WEAVEMUSE: AN OPEN AGENTIC SYSTEM FOR MULTIMODAL MUSIC UNDERSTANDING AND GENERATION

Emmanouil Karystinaios
Institute of Computational Perception, Johannes Kepler University Linz, Austria
firstname.lastname@jku.at

ABSTRACT

Agentic AI has been standardized in industry as a practical paradigm for coordinating specialized models and tools to solve complex multimodal tasks. In this work, we present WeaveMuse, a multi-agent system for music understanding, symbolic composition, and audio synthesis. Each specialist agent interprets user requests, derives machine-actionable requirements (modalities, formats, constraints), and validates its own outputs, while a manager agent selects and sequences tools, mediates user interaction, and maintains state across turns. The system is extendable and deployable either locally, using quantization and inference strategies to fit diverse hardware budgets, or via the HFApi to preserve free community access to open models. Beyond out-of-the-box use, the system emphasizes controllability and adaptation through constraint schemas, structured decoding, policy-based inference, and parameter-efficient adapters or distilled variants that tailor models to MIR tasks. A central design goal is to facilitate intermodal interaction across text, symbolic notation and visualization, and audio, enabling analysis-synthesis-render loops and addressing cross-format constraints. The framework aims to democratize, implement, and make accessible MIR tools by supporting interchangeable open-source models of various sizes, flexible memory management, and reproducible deployment paths.

1. INTRODUCTION

Large language models increasingly act as planners that coordinate specialized tools for music tasks spanning text, symbolic notation, and audio. At the same time, practical use is often limited by the cost of inference, the difficulty of deploying heterogeneous models as tools, and the lack of mechanisms for cross-modal control.

In this paper, we introduce WeaveMuse, an open, agentic system that orchestrates music understanding, generation, and analysis across modalities. A manager agent interprets user goals, maintains dialogue and task state, and composes pipelines by selecting among specialist agents and tools. Specialist agents translate requests into machine-actionable specifications (modalities, formats, constraints), execute analysis or generation, and perform basic self-checks against musical requirements. The system is designed to be extendable and modest in its assumptions: it can run locally by using quantization and other efficiency strategies to fit diverse hardware budgets, or be accessed via HFApi, so that open models remain freely accessible to the community.

We develop the WeaveMuse as a reference system and toolkit to lower the barrier to cross-modal interaction between text, score, and audio. Compared to other music agentic frameworks such as [1–3], WeaveMuse is positioned as an open, multi-agent, modular framework that focuses equally on both symbolic and audio representations. The system serves as a demonstration and adaptation of, for the moment a few, useful tools that the MIR community develops. The paper focuses on deployment and usability considerations, reports initial behaviors under constrained settings, and outlines limitations and directions for integrating, distilling, and extending models for finer control and more interactive music creation.

2. SYSTEM OVERVIEW

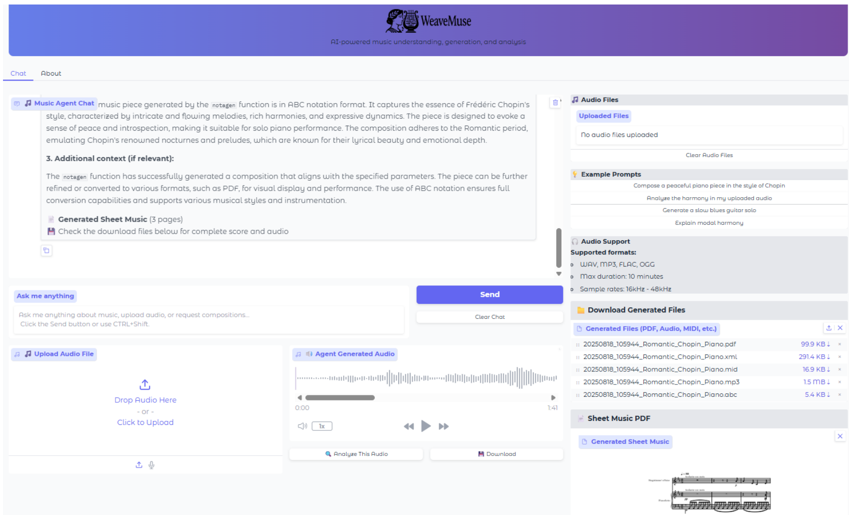
2.1 Core agent

The WeaveMuse core agent maintains task context and a tool/specialized agent router that selects and sequences tools while considering resource hints. It is also responsible for applying reasoning or verification strategies. All agents are built with the smolagents library [4].

