

DOI: [https://doi.org/10.48009/4\\_iis\\_2026\\_106](https://doi.org/10.48009/4_iis_2026_106)

## The development of a Socratic facilitation layer with generative AI in IS database education

Katherine Carl, *Governors State University*, [kcarl@govst.edu](mailto:kcarl@govst.edu)

### Abstract

This design-stage paper presents the design of a Socratic facilitation layer for IS database education and a planned proposed exploratory pilot study. Students in IS courses frequently reproduce procedures without understanding the underlying logic, exhibiting procedural fluency without conceptual depth (Anderson & Krathwohl, 2001). The arrival of generative AI tools amplifies this risk, as students can now produce technically correct outputs without conceptual reasoning. This paper proposes an alternative role for generative AI in IS education, as a Socratic facilitation layer rather than a content delivery tool. The paper describes a designed scaffolded inquiry exercise in entity relationship modeling, deployed within a faculty-authored web tool that loads an instructor-designed system prompt and constrains the AI to respond only with gentle confirmation and follow-up questions. The student is positioned as the reasoning agent and the AI is positioned as the questioning instrument. The paper presents the instructional architecture, a six-step pedagogical algorithm for ERD construction, design principles, and an exploratory pilot plan for analyzing student-AI interaction transcripts as both course deliverable and research instrument. Contributions include a design artifact, a set of design principles for AI-facilitated IS pedagogy, and a framework for distinguishing tool-as-answer-source from tool-as-facilitation-layer in generative AI educational deployment.

**Keywords:** generative AI, Socratic method, conceptual knowledge, entity relationship modeling, IS education, design science research

### Introduction

A graduate student brings a completed assignment with them to their instructor's office hours and explains that they have used a generative AI tool embedded in a diagramming platform to produce an entity relationship diagram. Upon inspection, the diagram is technically correct. The student, however, cannot explain why ENROLLMENT must be its own entity rather than an attribute of STUDENT, nor articulate what the cardinality notation means about the underlying business rules. They have produced the diagram, but they cannot reason from it. This is the central problem this paper addresses. The AI tool helped the student complete the assessment and successfully generated a desired deliverable, but in doing so it bypassed the conceptual reasoning the assessment was designed to develop. The student has explicit procedural output without the conceptual understanding to evaluate, defend, or transfer it (Anderson & Krathwohl, 2001). This is a routine occurrence in IS courses where generative AI tools are now embedded in the diagramming environments students use.

Much of the existing literature on generative AI in education treats AI as a content delivery tool, a study aid, or a productivity multiplier, and a smaller but rapidly growing body of recent work has begun to explore

Socratic and dialogic uses of the technology. Where AI is treated as content tool, the pedagogical question becomes how do we teach students to use AI tools well and for what purpose? This paper argues for a different framing. The question is not whether students can use AI tools, but whether they have the conceptual grounding to know what to ask them, how to evaluate their answers, and whether or not the output was, in fact, correct. The instructional task is therefore not to train students to produce outcomes using AI tools, but to cultivate the conceptual understanding that makes the use of such tools consequential.

This paper presents an artifact and a set of design principles for using generative AI in a fundamentally different role, as a Socratic facilitation layer rather than a content delivery tool. The artifact is a faculty-authored web tool that calls the Anthropic Claude API, loaded with an instructor-designed system prompt that constrains the AI to respond only with questions and gentle confirmation. The student opens the tool, encounters the opening prompt, and must reason their way through an entity relationship modeling problem.

The paper makes three contributions. First, it presents the instructional architecture of the tool, including the context engine, the Socratic constraint, and the six-step pedagogical algorithm for ERD construction that the system prompt encodes. Second, it articulates a set of design principles for AI-facilitated IS pedagogy that distinguish tool-as-answer-source from tool-as-facilitation-layer. Third, it presents an exploratory pilot plan for analyzing student-AI interaction transcripts as both course deliverable and research instrument.

## Theoretical Foundations

### Procedural and Conceptual Knowledge in IS Education

The distinction between procedural and conceptual knowledge articulated in the revised Bloom's taxonomy (Anderson & Krathwohl, 2001) and developed extensively in the learning sciences literature provides the conceptual frame for this paper. In database courses specifically, this distinction takes a precise form. A student can follow an algorithm to produce a correct entity relationship diagram using an AI tool without understanding why the algorithm produces correct output. The student has explicit procedural knowledge but lacks the conceptual understanding that would allow them to evaluate their own output, recognize a similar problem in a different domain, or write the SQL query that the diagram implies.

