Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 7, No. 1, 2026

A Multimodal Conversational Agent for Co-Creating Teaching Slides and Lesson Plans

Huaxin Zheng *

Wenzhou Polytechnic, Wenzhou, China

* **Corresponding author**: (Email: huaxin179@163.com)

Abstract: This paper presents a multimodal conversational teaching agent that co-creates classroom slide decks and lesson plans with teachers through a staged review process. The prototype was motivated by a practical teaching-design scenario in which courseware preparation is fragmented across several tools and teacher intent is often captured only superficially. The proposed system combines three capabilities: multi-turn intent clarification, retrieval-augmented generation over a locally curated knowledge base, and optional reference-file grounding from uploaded PDF, DOCX, PPT, image, or video materials. Instead of exporting final artifacts at the first interaction, the agent follows a text-first pipeline. In Round 1 it produces a slide draft and a lesson-plan draft for review. In Round 2 it converts the approved slide draft into a structured slide object and then into PPTX. In Round 3 it converts the approved lesson-plan draft into Markdown and then into DOCX. This design keeps the teacher in control of pedagogical structure, terminology, and difficulty level while preserving the efficiency gains of large language models. The implementation separates chat-oriented intent handling from deterministic formatting and export steps, and it uses explicit state variables to support revision, regeneration, and artifact continuity across rounds. A case-based verification on networking topics such as VLAN, WLAN, and STP shows that the system can align courseware drafts, exported slides, and exported lesson plans under a single interaction loop. The paper contributes a reproducible low-code architecture, a multimodal fusion strategy that does not require writing uploaded references into the knowledge base, and a teacher-centered workflow for controllable educational content generation.

Keywords: Artificial intelligence in education, human-in-the-loop generation, lesson plan generation, multimodal references, retrieval-augmented generation.

1. Introduction

Recent progress in large language models and multimodal assistants has created new opportunities for educational content generation, especially in scenarios that require intent understanding, iterative drafting, and domain grounding [1-5]. Yet many teaching-support tools remain fragmented: one tool is used for slide drafting, another for lesson plans, and still another for reference extraction. In classroom preparation, this fragmentation increases operational burden and makes it difficult for teachers to preserve a consistent pedagogical plan across different artifacts.

The prototype described in this paper targets a concrete instructional design problem: how to help teachers co-create slide decks and lesson plans for vocational networking courses while preserving teacher control over objectives, difficulty, examples, and practical tasks. The design goal is not merely to automate file export, but to support a teacher-centered co-creation loop in which the system can clarify requirements, use a local knowledge base, interpret uploaded references, generate draft artifacts, and revise them before export. This requirement aligns with the project specification that emphasizes multimodal references, local knowledge retrieval, two export targets (PPT and DOCX), and iterative revision.

To address this need, we implemented a multimodal conversational teaching agent with three design commitments. First, the system is text-first: draft content is reviewed before file export. Second, the system is grounded: generation can use both retrieval results from a locally curated knowledge base and cleaned summaries of uploaded references. Third, the system is stateful: draft artifacts are preserved across rounds so that modifications can target the latest approved version rather than restart generation from scratch. Figure 1 summarizes the architecture.

