Expanding Implementation of an AI Conversational Agent to Support Troubleshooting in an Engineering Lab Environment

Dr. D. Matthew Boyer, Clemson University

Dr. Boyer is a Research Associate Professor in the Department of Engineering and Science Education and an Educational Proposal Writer in the College of Engineering, Computing and Applied Sciences.

Mitra Varun Anand, Worcester Polytechnic Institute

Mitra Anand serves as the Associate Director of Makerspace, and Innovation and Entrepreneurship, in addition to being an Adjunct Faculty of Mechanical Engineering at Worcester Polytechnic Institute.

Anand's research interests lie in combining hands-on Maker skills with an entrepreneurial mindset and value creation, aiming to develop practical solutions for real-world problems. He is enthusiastic about innovation in engineering education, design thinking, prototyping, program development, crafting interactive curricula, and bringing ideas to fruition.

With over 8 years of experience in Mechanical and Mechatronics Engineering, Anand possesses a solid background in Innovation and Entrepreneurship education, Additive Manufacturing, and Digital Fabrication technologies. He has taught lectures and workshops on advanced subjects to more than 1000 undergraduate engineering students and 150 graduate students, while advising on over 500 student and faculty research projects.

His academic credentials include an M.S. in Mechatronics and Robotics Engineering from New York University and a B.E in Mechanical Engineering from Anna University.

Sahil Mirani, Worcester Polytechnic Institute
Mr. Tim Ransom, Clemson University

Computer Science Lecturer, PhD candidate with the Engineering and Science Education Department at Clemson University.

Prof. Ahmet Can Sabuncu, Worcester Polytechnic Institute

I hold a Ph.D. in Aerospace Engineering and am currently a faculty member in the Department of Mechanical Engineering at Worcester Polytechnic Institute. My research focuses on engineering education, particularly on problem solving in ill-structured domains, problem typologies, and the role of complexity in learning. I investigate how digital and AI-supported learning environments can be designed to foster adaptive expertise in engineering practice. My work also explores innovation in engineering product design and the integration of entrepreneurial mindset into engineering education. Through both research and teaching, I aim to bridge foundational engineering knowledge with real-world problem-solving practices.

Jonah Cohen, Worcester Polytechnic Institute

Jonah Cohen is a third-year engineering student at Worcester Polytechnic Institute pursuing a BS/MS in Mechanical Engineering.

WIP: Expanding Implementation of an AI Conversational Agent to Support Troubleshooting in an Engineering Lab Environment

Introduction

This Work-in-Progress paper reports on the expanded implementation of an AI conversational agent designed to provide just-in-time troubleshooting support in an undergraduate mechanical engineering experimentation course centered on electronics and instrumentation. Building on an initial classroom deployment reported previously, the current iteration emphasizes two design commitments. First, the system is retrieval-first: it prioritizes instructor-authored course materials and lab documentation, ensuring responses remain grounded in the specific equipment, procedures, and constraints of the course. Second, the agent is intentionally scaffolded to guide rather than tell, using structured questioning and diagnostic prompts intended to support students in generating and testing multiple solution pathways before converging on a fix.

Engineering labs centered on electronics and instrumentation create a persistent instructional tension. Students often need timely help to keep moving, yet the most educational support does not simply provide fixes, but also helps students learn how to diagnose problems. In practice, delivering this kind of guided interaction consistently is difficult when support is reactive, repeated across many students, and constrained by the limited time of instructors and TAs. High student-to-teaching assistant ratios create predictable bottlenecks when students encounter circuit errors, sensor malfunctions, or code-debugging hurdles during hands-on work with Arduino or Raspberry Pi systems.

This WIP submission focuses on design revisions and the implementation shift planned for Fall 2025, moving from broad tool access to case-specific use cases to keep the agent contextually aligned with the particular lab activity and equipment in play. Our goal is twofold: (1) to reduce bottlenecks in routine procedural and troubleshooting questions that burden instructional staff, and (2) to foster divergent thinking in troubleshooting, which is operationalized in this course context as exploring multiple plausible explanations, linking symptoms to potential root causes, and selecting next steps based on evidence rather than trial-and-error alone. We introduce a transferable design pattern and evaluation blueprint intended to support adoption by instructors who want to implement retrieval-augmented, scaffolded conversational support in lab settings without needing to be AI experts.