This gap is observable behaviorally. Prior work on game-based behavioral research (Payne et al., 2017; Carl et al., 2021) has demonstrated that procedural execution and conceptual understanding can be operationally distinguished through analysis of student reasoning traces. The present paper extends this approach to Socratic dialogue transcripts. The transcript becomes the artifact in which conceptual movement (or its absence) is rendered visible.

### Scaffolded Socratic Inquiry

The pedagogical framework underlying the designed artifact is scaffolded inquiry, in the tradition of Wood, Bruner, and Ross (1976) on tutoring as the temporary support that enables a learner to operate within their zone of proximal development (Vygotsky, 1978). The student remains the agent of reasoning, and the scaffold—a Socratic AI tutor—provides structure that directs attention, surfaces gaps, and yields as competence develops. This is distinct from problem-based learning (PBL) in its classical formulation (Barrows & Tamblyn, 1980; Hmelo-Silver, 2004), which centers on ill-structured problems, small-group collaboration, and self-directed inquiry. The present design is individual rather than collaborative, structured rather than ill-structured, and short rather than extended. It is more accurately characterized as scaffolded Socratic inquiry than as PBL. A broader pedagogical architecture under development incorporates the social and collaborative dimensions through a peer knowledge-sharing component. The present paper focuses on the individual-cognition layer.

## **The Socratic Method and Its Computational Approximation**

Socratic questioning has been used pedagogically since classical antiquity as both a diagnostic and developmental tool. Well-designed questions surface the boundary of a learner's understanding and create the conditions for new understanding to emerge (Paul & Elder, 2008). The structure of effective Socratic questioning, including the sequence of questions, the timing of redirection, the recognition of conceptual stalling, is itself a form of instructional expertise.

Generative AI tools can approximate Socratic questioning not because they understand the student's reasoning, but because they can be constrained through system prompts to generate questions rather than answers, conditioned on the running dialogue and on a designed pedagogical structure. The acknowledgment matters, as the design rests on prompt-level constraints, not on any claim about the model's comprehension. The pedagogical work is done by the instructor's design of the questioning sequence and the model is the delivery mechanism.

## **Design Science Research**

The paper's contribution is a designed artifact and the design principles extracted from its development, the canonical form of Design Science Research (DSR) (Hevner, March, Park, & Ram, 2004; Peffers, Tuunanen, Rothenberger, & Chatterjee, 2008). The artifact is the Socratic facilitation tool and the design principles are the generalizable claims about how generative AI should be configured to function as a facilitation layer rather than a content delivery tool. Following Gregor and Hevner (2013), this contribution is positioned as Improvement—a new solution for a known problem (the use of generative AI as a content delivery tool that bypasses conceptual reasoning). The present paper covers the first three of Peffers et al.'s (2008) six DSR activities—problem identification and motivation, definition of objectives for a solution, and design and development of the artifact. The remaining three activities—demonstration, evaluation, and communication—are addressed by a planned exploratory pilot described in the Evaluation Plan section of this paper.

## **Related Work**

The pedagogical use of generative AI as a Socratic facilitator rather than as a content delivery tool has emerged as a distinct research direction over the past two years. The present work also engages a longer pre-LLM lineage of intelligent tutoring systems for database education. This section reviews both lines of work and positions the present paper within them.

## **Intelligent Tutoring Systems for Database Modeling**

The most direct domain precursor is KERMIT (Suraweera & Mitrovic, 2004), a constraint-based intelligent tutoring system for entity relationship modeling. KERMIT models the ER domain as an explicit set of constraints derived from Ohlsson's constraint-based modeling framework, evaluates student solutions against those constraints, and generates corrective feedback when a constraint is violated. KERMIT and its successors established that ER modeling is teachable via computational tutoring and demonstrated measurable learning gains in classroom evaluation. The pedagogical model is fundamentally different from the present design, however. KERMIT diagnoses and corrects errors against a predefined constraint set, while the present artifact withholds correction and instead elicits answers through Socratic questioning. KERMIT tells the student what they got wrong, while the present design asks the student follow-up questions to guide them to discovering solutions on their own. The proposed artifact lies within the same domain but the opposite pedagogical model. A second phase of the broader research program will introduce designed error scenarios after the Socratic discovery phase, bringing the two paradigms into a sequenced relationship rather than treating them as alternatives.

## **Design Science Approaches to Socratic AI in Education**

Ma, Chen, Li, and Liu (2024) presented one of the earliest DSR efforts to build a Socratic ChatGPT, proposing a system that employs structured questioning to address the over-reliance problem in AI-assisted learning. The framework was developed at a general-purpose level and evaluated through a field experiment. The shared methodological commitment to DSR provides a direct precedent for the present work, which adopts the same paradigm but applies it within a specific IS pedagogical domain rather than at the architectural level.