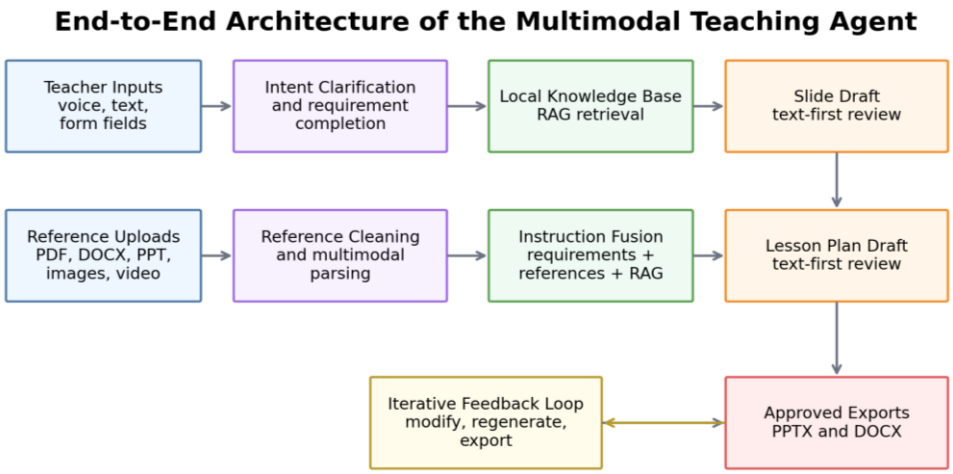

Figure 1. End-to-end architecture of the agent

Figure 2 shows the three-round interaction loop.

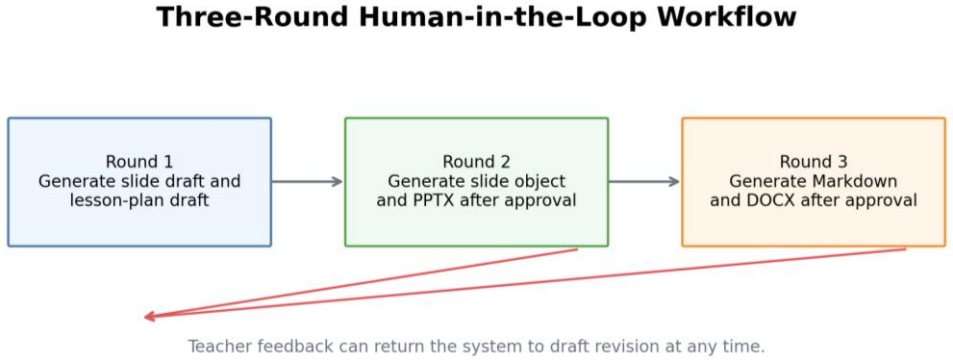

Figure 2. Three-round human-in-the-loop workflow

2. System Requirements and Design Principles

The system was specified around five functional requirements. The first requirement is multimodal input. Teachers should be able to provide course fields through structured form controls, natural-language instructions through a chat interface, and optional references through uploaded files. The second requirement is requirement clarification. Because early teacher inputs are often incomplete, the agent must ask follow-up questions until the instructional task is sufficiently specified.

The third requirement is grounded generation. Locally curated networking materials are stored in a vector knowledge base and retrieved through a retrieval-augmented generation pipeline [6, 7]. Uploaded reference files are not written into the knowledge base; instead, they are parsed on demand, cleaned, and selectively fused into the generation brief. The fourth requirement is dual-artifact generation. The agent must produce both slide content and a lesson plan while maintaining alignment between them. The fifth requirement is iterative control. Teachers should be able to revise draft content before exporting PPTX or DOCX [8].

These requirements led to a low-code architecture that separates high-variance conversational reasoning from deterministic formatting and export. Large language models are used for intent classification, clarification, fusion, draft generation, and revision [9]. Deterministic code nodes are used to normalize structured objects, split learning objectives into stable subtypes, clean whitespace and fields, and prepare export-safe representations [10]. This separation improves controllability and helps prevent formatting drift between review rounds.

Table 1. Functional modules in the proposed system.

Module	Primary Input	Core Function	Primary Output
Input and intent layer	Course fields, teacher prompts, optional files	Collects structured and free-form inputs; identifies generation, revision, slide export, and lesson-plan export intents	Structured requirement object
Clarification layer	Initial requirement object	Asks follow-up questions until missing objectives, constraints, and export targets are resolved	Completed requirement state
Grounding layer	Completed requirements, local corpus, uploaded references	Retrieves from the local knowledge base and parses optional references without persisting them into the knowledge base	Retrieval results and cleaned reference summary
Fusion layer	Requirements, retrieval results, cleaned references	Prioritizes teacher intent, then useful uploaded references, then knowledge-base supplements	Generation brief
Draft generation layer	Generation brief	Creates reviewable slide drafts and lesson-plan drafts before any file export	Draft artifacts
Normalization and export layer	Approved drafts	Transforms drafts into PPT objects and lesson-plan Markdown, then calls export plugins	PPTX and DOCX

3. Architecture and Workflow

As shown in Figure 1, the architecture contains six layers. The first layer is the teacher input layer, which collects course metadata, free-form requirements, and optional reference files. The second layer is the understanding layer, where an intent classifier distinguishes first-time generation, draft revision, slide export, and lesson-plan export. A clarification subroutine expands missing instructional information through follow-up dialogue. The third layer is the grounding layer, where a local vector knowledge base is queried and uploaded references are parsed and summarized.