Pedagogical Framework and Learning Objectives

This project is designed around two distinct but complementary goals. The first goal is operational: to reduce bottlenecks in laboratory instruction by providing consistent, immediate troubleshooting support for common procedural and instrumentation issues that otherwise require repeated TA attention. The second goal is pedagogical: to strengthen students' troubleshooting practices by guiding them toward evidence-based reasoning rather than answer-seeking, with support that promotes independent problem-solving and deeper conceptual thinking during lab work.

Our pedagogical approach builds on established engineering troubleshooting techniques that can be implemented through structured dialogue, including Socratic questioning, first principles reasoning, and decision-tree logic. In our earlier implementation work, we treated troubleshooting as a learning mechanism that can be supported by a structured diagnostic, guiding students to their own answers rather than revealing solutions immediately. This design

stance is retained here, with an explicit commitment to "guide rather than tell" as the central interaction principle.

In practice, the pedagogical framework is enacted through scaffolded conversational routines that function as Socratic scaffolds. The agent prompts students to articulate what they observe, identify what has been tried, and link symptoms to plausible causes before proposing next steps. The intent is not to replace TA support, but to provide an always-available layer of guided inquiry that helps students remain productively engaged when human assistance is delayed.

Student Learning Objectives

Within this project, we define divergent thinking in troubleshooting as the student's ability to explore multiple plausible pathways before converging on a fix, with decisions anchored in evidence rather than random trial-and-error. This is operationalized in the lab context through the following learning objectives:

1. **Generate plausible hypotheses**: Students will propose more than one plausible cause for a malfunction by connecting observed system symptoms to potential error sources (root cause reasoning), rather than treating the problem as a single unknown with a single correct answer.
2. **Select next steps using evidence**: Students will justify diagnostic actions, e.g., tests, checks, parameter changes, based on what they already know about the system, the procedure, and the observed behavior, using structured inquiry routines aligned with first principles and decision logic.
3. **Reflect and iterate**: Students will iteratively refine their hypotheses and actions in response to outcomes, treating troubleshooting as a learning process that builds confidence and competence over time.

These objectives drive both the interaction model and the planned evaluation approach. In later sections, we describe how these learning objectives are used to structure the analysis of chat interactions and student feedback to assess whether the conversational scaffolds prompt deeper inquiry while also supporting the operational goal of reducing routine instructional load.

System Design and *The Blueprint*

We define a “Transferable Design Pattern and Evaluation Blueprint” to create a reproducible and scalable framework. The blueprint serves as a roadmap for instructors to reimplement a similar AI support without requiring deep expertise with the workings of Flowise or another UI-focused AI agent builder. The system is built upon a modular architecture that prioritizes stored course knowledge over the broad scope of the general-purpose large language models.

Architecture: Retrieval-Augmented Generation (RAG)

The technical foundation of the agent relies on a RAG structure, which ensures the agent’s context is specific to the engineering lab. As shown in the “Knowledge & Indexing” section of the system architecture blueprint (Figure 1), the process begins with a Document Store containing the instructor-authored case study and Socratic guides.

These documents are processed through an Embeddings engine and stored in an online Pinecone Vector Database to prioritize relevant course information. When students prompt the agent, the system does not only use the LLM’s training data. Instead, it uses a Retrieval Tool, seen in the “Retrieval Logic” section of the blueprint, to query the vector database for the relevant course information prompted by the user. Using the retriever tool over the LLM’s

training data prevents hallucinations commonly found in AI agents and forces the agent to reference specific case study information before providing the user a final response.

Interaction Model: Socratic Scaffolds

The agent's logic is modeled on Socratic dialog with students. Unlike standard chatbots that provide direct answers, our interaction model utilizes a Buffer Memory, as shown in the "AI Processing" section, to ensure the agent has access to the entire chat history, which is essential for a Socratic Scaffold. Doing so allows the agent to recognize where a student is in the troubleshooting process.