## **Role-Engineered and Fine-Tuned Socratic Systems**

Two recent papers represent contrasting implementation paths for Socratic AI. Favero, Pérez-Ortiz, Käser, and Oliver (2024) developed a Socratic chatbot using fine-tuned and prompt-tuned Llama2 models (7B and 13B parameters), deployed locally to address privacy and access concerns. Their evaluation showed that the Socratic tutor outperformed baseline chatbots in fostering reflection and critical thinking. Tufino and Gregorcic (2025) took a different route, using Google's Gemini Gems feature to role-engineer a physics tutor via a detailed instructional script, without model fine-tuning. Both establish the viability of Socratic constraint via instructor-authored configuration. The present paper takes the same instructor-configuration path on a different commercial model (Anthropic Claude), within a different conceptual domain (IS database design), with a different theoretical grounding (procedural-conceptual knowledge), and with a different deliverable structure (the dialogue transcript as submitted assignment).

## **Transcript-Based Analytics in Socratic AI Deployment**

Hashmi and Rebello (2025) deployed a custom Socratic AI chatbot in a large-enrollment introductory mechanics course and analyzed full dialogue transcripts from 150 first-year STEM majors. Their work is the closest methodological neighbor to the present paper. Transcripts are treated as a research instrument that captures the structure and progression of student reasoning, and question specificity is shown to correlate with academic performance. The present paper extends this approach by structurally embedding the transcript as the submitted assignment deliverable rather than collecting it separately for research, eliminating the gap between instructional and research data streams.

## **Prompt Engineering as Instructional Design**

Chukhlomin (2024) documents the use of Socratic prompt engineering in the design of a business course at SUNY Empire State University. The framing centers the educator's role in engineering AI-mediated dialogue and treats prompt design as a form of pedagogical craft. This work is conceptually adjacent to the present paper, with the difference being one of granularity. Chukhlomin focuses on AI-supported course development, while the present paper focuses on a single faculty-authored Socratic exercise structured around a specific conceptual learning objective. A related line of work, Gregorcic, Polverini, and Sarlah (2024), examines ChatGPT as a tool for honing teachers' Socratic dialogue skills—instructor development rather than student instruction.

## **Positioning**

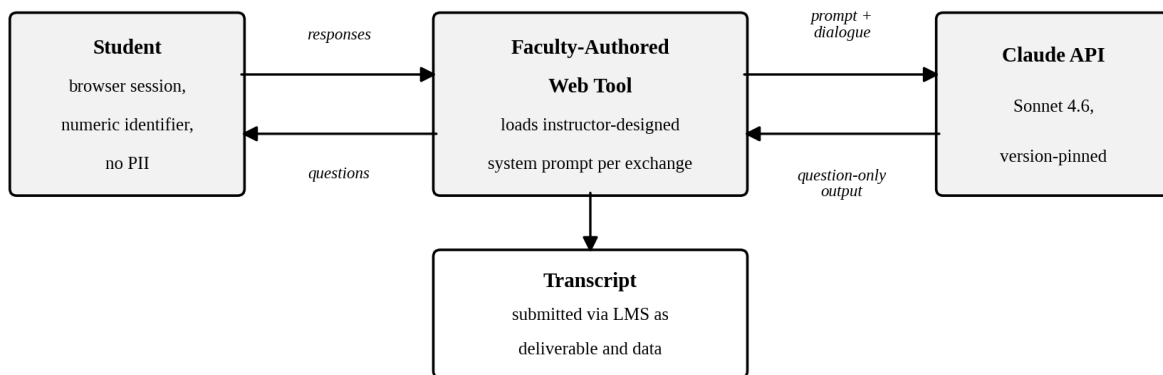
Across these literatures, Socratic facilitation has been implemented in business communication, in general critical thinking instruction, in undergraduate physics, in instructor development, and in pre-LLM intelligent tutoring for database modeling. None of the recent Socratic AI work addresses IS education, and the pre-LLM ITS work in the database domain (KERMIT and its lineage) operates on a correction-based pedagogical model rather than a discovery-based one. The present paper fills a specific gap with a designed artifact, develops a procedural-conceptual framing as theoretical grounding for the design, treats the student transcript as both course deliverable and research instrument, and articulates a pedagogical model (Socratic discovery preceding designed-error correction) distinct from both the recent LLM-based Socratic systems and the older constraint-based ITS lineage.

