson creation and validation tools iterate on the production of concept lesson content inside the broader instructional design cycle as shown in Figure 1. It applies the IEEE ICICLE definition of Learning Engineering by coupling human-centered design with data-informed, iterative improvement (Goodell et al., 2022).

Figure 1.

Learning Engineering Process With Adapted Depiction of Nested Creation Phase.

Note. Adapted from Craig et al. (2025). CC BY.



The AI-assisted algorithmic content creation and validation automate the nested cycle portion of the Learning Engineering process. This greatly accelerates the creation-phase iteration by shifting portions of the iteration that typically require expert human judgement from creation to review and testing. Each MCP tool run challenges the LLM's outputs against Merrill & Tennyson constraints, applies structured fixes via schemas, and investigates compliance before passing results downstream. This operationalizes Baker et al.'s (2022) call for computational methods that enable faster experimentation and iteration, turning learning-science prescriptions into standardized, reusable components. This greatly accelerated experimentation and iteration allowed the author to produce a set of three online courses for the current semester. Each course is structured to have seven "sprint" sessions in the semester with each sprint having between 33 to 36 topics. Each sprint tutorial is linked to concept lessons which required the generation of 281 concept lessons (each lesson containing approximately 8,000 lines of HTML) within an approximately 4-week time span; a task that is clearly only possible with the help of generative AI and agentic tooling.

Results

The system successfully operationalized Merrill & Tennyson's key instructional prescriptions into executable code and structured prompts that achieve critical attribute coverage, example diversity, example balance, boundary representation, and tool chain coherence.

Given the system's current development stage as a practitioner-focused tool, evaluation focused on:

1. Architectural validation: Verification that M&T principles are correctly encoded in prompts, schemas, and validation algorithms.
2. Tool chain coherence: Testing that outputs from one tool properly feed into subsequent tools (e.g., 'define concept' → 'analyze concept' → 'generate examples').
3. Practical usability: Observation of lesson generation workflows in Claude Desktop, including error handling and iterative refinement.
4. Code quality review: Manual inspection of generated code examples for production quality, realistic contexts, and pedagogical clarity.

The focus on architectural fidelity rather than large-scale empirical testing reflects the system's status as an applied learning engineering prototype designed to demonstrate feasibility rather than claim efficacy at scale.

Discussion

This work demonstrates that classical instructional design theory can be systematically encoded into AI tool chains that maintain theoretical fidelity while leveraging LLM semantic capabilities. The success of the hybrid architecture—LLM semantic analysis combined with algorithmic validation—suggests that M&T's principles possess sufficient precision to serve as formal specifications for automated instructional systems.

The most significant finding was that structural constraints are essential for theoretical fidelity. Allowing LLMs to generate lessons from unstructured prompts produced inconsistent and theoretically misaligned outputs. Enforcing strict input/output schemas through MCP tool interfaces and careful wording of system prompts achieved reliable adherence to instructional design principles.

Following Merrill's (2002) "First Principles of Instruction", future work will investigate embedding concept lessons within larger problem-based scenarios. This integration addresses Merrill et al.'s (1996) core argument that effective learning occurs when information supports problem-solving rather than existing as isolated content.

References

Baker, R. S., Boser, U., & Snow, E. L. (2022). Learning engineering: A view on where the field is at, where it's going, and the research needed. *Technology, Mind, and Behavior*, 3(1). <https://doi.org/10.1037/tmb0000058>

Craig, S. D., Avancha, K., Malhotra, P., Gorman, J. C., Verma, V., Likamwa, R., Gary, K., Spain, R., & Goldberg, B. (2025). Using a nested learning engineering methodology to develop a team dynamic measurement framework for a virtual training environment. International Consortium for Innovation and Collaboration in Learning Engineering (ICICLE) 2024 Conference Proceedings: Solving for Complexity at Scale. <https://doi.org/10.59668/2109.21735>

Goodell, J., Kolodner, J., & Kessler, A. (2022). Learning engineering applies the learning sciences. In J. Goodell & J. Kolodner (Eds.), *Learning engineering toolkit: Evidence-based practices from the learning sciences, instructional design, and beyond* (pp. 47-82). Routledge.

Merrill, M. D. (2002). First principles of instruction. *Educational Technology Research and Development*, 50(3), 43-59.

Merrill, M. D., Drake, L., Lacy, M. J., Pratt, J., & the ID2 Research Group. (1996). Reclaiming instructional design. *Educational Technology*, 36(5), 5-7.

Merrill, M. D., & Tennyson, R. D. (1977). *Teaching concepts: An instructional design guide*. Educational Technology Publications.