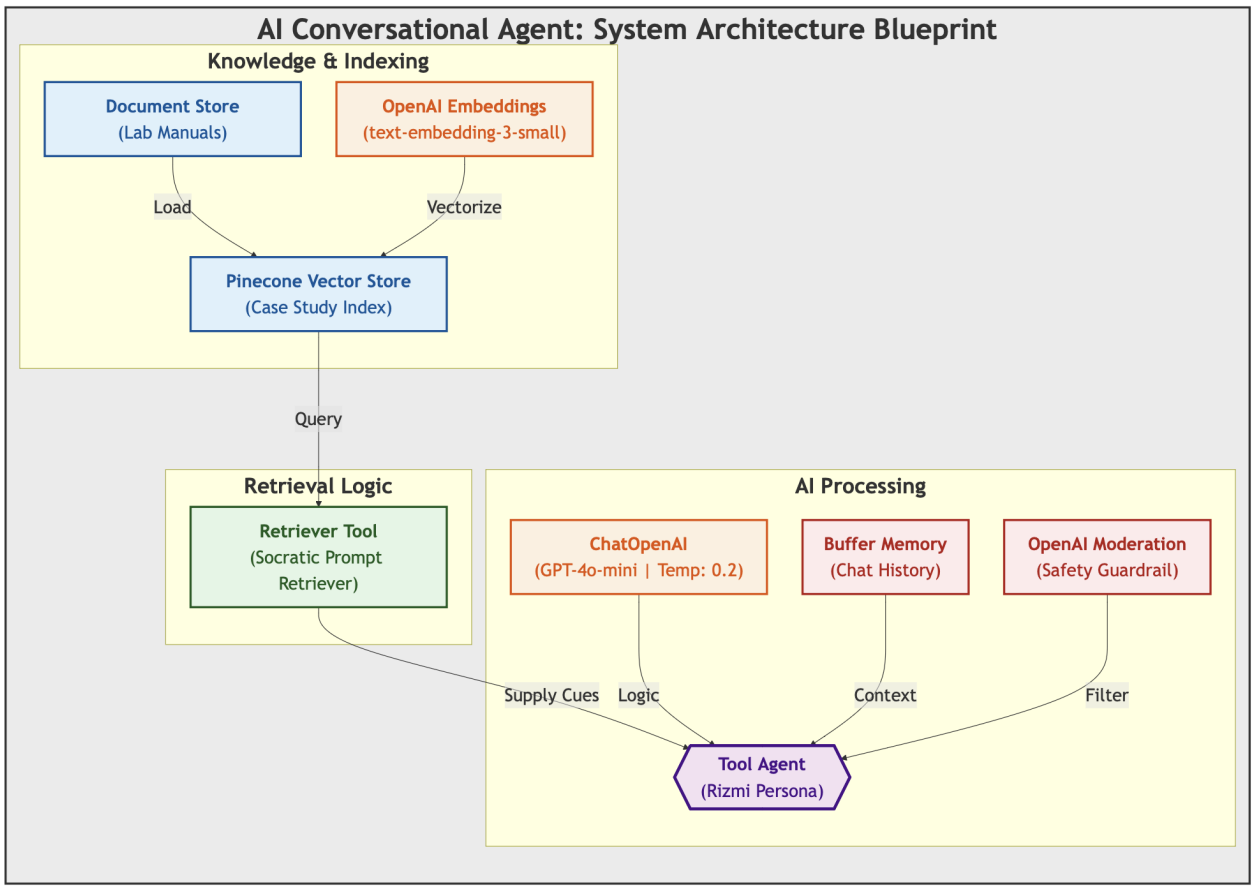

Figure 1: Modular AI Architecture for Socratic Scaffolding and Root-Cause Analysis

The agent is provided with detailed instructions designed to prevent the agent from giving direct answers. The agent is aware of the final answers for the case studies and is instructed to use Socratic dialogue to help students reach the final answer rather than providing the direct answer. For example, if a student requests a direct solution, the agent will trigger a Socratic redirection, asking the student to hypothesize why the current state differs from the expected output. Additionally, for the case study, the agent can also bring attention to other symptoms rather than honing in on incorrect symptoms.

The agent uses the Socratic Scaffolding to guide the students towards the final answers. For the experiment, the final answers were focused on different areas of the experiment. As a result, the agent keeps a track of what solutions the student has found, and provides a "win-condition" or final prompt to let the student know if the case study has been solved.

Implementation Guide: Guardrails and Logic Nodes

The final component of the blueprint is a set of implementation guardrails designed for safe deployment. These guardrails are located within the "AI Processing" and act as a buffer between the student and the AI agent. These heuristics include:

- **Precision Calibration:** The ChatOpenAI model is set to a low temperature (0.2) to ensure factual accuracy and consistency over creative solutions. Ensuring the agent prioritizes guiding students through the local coursework over the LLM's training data.
- **Automated Moderation:** An OpenAI Moderation node sits between the agent and the student to ensure no inputs violate safety protocols per OpenAI's moderation policies.
- **Contextual Routing:** The blueprint allows for "case-specific" retrieval, where the agent can focus on a specific case study. The current deployment is focused on one case study, but the contextual routing can be used if the agent is to be trained on multiple case studies.