## The Instructional Architecture

### The Context Engine

The instructional tool operates by loading an instructor-designed system prompt at the start of every student session. The system prompt encodes the scenario (a university database with entities such as STUDENT, COURSE, and PROFESSOR), the business rules that the scenario must satisfy, the six-step pedagogical algorithm for ERD construction (detailed in the Artifact section below), and the Socratic constraint that the AI must respond only with questions and brief confirmations. The Claude API is called with this system prompt loaded for every exchange, ensuring that the Socratic constraint is enforced at the architectural level rather than relying on user-side prompting. Figure 1 presents the architecture.

The design principle is structural in intent, as the AI is configured to respond with questions and to confirm gently when responses are correct, rather than to volunteer answers. Prompt-level constraints cannot guarantee model behavior in all cases, as LLM outputs remain probabilistic, and constraint adherence may vary across model versions, but the system is designed to operate within the constraint by default. The tool calls Claude Sonnet 4.6 (model string `claude-sonnet-4-6`) and the model string is version-pinned for the duration of data collection so that model drift does not introduce uncontrolled variance into the pilot.



**Figure 1. Architecture of the Socratic Facilitation Layer**

### The Student Experience

The student accesses the tool via a direct URL and authenticates using an instructor-assigned numerical identifier. No personally identifying information passes to the tool or to the Anthropic API. The student encounters an opening prompt, responds in natural language, and receives the next question. The exchange continues until the student has worked through the modeling problem to its conclusion.

Two illustrative prompts demonstrate the structure. The opening prompt invites the student to define modeling: “What does the word ‘model’ mean to you? Can you provide an example?” The follow-up prompt invites the student to identify entities from the business scenario before introducing the term entity at all. Throughout, the AI gently confirms correct moves and asks further questions when responses are incomplete or incorrect. The conversation ends when the student has constructed the diagram through their own reasoning.

### The Transcript as Data and Deliverable

Students’ full chat transcripts are submitted as the assignment deliverable. This serves two purposes simultaneously. As an instructional artifact, the transcript demonstrates the student’s reasoning process, including what they understood immediately, where they paused, what required redirection, and what

required multiple iterations. As a research artifact, the same transcript constitutes the primary data for analyzing patterns of conceptual movement under Socratic facilitation.

This dual function eliminates the need for separate data collection mechanisms and ensures that the data are the learning activity rather than an external observation of it. The trade-off—that the data are shaped by the assignment context, including grading stakes, is acknowledged and addressed in the evaluation plan.

## **Guardrails and Academic Integrity**

Several structural features address academic integrity concerns. First, the system prompt enforces topic scope, so that off-topic queries are redirected back to the exercise rather than answered. Second, the Socratic constraint discourages direct answer-extraction. A student who attempts to extract the answer typically receives another question rather than a solution. Third, transcript submission renders evasion visible, as a student who refuses to engage substantively produces a transcript that reveals as much. Finally, circumvention behavior itself becomes data. A student who attempts to extract the answer rather than engage with the questioning reveals something about their perception of the exercise. Characterizing circumvention behavior is theorized as a future research direction.

## **The Artifact: A Designed ERD Exercise**

### **Domain Selection and Rationale**

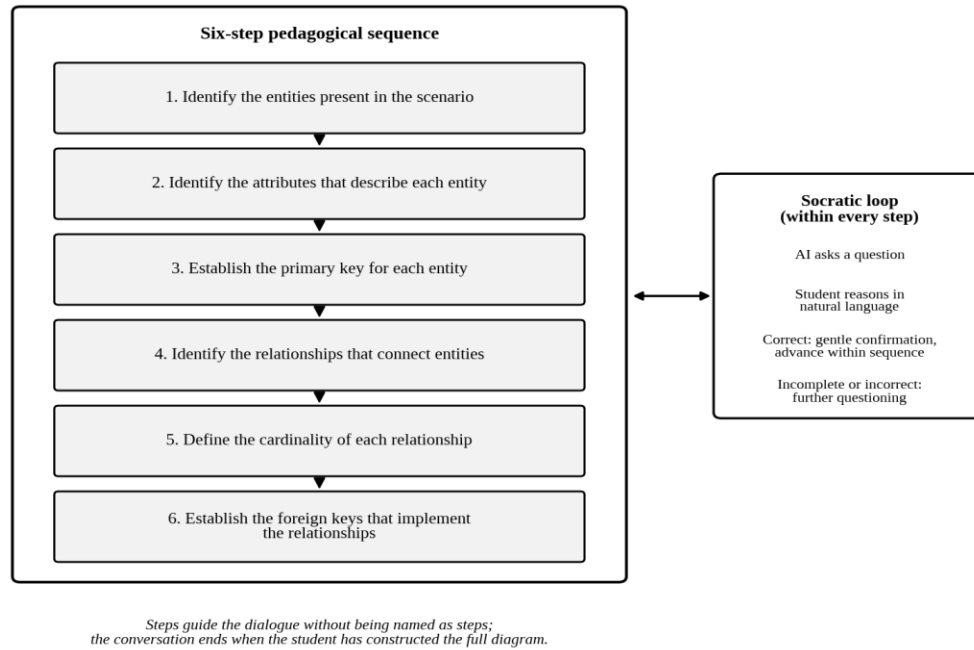
Entity relationship modeling (Chen, 1976) was selected as the primary domain for three reasons. First, it sits precisely at the boundary of procedural and conceptual knowledge. A student can follow the six-step ERD construction algorithm prescribed by the instructor without understanding what each step accomplishes or why. Second, ER modeling errors propagate. A misidentified entity or a misunderstood cardinality constraint produces downstream failures in normalization, query construction, and database implementation. Third, the domain has a well-developed prior literature in intelligent tutoring (Suraweera & Mitrovic, 2004) but has not been addressed within the emerging Socratic AI tradition.