2.2 Integrated tools

Adapters include: ChatMusician [5] for music-theory reasoning and text/symbolic grounding; NotaGen [6] for ABC notation generation with pdf and audio compilation; Audio Analysis [7, 8] models for segmentation and understanding; Stable Audio Open [9] for 44.1 kHz stereo synthesis; and Score Visualization for sheet rendering with MuseScore and score engraving correction using [10]. Alternative open models/agents of varying sizes can be swapped per task and budget, supporting progressive enhancement (small/large when resources permit). Some adapters include their own agent, which is responsible for formatting arguments and prompts as well as adapting user queries.

Attribution: E. Karystinaios, “WeaveMuse: An Open Agentic System for Multimodal Music Understanding and Generation”, in *Proc. of the 26th Int. Society for Music Information Retrieval Conf.,* Daejeon, South Korea, 2025.



music piece generated by the `notagen` function is in ABC notation format. It captures the essence of Frédéric Chopin's style, characterized by intricate and flowing melodies, rich harmonies, and expressive dynamics. The piece is designed to evoke a sense of peace and introspection, making it suitable for solo piano performance. The composition adheres to the Romantic period, emulating Chopin's renowned nocturnes and preludes, which are known for their lyrical beauty and emotional depth.

3. Additional context (if relevant):

The `notagen` function has successfully generated a composition that aligns with the specified parameters. The piece can be further refined or converted to various formats, such as PDF, for visual display and performance. The use of ABC notation ensures full conversion capabilities and supports various musical styles and instrumentation.

Generated Sheet Music (3 pages)
💾 Check the download files below for complete score and audio

Ask me anything about music, upload audio, or request compositions...
Click the Send button or use CTRL+Shift.

No audio files uploaded

Compose a peaceful piano piece in the style of Chopin
Analyze the harmony in my uploaded audio
Generate a slow blues guitar solo
Explain modal harmony

Supported formats:

- WAV, MP3, FLAC, OGG
- Max duration: 10 minutes
- Sample rates: 16kHz - 48kHz

20250818_105944_Romantic_Chopin_Piano.pdf	99.9 KB
20250818_105944_Romantic_Chopin_Piano.xml	291.4 KB
20250818_105944_Romantic_Chopin_Piano.mid	16.9 KB
20250818_105944_Romantic_Chopin_Piano.mp3	1.5 MB
20250818_105944_Romantic_Chopin_Piano.abc	5.4 KB

Figure 1. The WeaveMuse interface.

2.3 Interfaces and Deployment Modes

Both local and hosted interfaces are the identical. An example of the WeaveMuse interface is displayed in figure 1. Interface is based on Gradio [11] with custom bindings and tools to integrade different modalities and their interactions. It offers to the user a straghtforward and modular GUI. Local interface implements models locally while the hosted inteface leverages HuggingFace Spaces and API for model calls.

3. AUTONOMOUS MUSIC AGENTS

The agents in WeaveMuse follows a perceive, plan, act loop with verification and repair when possible:

1. **Perceive:** Ingest text, audio, or symbolic inputs.
2. **Plan:** Compose a tool graph (e.g. enrich query, compose, engrave, synthesize), subject to explicit constraints and resource hints.
3. **Act:** Invoke tools, use cache intermediates (ABC, MIDI, stems, analysis reports).
4. **Critique & repair:** When execution fails, or after a tool action is operated, analysis could be run to verify if the result matches the user's query.

3.1 Quantization & Efficiency

To operate under limited memory and compute while preserving musical quality, we employ quantization techniques and memory offloading. Furthermore, we support (i) dynamic precision switching per tool; (ii) CPU/GPU device placement and paging; (iii) lazy loading and on-disk caching for models and embeddings; and (iv) memory-aware batching for offline analysis jobs.