Fall 2025 Revisions: Move to Case-Specific Use Cases

Building on findings from the initial deployment in Fall 2024, the Fall 2025 implementation shifts from a "broad access" to a "case-specific use case." In the earlier versions, the agent had access to the entire repository of course documents simultaneously, which resulted in more hallucinations and made it challenging to enforce detailed instructions in the system message. Furthermore, the broad access deployment had an inconsistent Socratic framework; the large knowledge base overwhelmed the Buffer Memory layer, causing the agent to default to repetitive prompting over facilitating productive convergent/divergent thinking [1].

The revised implementation significantly reduces the scope of the provided knowledge documents. The agent has access to one specific case study with details of the full experiment. Following this, students can interact with the chatbot to learn about the case study, with the agent providing Socratic prompts to help the users diagnose the problems with the experiment. This refinement ensures that the "just-in-time" support is not only conceptually sound but contextually accurate to the student's immediate concerns when discussing the case study.

Methodology and Data Collection

The methodology for this work-in-progress study is designed to evaluate the responses from the AI agent and the impact on student pedagogical growth. The Fall 2025 implementation focuses on refining data collection streams to isolate the development of divergent thinking in engineering students while using the chatbot.

Participants and Context

The study involves undergraduate engineering students enrolled in the ME3902: Project-Based Engineering Experimentation, a junior-level course focused on developing high-level skills in measurement methods and computer-based data acquisition. The course utilizes platforms like Raspberry Pi and Arduino, which allow students to engage in laboratory modules that span mechanical, energy, and material measurement, culminating in an open-ended project where students must synthesize these skills to control a complex physical system of their choosing. Additionally, this course serves as a foundation for mechanical engineering students before their Major Qualifying Project (MQP), making these students the ideal demographic for testing the efficacy of "just-in-time" scaffolding.

Data Sources

To gain a comprehensive view of the agent's impact, data is focused on two primary sources:

- **Chat Logs and Interaction Analytics:** All chatlogs are stored using Langfuse, an open-source LLM engineering platform allowing users to analyze and debug AI applications. These logs were used to code for performance indicators that measure interaction depth, the number of turns taken to reach a solution, the frequency of "Socratic" versus "procedural" queries, and the pedagogical growth of the student through conversations with the agent.
- **Student Feedback Surveys:** Post-lab surveys were conducted to prompt the user to explain their thought process while walking through the troubleshooting case study and to describe the agent's helpfulness in diagnosing the problems with the experiment.

Ethical Considerations and Anonymity

The team put significant emphasis on the ethical considerations of the agent's deployment to ensure students would feel comfortable choosing to work with the agent or opting out. One risk the team foresaw was that students may feel "judged" by the system, potentially preventing them from asking questions they feel are dumb or revealing their true confusion. To foster a sense of trust and demonstrate the team's stance, a rigorous privacy protocol was established.

- **Protocol for Anonymity:** The team utilized Langfuse to perform qualitative research analysis on the chatlogs in fully anonymized datasets. Additionally, the students were informed to keep out all personal information when communicating with the agent. Lastly, if any personal identifiable information (PII) was found, the team removed it before performing any coding. The member responsible for removing the PII did not perform the coding to prevent any biases.
- **Establishing Trust:** To establish trust, the team explicitly informed students that their queries and errors had no impact on their final grade. Additionally, the chatlogs were not reviewed by TAs if it was a concern for any student. By positioning the agent as a "non-judgemental" thought partner rather than a monitoring tool, the goal was to preserve the integrity of the Socratic dialogue and encourage students to use the agent for experimentation and intellectual risk-taking.

Evaluation Plan

As the AI conversational agent is built to support students' divergent thinking, analysis must be done to understand if the agent successfully promotes this thinking in students. This evaluation is conducted by analysing all of the conversation logs to find how the agent performs and how the student responds. While evaluating both the agent and the student, definitions of types of thinking must be defined. Colin G. DeYoung, Joseph L. Flanders, and Jordan B. Peterson in their report, "Cognitive Abilities Involved in Insight Problem Solving: An Individual Differences Model," describe three types of thinking when solving a problem:

1. **Convergent thinking:** Skills of verbal intelligence and analytical problem solving. This type of thinking is used when students have a starting point to solve a problem and think linearly to find the solution.
2. **Divergent Thinking:** Skills of creative thinking. Students use this type of thinking to find a potential starting point of a problem.
3. **Breaking the Frame:** The ability to transition from convergent thinking to divergent thinking. Students break the frame when they can no longer convergently think and they have to find a new starting point to solve the problem.