The exercise uses a university scenario (STUDENT, COURSE, PROFESSOR) as the instructional context. This scenario is familiar to every student, requires no domain expertise to engage, and supports rich exploration of cardinality, relationships, and constraint structures while remaining concrete enough to ground reasoning.

### **The Six-Step Pedagogical Algorithm**

The pedagogical structure of the exercise and the substantive instructional contribution of the artifact is a six-step algorithm for ERD construction that the instructor has developed through classroom practice. The Socratic facilitation is organized around guiding the student through these six steps without naming them as steps. Figure 2 presents the flow of the sequence and the Socratic loop that operates within each step. Given a business specification, the student must:

1. Identify the entities present in the scenario.
2. Identify the attributes that describe each entity.
3. Establish the primary key for each entity.
4. Identify the relationships that connect entities.
5. Define the cardinality of each relationship.
6. Establish the foreign keys that implement the relationships.



**Figure 2. Flow of the Six-Step Pedagogical Sequence With the Socratic Loop**

The full system prompt is not reproduced in this paper. The six-step algorithm is the substantive pedagogical structure that the prompt encodes and the prompt itself is the implementation of that structure in natural language. Disclosure of the algorithm makes the design evaluable at the pedagogical level where other researchers can adopt, adapt, or critique the structure, while the verbatim prompt remains instructor intellectual property. This separation between disclosed pedagogical structure and reserved implementation language is consistent with the disclosure conventions in cognate fields where the curricular design, not the verbatim script, is the unit of shareable contribution. Table 1 provides a redacted structural skeleton of the system prompt, disclosing each component and its function while withholding implementation language.

**Table 1. Redacted Structural Skeleton of the System Prompt**

| Prompt component            | Function                                                                                  | Disclosure status                         |
|-----------------------------|-------------------------------------------------------------------------------------------|-------------------------------------------|
| Role and tone definition    | Establishes the AI as questioner; brief, gentle confirmations only                        | Structure disclosed; language withheld    |
| Scenario and business rules | University domain (STUDENT, COURSE, PROFESSOR) and the constraints the model must satisfy | Disclosed in this paper                   |
| Six-step sequence encoding  | Orders the dialogue through the pedagogical algorithm without naming the steps            | Algorithm fully disclosed in this section |
| Socratic constraint         | Question-only responses; no answer generation                                             | Structure disclosed; language withheld    |

|                    |                                                                          |                                        |
|--------------------|--------------------------------------------------------------------------|----------------------------------------|
| Redirection rules  | Returns off-topic queries and answer-extraction attempts to the exercise | Structure disclosed; language withheld |
| Confirmation style | Brief affirmation of correct moves before the next question              | Structure disclosed; language withheld |

## Exercise Walkthrough

The exercise opens with the modeling concept itself. The student is asked what modeling means—not the technical concept, but the everyday one. The intent is to establish that the student is reasoning, not retrieving. The Socratic structure then guides the student through the six-step algorithm, beginning with entity identification and ending with foreign keys, without naming the steps as steps. The AI asks, the student answers, and the AI confirms or asks again.