The fourth layer is the instruction-fusion layer. It merges structured requirements, knowledge-base retrieval results, and cleaned reference summaries into a single generation brief. The fusion policy assigns the highest priority to explicit teacher requirements, the second priority to useful uploaded references, and the third priority to background knowledge retrieved from the local corpus. This ordering prevents retrieved content from overriding teacher intent.

The fifth layer is the artifact-generation layer. It first creates a slide draft for review, then a lesson-plan draft, and only after confirmation does it proceed to export-oriented formatting. The final layer is the feedback and export layer. Here the user can modify drafts, request PPT generation, or request DOCX generation. State variables preserve the latest slide draft, lesson-plan draft, review stage, and artifact history so that each round can continue from the latest approved text.

4. Multimodal Reference and Knowledge Fusion

Reference grounding is handled in two parallel branches. The knowledge-base branch retrieves networking concepts, standard terminology, troubleshooting points, and reusable practical hints from a local corpus curated for the target discipline. The uploaded-reference branch reads optional teacher-supplied files such as PDF notes, Word documents, slide decks, or other media. Because raw parser output is often noisy, the system applies a dedicated cleaning-and-distillation step to remove headers, repeated directory pages, image placeholders, irrelevant fragments, and off-topic material.

The cleaned reference summary is then inspected for usefulness with respect to the current course topic. If it is useful, the system extracts candidate knowledge points, teaching cases, terminology, and practice hints. If it is not useful, the branch is safely ignored. This strategy is important for two reasons. First, it prevents poor-quality uploads from contaminating generation. Second, it avoids the engineering overhead of inserting every uploaded reference into the local knowledge base. Instead, uploaded files remain ephemeral contextual evidence, while the knowledge base remains a stable domain memory.

This distinction between persistent domain grounding and transient reference grounding is central to the design. Persistent grounding supports reproducibility and terminology consistency, whereas transient grounding supports teacher customization. The resulting generation brief therefore contains both stable pedagogical structure and task-specific adaptation.

5. Artifact Generation and Deterministic Normalization

The system uses a text-first workflow for both export targets. For slides, the first step is not a PPT object but a reviewable slide draft. Each page contains a page title, page type, teaching purpose, key message, content blocks, and a visual hint. This representation is rich enough for teacher review and simple enough for targeted modification. After the teacher confirms the draft, a later node converts it into a normalized slide object with title, subtitle, sections, contents, and item lists, which can then be passed to a PPT export plugin.

The lesson-plan branch follows the same philosophy. Instead of generating a DOCX file immediately, the system first produces a reviewable lesson-plan draft aligned with the approved slide draft. The draft contains teaching objectives, focus points, difficulties, methods, preparation, process stages, assessment methods, and homework. After confirmation, a Markdown conversion node transforms the approved lesson-plan draft into export-ready Markdown, which is then sent to a DOCX plugin. This parallel design reduces the risk of premature file generation and gives teachers a consistent review experience across both artifacts.

A notable implementation detail is the use of deterministic patching before final export. For example, objective blocks can be normalized into knowledge, skills, and literacy targets, and homework blocks can be normalized into consolidation, practice extension, and reflective extension. These small deterministic steps help keep export objects structurally valid while leaving pedagogical phrasing under teacher control.

6. Case-Based Verification

To verify the prototype, we exercised it on representative networking topics such as VLAN, WLAN, and STP. These topics were chosen because they require a mix of conceptual explanation, troubleshooting logic, procedural configuration, and practical verification. They also expose common differences between purely theoretical slides and vocationally oriented training materials.