Evaluating AI conversational agent usability

The purpose of the AI conversational agent is to provide educational support for students as they complete their lab work. To this end, a qualitative evaluation criteria is our current evaluation strategy. Alongside teaching assistant reflections on the student usage that they observe, the students' self-reflections are a utilized evaluation tool. TAs have observed the tool being well-received during the labs and have noted that students are not demonstrating obvious difficulties with the tool usage. Likewise, student reflections on the tool have thus far been positive.

Preliminary Results

The chatlogs showed interesting results when looking at how the agent engaged in Socratic dialogue and how it promoted divergent or convergent thinking. The chatbot often began each conversation by describing the experiment and providing a symptom of the failed experiment. After engaging with the cue, the agent would offer Socratic dialogue, such as “What do you think?” or “Any ideas on what might be going wrong?” The initial dialogue would help the agent understand the student’s level of understanding. Following this, the agent would promote convergent thinking as students would focus on the symptoms they wanted the agent to help analyze. For example, one student described their problem-solving process in a reflection:

> The first cues I gathered were about the experimental setup. I [asked] "is there any gap between the fins and the can" and "what is wrong with the electrical setup" to start. It indicated that there was a gap between the fins and the can and that the voltage measured across the reference resistor seemed to be too high. This indicates that the problem was in the physical setup most likely and not the code. [...]
>
> The cue about the fins prompted me to ask about the fins, and [the chatbot] said they were cool to the touch, further indicating poor contact. Based on the two cues that I collected from that it was [sufficient] to say that the reasoning for the poor and unexpected results was for the wrong resistor and the poor contact between the fins. [...] The biggest insight was that he measured the wrong voltage over the reference resistor.

The chatbot was demonstrated to somewhat guide students through their problem solving process using divergent thinking as well. The logs indicate an ability to redirect student attention away from their current point of focus and onto a more direct path to a solution. The overall problem solving pathway was determined by the student, but was guided by the chatbot towards a solution by interleaving cues known from the dataset into responses given to the student. Preliminary coding phases indicate that this behavior is supporting student learning while enabling students to approach the problems in their own ways.

> Student: You should look into the thermistor response time and see if adjusting the data logging intervals and sample durations changes anything.
>
> Chatbot: That's a solid suggestion! I hadn't really considered adjusting the data logging intervals based on the thermistor response time. It could definitely help clarify if the readings are lagging behind the actual temperature changes. I’ll have to dig into the specifications of the thermistors to see what their response time is. **But, you know, even**

> **with that in mind, I'm still a bit confused about why the finned can is cooling slower than the control can.** Do you think there might be something else affecting the cooling performance? I can pull up some cues related to the fins or the setup if you want!

This chat log excerpt shows the AI helping a student acknowledge their observation in the overall system, but still nudging them towards the solution - in this case, the symptom most related to the root cause of the issue.

Conversation Agent Results

Evaluating the AI conversational agent showed multiple areas of improvement. The area where the agent struggled the most was not promoting convergent thinking. There were 78 responses in 29 chat logs that gave the student a starting point for solving the problem. This action counters the goal of this research by preventing the student from breaking the frame and divergently thinking for a different starting point. Additionally, the agent showed 22 instances where it did not tell the student that they solved the problem and 18 instances where it gave away the solution. This AI conversational agent needs significant work to achieve the goal of this project.

Discussion and Future Work

This work is currently in an iterative cycle, developing the interactions with the students based on educational scaffolding. The efficacy of the Socratic dialog method is well known to apply in traditional educational delivery methods, but has yet to reach full verification using generative artificial intelligence tools. Further work to validate the educational efficacy of the tool is required before wider adoption can be recommended. Focus groups with the students, as well as structured reflections from the teaching assistants, will be used for this validation.

Additionally, the architecture of the artificial agents is still in active development. The current RAG agent architecture has been applied to this particular coursework, but it is unknown what course characteristics are required for successful transfer into other courses and domains.

References

[1] M. Ismail, D. Kachadoorian, S. Mirani, D. Boyer, T. Ransom, and A. Sabuncu, “BOARD # 63: AI Chatbot for Enhancing Troubleshooting in Engineering Labs,” *2025 ASEE Annual Conference & Exposition Proceedings*, 2025, doi: https://doi.org/10.18260/1-2--55878.

[2] C. G. DeYoung, J. L. Flanders, and J. B. Peterson, “Cognitive Abilities Involved in Insight Problem Solving: An Individual Differences Model,” *Creativity Research Journal*, vol. 20, no. 3, pp. 278–290, Aug. 2008, doi: https://doi.org/10.1080/10400410802278719.