Each step in the exchange is a diagnostic event. The student who identifies STUDENT and COURSE as entities but cannot articulate why ENROLLMENT must be its own entity has revealed a specific conceptual gap. The AI does not tell them they are wrong, but asks them what would happen if a student enrolled in two courses, or if a course had a limited capacity, and lets the student construct the recognition themselves. The conversation continues until the student has worked the full diagram through their own reasoning.

A short constructed excerpt illustrates the diagnostic structure. The exchange below is synthetic but representative of the move from procedural identification to conceptual recognition. Coding annotations follow the excerpt.

*AI: What entities can you identify in this scenario?*

*Student: STUDENT and COURSE.*

*AI: That's a good place to start. What kinds of things, based on the description, should we store related to these entities?*

*Student: Student names, ID numbers... course numbers and course names.*

*AI: Anything else?*

*Student: Student email addresses and phone numbers.*

*AI: That takes care of all of the attributes in the scenario. Now, is there something we can use to uniquely identify a particular student and course?*

*Student: Student ID? Course Number?*

This excerpt illustrates procedural moves, where the student names elements directly from the scenario. When it is time to define the relationship, the student unfamiliar with the concept of a many-to-many relationship may suggest foreign key positioning that will lead to redundancy. At this point, the AI dialogue will shift from procedural to conceptual and guide the student through a line of questioning to illustrate the implications of incorrectly assigning foreign keys in many-to-many relationships and lead to their explanation. Such transitions are the analytic target of the transcript-coding scheme described in the Evaluation Plan.

## Design Principles

Five design principles are extracted from the artifact:

- Student as protagonist. Every move the student makes is their own. The AI surfaces, probes, and redirects, but it does not substitute its reasoning for theirs.

- Instructor-loaded context. The pedagogical design is the instructor's contribution, encoded in the system prompt and made explicit through the six-step algorithm. The AI is the delivery mechanism.
- Socratic constraint as structure. The AI is configured to respond with questions rather than answers. The constraint is implemented at the prompt level and reinforced by the pedagogical structure rather than enforced behaviorally.
- Transcript as artifact. The exchange is both the learning activity and the research instrument. The deliverable is also the data.
- Circumvention as signal. A student who attempts to extract the answer rather than engage with the questioning reveals their perception of the exercise. This is data and not failure.

## Evaluation Plan: The Exploratory Pilot

The pilot described here is framed as an exploratory, descriptive investigation. Given the modest sample size (approximately 30 graduate students at a regional university) and the developmental stage of the artifact, the pilot is not designed to support inferential statistical claims and no a priori power analysis is offered. The aim is to characterize what happens when the designed Socratic facilitation layer is deployed in a real graduate database course, to surface the patterns and behaviors that warrant deeper investigation, and to refine the artifact for a subsequent larger study under separate funding.

### Research Questions

The pilot is structured around three exploratory research questions:

- RQ1: What patterns of conceptual movement are observable in students' transcripts generated through the Socratic facilitation layer?
- RQ2: How do students who demonstrate conceptual understanding (as evidenced by transcript content and subsequent course performance) interact with the system differently from those who do not?
- RQ3: What circumvention behaviors emerge in the transcripts, and what do these behaviors reveal about students' perception of AI-facilitated learning?

### Participants and Setting

The pilot site is a graduate database course (approximately 30 students) at a regional university. The course is required for the program and is taught by the Principal Investigator. All students complete both the AI-facilitated Socratic exercise and the instructor's traditional video-based demonstration of the same ERD material during the two-week unit, in counterbalanced order, where one cohort half engages with the AI exercise in week one and the videos in week two, while the other half engages with the videos first and the AI exercise second. This design ensures equitable instructional access for all students regardless of research participation. The analytic focus of the pilot is on the transcripts generated by the AI exercise. The design is qualitative and exploratory rather than comparative.

### IRB Protocol

The pilot has been submitted for institutional review and is anticipated to qualify for exempt review under the regulatory category for educational research conducted in normal classroom contexts. Several features of the protocol mitigate concerns characteristic of instructor-led classroom research. Consent for research use of transcripts is administered by the Co-Principal Investigator, not by the course instructor, to avoid the appearance of grade-related coercion. Transcripts are not used for research purposes until course grades have been submitted. Students who decline research participation still complete the assignment and receive

normal credit, and their transcripts are excluded from the research corpus. Personally identifying information is replaced with study identifiers before analysis.

## Tool Description and Hosting

The instructional tool is a faculty-developed Socratic facilitation interface accessed via web browser. The tool calls the Anthropic Claude API on the student's behalf, with the instructor-designed system prompt loaded on every exchange. The Claude model is Sonnet 4.6, version-pinned via the API model string for the duration of data collection. The tool is hosted on infrastructure controlled by the Principal Investigator. The hosting arrangement does not introduce additional student data exposure beyond what is required to call the Claude API.

