MusiCoLab: Towards a Modular Architecture for Collaborative Music Learning

Chrisoula Alexandraki
Hellenic Mediterranean University
chrisoula@hmu.gr

Demosthenes Akoumianakis
Hellenic Mediterranean University
da@hmu.gr

Michael Kalochristianakis
Hellenic Mediterranean University
kalohr@hmu.gr

Panagiotis Zervas
University of Peloponnese
p.zervas@go.uop.gr

Maximos Kaliakatsos-Papakostas
Aristotle University of Thessaloniki
maxk@mus.auth.gr

Emilios Cambouropoulos
Aristotle University of Thessaloniki
emilios@mus.auth.gr

ABSTRACT

This paper presents current developments taking place in the context of the MusiCoLab project. MusiCoLab aims at delivering a comprehensive and efficient web platform for music learning and teaching, by building on prior research experience of project partners, as well as by investigating the integration of state-of-the-art tools in intelligent music composition, performance, and discovery within an educational context. Compared to existing relevant initiatives, MusiCoLab offers a suite of innovative tools that may be used to enhance collaboration and engagement in networked/virtual settings. These tools are sought both in the context of asynchronous student/teacher interactions (i.e., course preparation and scheduling, student assignments and self-practice) as well as synchronously, i.e., serving as groupware to facilitate live music lessons by manipulating intelligent collaborative digital artifacts.

1. INTRODUCTION

The Internet is vastly generous in offering resources for music learning. Interested users (i.e., teachers and students) may seek for video performances in general purpose platforms (e.g., Youtube), backing tracks for play-along practicing in music content providers (e.g., Aebersold Spotify Backing Tracks, Hal Leonard Play-along series, Music Minus One), and sheet music in relevant digital collections and databases, such as MusicBrainz, Mutopia, MuseOpen, etc. Moreover, there is a plethora of music lessons offered by Massive Open Online Course (MOOC) platforms, such as Udemy and Coursera. These courses provide filmed lectures, quizzes, and student assignments of immediate feedback as well as interactive courses, in which the interaction capabilities are commonly offered via social media platforms and discussion groups. Additionally, there are commercial platforms for music learning that offer dedicated applications and resources. For example, Yousician[1], focuses on instrument performance learning for novice students, Soundslice[2] provides a web application offering video performances aligned with sheet music that may be adapted in tempo, Genius Jamtracks[3] provides play-along Jazz tracks of adjustable rhythmic and harmonic complexity and so on.

Finally, there rather few academic initiatives for music learning, most of which base their rationale on the paradigm of blended learning, a.k.a. hybrid learning, which combines opportunities for online interaction with conventional place-based classrooms. For example, the Intermusic project, funded by the Erasmus+ program, aims at establishing European partnerships in music learning, while other initiatives aim at drawing case studies, experiments, and evaluations of blended and autonomous music learning [4][6][8].

In short, in an attempt to classify online resources for music learning, one can discern online music repositories, MOOC-based music lessons, commercial platforms for music learning and finally platforms developed in the context of research initiatives and academic partnerships. Compared to these initiatives, the MusiCoLab environment focuses on the collaborative nature of music learning. From a pedagogical point of view, MusiCoLab adheres to educational theories of social constructionism and constructivism [9]. These theories state that people work together to construct artifacts and that learning of individuals takes place because of the social/collaborative interaction with these artifacts.

MusiColab aims at increasing student engagement and learning by developing shared knowledge through a common educational workspace that allows the construction-of as well as the interaction-with digital artifacts. These artifacts are represented as audio files and symbolic music notation files, which are supplemented by descriptive metadata. The environment allows artifact generation and processing either through synchronous collaboration, or through asynchronous interaction activities. In addition to interaction and collaboration, MusiCoLab incorporates intelligent mechanisms to improve the efficacy of the learning practice by providing automatic suggestions of artifact annotations, which include descriptive metadata as well as structural music elements (e.g., beats, chords). These automatic suggestions serve to enhance user engagement, by promoting a curating, rather than a manual,

[1] https://yousician.com/

[2] https://www.soundslice.com/

[3] https://geniusjamtracks.com/

Web Audio Conference WAC-2022, July 6–8, 2022, Cannes, France.

often labor-intensive, perspective, thus allowing to concentrate on artifact manipulations that are most relevant to the learning process.