The tests showed that the system can keep slide drafts and lesson-plan drafts aligned when both are generated from the same approved brief. When a teacher uploaded a topic-specific reference deck, the cleaned reference branch could add terminology, examples, or conceptual framing without forcing those uploads into the persistent knowledge base. When no reference was uploaded, the system fell back to the local knowledge base and explicit teacher requirements. The workflow therefore remained stable in both reference-rich and reference-free conditions.

The most visible practical gain was controllability. Teachers could revise the slide draft first, approve slide export later, and independently approve lesson-plan export in a separate round. This staged workflow prevented irreversible formatting work from happening too early. Table 2 summarizes the role of each round and the expected user action.

Table 2. Output artifacts and approval triggers across the three interaction rounds.

Round	Teacher Action	System Output	Next Decision
Round 1	Generate slide draft and lesson-plan draft	Textual slide draft and textual lesson-plan draft	Revise or approve drafts
Round 2	Generate slides after draft approval	Normalized slide object and PPTX export	Revise slide draft or accept PPTX
Round 3	Generate lesson plan after draft approval	Lesson-plan Markdown and DOCX export	Revise lesson-plan draft or accept DOCX

7. Discussion

The prototype suggests three implications for AI-supported instructional design. First, a conversational generator should not export final files before the teacher has seen editable draft text. Second, locally curated domain memory and teacher-uploaded references should be handled differently rather than merged blindly into one storage layer. Third, low-code orchestration becomes substantially more reliable when content generation, object normalization, and export are separated into distinct nodes.

The current study also has limitations. It reports a design-and-prototype validation rather than a controlled classroom experiment. Therefore, it does not claim measured learning gains, time savings, or teacher satisfaction statistics. Future work should add user studies, compare draft quality across multiple model settings, and test stronger visual reference use for layout-aware slide generation. Video parsing and richer multimodal references also remain open engineering directions.

Even with these limitations, the design provides a practical pattern for educational artifact co-creation: conversational clarification, grounded fusion, text-first review, deterministic normalization, and stage-specific export.

8. Conclusion

This paper presented a multimodal conversational teaching agent for co-creating slide decks and lesson plans. The system combines multi-turn requirement clarification, local knowledge-base retrieval, optional uploaded-reference grounding, and a three-round human-in-the-loop workflow. Its key contribution is a text-first architecture that preserves teacher control before PPTX and DOCX export, while still benefiting from retrieval-augmented and multimodal generation. The prototype demonstrates that draft alignment, artifact continuity, and staged approval can be achieved in a low-code orchestration environment without writing uploaded references into the persistent knowledge base. This makes the design suitable for teacher-centered courseware co-creation in vocational networking education and similar instructional settings.

Acknowledgment

This manuscript is an English draft prepared from an internal project description and workflow prototype. No external funding was declared for this draft.

References

[1] O. Zawacki-Richter, V. I. Marin, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education - where are the educators?" Int. J. Educ. Technol. High. Educ., vol. 16, art. 39, 2019.

[2] W. Holmes, M. Bialik, and C. Fadel, Artificial Intelligence in Education: Promises and Implications for Teaching and Learning. Boston, MA: Center for Curriculum Redesign, 2019.

[3] UNESCO, Artificial Intelligence in Education: Challenges and Opportunities for Sustainable Development. Paris: UNESCO, 2019.

[4] L. Ouyang et al., "Training language models to follow instructions with human feedback," arXiv preprint arXiv:2203.02155, 2022.

[5] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," arXiv preprint arXiv:2201.11903, 2022.

[6] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," arXiv preprint arXiv:2005.11401, 2020.

[7] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," arXiv preprint arXiv:2004.04906, 2020.

[8] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," arXiv preprint arXiv:2304.08485, 2023.

[9] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "MiniGPT-4: Enhancing vision-language understanding with advanced large language models," arXiv preprint arXiv:2304.10592, 2023.

[10] E. R. Mollick and L. Mollick, "Assigning AI: Seven approaches for students, with prompts," SSRN Electron. J., 2023, doi: 10.2139/ssrn.4475995.