## Data Collection

The primary instrument is the full Claude chat transcript, submitted by the student as part of the assignment deliverable through the institution's learning management system. Each transcript captures the complete student-AI exchange from opening prompt to exercise completion. Course performance data (grades on subsequent ERD-related assessments) provides a secondary measure for characterizing the relationship between transcript-observable reasoning patterns and downstream performance.

## Analysis Approach

The unit of analysis is the student session—one complete transcript from opening prompt to exercise completion. Primary analysis is qualitative thematic coding of three move types: (1) procedural moves, in which the student names elements (entities, attributes, relationships) by direct extraction from the scenario or by following the algorithm's prescribed sequence; (2) conceptual moves, in which the student infers structural necessity that is not directly given by the scenario—recognizing, for instance, that an attribute cannot attach to a given entity, that a relationship requires entity status of its own, or that a cardinality constraint follows from a business rule rather than from notation; and (3) points of stall, at which the student pauses, expresses confusion, or attempts to extract an answer rather than reason through, with the AI's response and the student's subsequent move both coded. Coding is performed by both the Principal Investigator and the Co-Principal Investigator independently, with inter-rater reliability assessed and disagreements resolved through discussion. The use of the Co-PI who is not the course instructor as an independent second coder addresses the validity concern that the PI alone might code transcripts in ways shaped by hypothesis expectations or by knowledge of individual students. Secondary analysis treats the exchange as a behavioral sequence, though this parallel will not be over-claimed prior to empirical testing. Coding will identify the conversational moves that precede a student's shift from procedural to conceptual articulation. Circumvention analysis will identify and characterize attempted evasions of the Socratic structure, theorized as signals of students' perception of the exercise.

Several validity threats are acknowledged. The transcript is both the assignment deliverable and the research data, which introduces grading stakes into the data the student produces. This is a feature of the design but also a limitation of inference. Model behavior may vary across Claude versions. Version-pinning addresses this for the pilot window but does not eliminate cross-model generalization concerns. The pilot is scoped to a single institution and a single instructor and replication across institutions and instructors is reserved for the subsequent study.

## Discussion and Implications

The framing developed in this paper has implications beyond the specific artifact described. Generative AI as a facilitation layer is fundamentally different from generative AI as a content tool. The distinction matters for instructional design, academic integrity policy, and faculty professional development. A faculty member

who treats AI as a content tool will face an academic integrity crisis. A faculty member who treats AI as a facilitation layer will design assignments in which AI use is built into the structure of the learning activity itself, with the AI configured by design to support rather than substitute for student reasoning.

The instructor's value, in this framing, is in the design of the questioning sequence and not in the delivery of answers. This has institutional implications. Faculty expertise in instructional design becomes more visible, more transferable, and more valuable as AI capabilities expand. The faculty member who can design a Socratic prompt sequence for entity relationship modeling has produced an artifact that, with appropriate evaluation and refinement, can potentially scale across cohorts and courses in ways that traditional lecture content cannot. The empirical substantiation of this claim is the work of the pilot and the subsequent larger study.

Several limitations bound the present contribution. The artifact has been designed but not yet empirically evaluated and the exploratory pilot is the necessary next step. The pilot is scoped to a single instructor and a single institution. The system prompt remains the instructor's intellectual property and is not released in this paper, though the six-step pedagogical algorithm that structures the prompt is disclosed and a redacted structural skeleton of the prompt is provided in Table 1. Model behavior under prompt constraint is probabilistic rather than guaranteed, and version-pinning addresses this for the pilot but not for long-term deployment. When Sonnet 4.6 is eventually deprecated, the pedagogical layer survives the model's retirement. The six-step algorithm, the Socratic constraint, and the redirection rules are model-agnostic design specifications, and migration to a successor model is a re-validation task—confirming constraint adherence on the new model before redeployment—rather than a redesign task. The long-term deployment story therefore treats the model as a replaceable component beneath a stable instructional design.

### Conclusion and Future Directions

The designed artifact is ready for deployment. The exploratory pilot will characterize what student-AI Socratic exchanges look like in practice, generating both pedagogical observations and a research corpus of student-AI interaction transcripts. The design principles extracted from the artifact are intended to be transferable to other IS pedagogical contexts where the procedural-conceptual gap is consequential.