The rest of this article is structured as follows. The next section presents the overall architecture of the MusiCoLab environment and describes its constituent modules. Section 3 presents implementation details and presents two use cases that describe how different modules may interoperate to support a) asynchronous learning activities and b) synchronous collaborative music lessons. Finally, section 4 concludes the paper by presenting the current state of developments and by elucidating future development plans that will allow to enhance the collaborative nature of music learning.

2. OVERALL ARCHITECTURE

As depicted on Figure 1, the central entities of the platform comprise a Collaboration Server, a Conferencing Server, and a Music Generation Server.

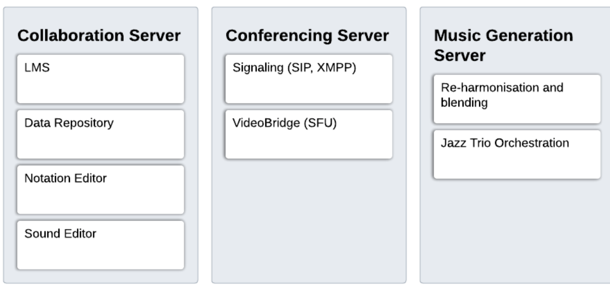

Figure 1: The central unities of the MusiCoLab platform

The Collaboration Server, essentially an HTTP server, provides the Learning Management System (LMS), the Data Repository for storing and sharing digital artifacts, the Notation Editor, i.e. an interactive tool for collaborative score composition and manipulation as well as the Sound Editor, integrating the functionalities of sound recording and audio signal annotating.

The modules of the Collaboration Sever may be used in combination with the Conferencing server to enable synchronous collaborative artifact manipulation. The conferencing server is a web-based conferencing application based on the WebRTC architecture [5].

Finally, the Music Generation Server represents the creative part for musical resources. It provides music orchestration examples as well as re-harmonisations of symbolic notation artifacts that may be retrieved either synchronously, i.e., during a collaborative learning session or asynchronously to provide learning examples that may be manipulated or reconfigured according to the required setup.

The following subsections present the constituent modules of the three server components.

2.1 Learning Management System

To date LMSs are indispensable components, not only of e-learning environments, but also for conventional classroom-courses. Especially, since the years of the pandemic, teachers and students of all levels became acquainted to the idea of using LMSs for classroom management, online content distribution, course scheduling, student assessment, user feedback and communication. Nowadays several software packages implement LMS functionalities and may be provided either as part of commercial applications or as open-source solutions. MusiCoLab uses Moodle[4] for course creation and management. Moodle is amongst the most popular LMSs worldwide, as it is open source, constantly updated and it allows for extending and tailoring learning environments by using community-sourced plugins.

From a user perspective, the Moodle LMS is the starting point of the MusiCoLab environment. Users register to the environment through the LMS, which allows LMS administrators to assign user roles such as course creator, teacher, and student. Through the LMS users are authenticated to the remaining components of the MusiCoLab environment including the Data Repository, the Notation Editor, the Sound Editor, the Conferencing Server and the Music Generation Server.

2.2 Data Repository

The data repository of the MusiCoLab environment has been designed to supplement the LMS by implementing file management processes that are more flexible than those provided by the LMS itself, and more suitable to music content with respect to inscribing metadata and providing tools for content generation, analysis, and processing. It has been implemented as a web application that is interoperable with the Moodle LMS, in terms of user authentication and file resource sharing. Unlike the LMS, the data repository allows teachers as well as students to maintain a public collection of file resources, in addition to the private one and the files-per-course, that are usually offered by LMSs. The public collection allows distribution of musical artifacts to the entire MusiCoLab community.