3.2 Deployment Considerations

Local: resource tiers (low/medium/high VRAM) map to default policies (INT4/INT8, cache offloading, attention kernels), ensuring functionality across different resource capacities. For reference, our high-resource tier capacity using Qwen3-Coder-30B [12] as the agent and reasoning backbone is achieved on a single NVidia A40. **Hosted**: HFApi access to open models provides immediate availability and community sharing. By using dynamic GPU allocation for the hosted application, we enable usability by everyone without overhead costs to the user or the host. The same planner and prompts run in both modes, aiding reproducibility.

3.3 Limitations

Agent-based systems are usually as efficient and effective as the underlying LLM model is potent. Computation or budget limitations can effectively downgrade the performance of the entire system. Furthermore, as this is a work in progress, tool orchestration and agentic prompting might not always work as expected. Smaller models (i.e. less than 3B parameters) do not always use the correct tools when interacting with user queries.

3.4 Availability & Reproducibility

The framework is open-source with a public repository. [1] Local and hosted configurations share identical planner and prompt templates. HFApi access to open models lowers the barrier for community evaluation.

4. CONCLUSION & FUTURE WORK

The framework demonstrates that an efficiency-first, agentic stack can deliver controllable end-to-end pipelines under tight resource budgets. Future work will focus on the integration and distillation of additional open models (symbolic and audio), on the addition of more structured control over form/mixing, and on improving cross-format interaction (notation/text/audio) via alignment signals. We aim to release distilled checkpoints and adapter suites, plus automatic policy selection from hardware probes for seamless local or hosted deployment.

[1] Code: `github.com/manoskary/weavemuse`

5. ACKNOWLEDGMENTS

This work was supported by the European Research Council (ERC) under Horizon 2020 grant #101019375 “Whither Music?”.

6. REFERENCES

[1] D. Yu, K. Song, P. Lu, T. He, X. Tan, W. Ye, S. Zhang, and J. Bian, “Musicagent: An ai agent for music understanding and generation with large language models,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2023.

[2] Q. Deng, Q. Yang, R. Yuan, Y. Huang, Y. Wang, X. Liu, Z. Tian, J. Pan, G. Zhang, H. Lin *et al.*, “Composerx: Multi-agent symbolic music composition with llms,” *arXiv preprint arXiv:2404.18081*, 2024.

[3] S. Liu, A. S. Hussain, Q. Wu, C. Sun, and Y. Shan, “Mumu-llama: Multi-modal music understanding and generation via large language models,” *arXiv preprint arXiv:2412.06660*, 2024.

[4] A. Roucher, A. V. del Moral, T. Wolf, L. von Werra, and E. Kaunismäki, “‘smolagents‘: a smol library to build great agentic systems.” https://github.com/huggingface/smolagents, 2025.

[5] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou *et al.*, “Chatmusician: Understanding and generating music intrinsically with llm,” *arXiv preprint arXiv:2402.16153*, 2024.

[6] Y. Wang, S. Wu, J. Hu, X. Du, Y. Peng, Y. Huang, S. Fan, X. Li, F. Yu, and M. Sun, “Notagen: Advancing musicality in symbolic music generation with large language model training paradigms,” *arXiv preprint arXiv:2502.18008*, 2025.

[7] A. Goel, S. Ghosh, J. Kim, S. Kumar, Z. Kong, S.-g. Lee, C.-H. H. Yang, R. Duraiswami, D. Manocha, R. Valle *et al.*, “Audio flamingo 3: Advancing audio intelligence with fully open large audio language models,” *arXiv preprint arXiv:2507.08128*, 2025.

[8] Y. Chu, J. Xu, X. Zhou, Q. Yang, S. Zhang, Z. Yan, C. Zhou, and J. Zhou, “Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models,” *arXiv preprint arXiv:2311.07919*, 2023.

[9] Z. Evans, J. D. Parker, C. Carr, Z. Zukowski, J. Taylor, and J. Pons, “Stable audio open,” in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[10] F. Foscarin, E. Karystinaios, E. Nakamura, and G. Widmer, “Cluster and separate: a gnn approach to voice and staff prediction for score engraving,” *arXiv preprint arXiv:2407.21030*, 2024.

[11] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou, “Gradio: Hassle-free sharing and testing of ml models in the wild,” *Presented at 2019 ICML Workshop on Human in the Loop Learning (HILL 2019), Long Beach, USA*, 2019.

[12] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, “Qwen3 technical report,” *arXiv preprint arXiv:2505.09388*, 2025.