This study presents three future directions. First, the present Socratic discovery phase is one component of a broader pedagogical architecture currently under development. A peer knowledge-sharing platform in which students post and resolve course-related misconceptions in a moderated community-of-practice structure will provide the social and collaborative dimension that the present individual-cognition layer does not. A designed-error phase, in which students encounter and diagnose constructed ERD errors after the Socratic discovery exercise, will sequence the discovery and correction paradigms rather than treating them as alternatives. Second, behavioral analysis of the transcript corpus as a standalone research contribution will treat the dialogue as event-coded data with potential parallels to game-based behavioral research. Third, multi-site replication and extension to additional IS domains will test the transferability of the design principles.

The broader research program situates this work within an emerging architecture for theorizing the conditions under which Generative AI tools support rather than supplant the development of conceptual understanding. The artifact stands ready. What the transcripts will make visible about student reasoning, about the boundary between procedural and conceptual knowledge, and about the architecture of facilitated inquiry is the work that follows.

## Acknowledgements

Claude (Anthropic) assisted with drafting, structuring, and editing sections of this manuscript. The author reviewed, revised, and approved all final content and is fully accountable for the originality, validity, and integrity of all content in this submission.

## References

- Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. Longman.
- Barrows, H. S., & Tamblyn, R. M. (1980). Problem-based learning: An approach to medical education. Springer.
- Carl, K., Babb, J., Keith, M., & Spruill, A. (2021). Inside the Antichamber: How purposeful game design reveals and refines player archetypes. *Issues in Information Systems*, 22(2), 1–20. [https://doi.org/10.48009/2\\_iis\\_2021\\_1-20](https://doi.org/10.48009/2_iis_2021_1-20)
- Chen, P. P. (1976). The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36. <https://doi.org/10.1145/320434.320440>
- Chukhlomin, V. (2024). Socratic prompts: Engineered dialogue as a tool for AI-enhanced educational inquiry. *Latin American Business and Sustainability Review*, 1(1), 1–14. <https://doi.org/10.70469/labsreview.v1i1.10>
- Favero, L., Pérez-Ortiz, J. A., Käser, T., & Oliver, N. (2024). Enhancing critical thinking in education by means of a Socratic chatbot. *International Workshop on AI in Education and Educational Research (AIEER 2024)*, co-located with ECAI 2024, Santiago de Compostela, Spain. <https://doi.org/10.48550/arXiv.2409.05511>
- Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, 37(2), 337–355. <https://doi.org/10.25300/MISQ/2013/37.2.01>
- Gregorcic, B., Polverini, G., & Sarlah, A. (2024). ChatGPT as a tool for honing teachers' Socratic dialogue skills. *Physics Education*, 59(4), 045005. <https://doi.org/10.1088/1361-6552/ad3d21>
- Hashmi, S. F. A., & Rebello, N. S. (2025). Analyzing undergraduate problem-solving in physics through interaction with an AI chatbot. <https://doi.org/10.1119/perc.2025.pr.Hashmi>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. <https://doi.org/10.2307/25148625>

- Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? *Educational Psychology Review*, 16(3), 235–266. <https://doi.org/10.1023/B:EDPR.0000034022.16470.f3>
- Ma, T., Chen, R., Li, A. T., & Liu, H. (2024). Socratic ChatGPT: Theory, design, and empirical evaluations. *PACIS 2024 Proceedings*, 12. [https://aisel.aisnet.org/pacis2024/track14\\_educ/track14\\_educ/12](https://aisel.aisnet.org/pacis2024/track14_educ/track14_educ/12)
- Paul, R., & Elder, L. (2008). Critical thinking: The art of Socratic questioning. *Journal of Developmental Education*, 31(3), 34–35.
- Payne, K., Giboney, J., Schuetzler, R., & Keith, M. (2017). Examining the learning effects of live streaming video game instruction over Twitch. *Computers in Human Behavior*, 77(1), 95–109. <https://doi.org/10.1016/j.chb.2017.08.029>
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2008). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
- Suraweera, P., & Mitrovic, A. (2004). An intelligent tutoring system for entity relationship modelling. *International Journal of Artificial Intelligence in Education*, 14, 375–417. [https://doi.org/10.3233/IRG-2004-14\(3-4\)07](https://doi.org/10.3233/IRG-2004-14(3-4)07)
- Tufino, E., & Gregorcic, B. (2025). Creating a customisable Socratic AI physics tutor. *Physics Education*, 60(6), 065037. <https://doi.org/10.1088/1361-6552/ae0d23>
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press. <https://doi.org/10.2307/j.ctvjf9vz4>
- Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2), 89–100. <https://doi.org/10.1111/j.1469-7610.1976.tb00381.x>