In fact, each user is permitted to access three collections of files, i.e. is his/her private files, the public files and the LMS files. Each file of the repository has an owner, i.e., the user that has created or uploaded the file. Owners have full read/write access to their files. Files may be transferred between these three collections. Moving or copying a file to the LMS allows making that file available to the courses supported by the LMS.

Table 1: Descriptive metadata of music learning material.

Label	Auto	Multiple values	Pre-defined
Music Genre	yes	no	yes
Title	no	no	no
Composer	no	no	no
Performers	no	yes	no
Instruments	no	yes	yes
Tonal Key/Scale	yes	no	yes
Content (Voiced/Instrumental)	yes	no	yes
Tempo (bpm)	yes	no	no

Besides file collections, the repository includes a database, which maintains file descriptions. File descriptions comprise file information and music content descriptors (i.e., metadata). File information is automatically derived from file header, while

4 https://moodle.org/

metadata may be either manually submitted or automatically derived using the auto tagging algorithms of the Essentia open-source library for music analysis and description [2]. Automatic retrieval of metadata is requested by file owners and provided as suggestions, that owners may accept or modify.

Table 1 lists the musical metadata used for describing musical artifacts in the repository. The columns indicate which of these properties can be automatically retrieved by analysis, which accept multiple values in the database, and which may take their value from a pre-defined set of options, e.g. provided in a dropdown list. Musical metadata are used for efficient search and retrieval of music learning material. For example, a music instructor may wish to perform a search in the public repository for jazz pieces containing vocals and having a tonality of F-minor.

Figure 2 provides a screenshot of the repository functionality for audio file rendering. The menu in the left side of the waveform has been intentionally kept in the figure to depict the intended functionalities, e.g. recording searching and tagging, most of which are currently under intense development. Audio file rendering uses the Sound Editor component (please refer to Figure 1), which uses the wavesurfer-js[5] open-source tool for waveform visualization. The sound editor provides a similar interface for sound recording. Descriptive metadata for the specific file are provided below the waveform.

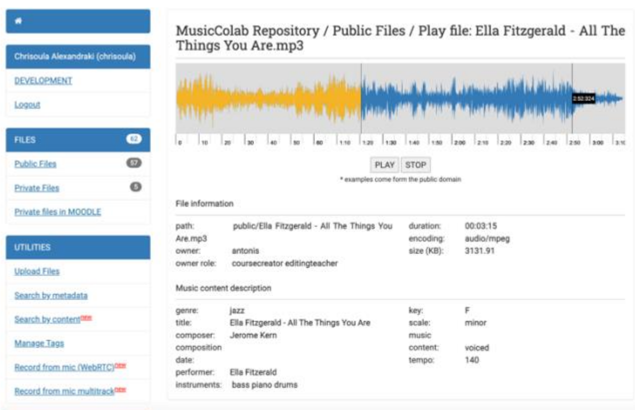

Figure 2: Audio file rendering in the repository.

The same file descriptions apply to symbolic music files. Currently supported formats are MIDI, Kern, MusicXML and MEI. Equivalently to the wavesurfer-js, we use Verovio Humdrum Viewer (VHV)[6], which is an open-source file notation editor that renders symbolic files as Support Vector Graphics (SVG) [10]. Musical metadata for symbolic music are currently provided as manual annotations, but we plan to investigate the use of music21[3], to automatically derive descriptions such as instruments, tonality, tempo, and time signature.

An additional functionality of the repository is the fact that users may provide hashtags for each artifact, therefore allowing to cross-reference music content based on topic or lesson.

2.3 Audio/Video Conferencing

In network-mediated music collaboration a plausible requirement is to support simultaneous music performance among connected peers. Unfortunately, this requirement is hard to address in commonly available network infrastructures and even in highly reliable ones (e.g. academic networks), when communications are performed over long geographical distance. Unlike verbal communication, music performance interactions have extremely low tolerance to communication latencies (an order of magnitude lower than speech) and demand for higher upload and download capacities, as well as elimination of audio signal distortions owing to network packet loss. Research efforts in the direction of Networked Music Performance (NMP) systems aim to address these challenges by devising workarounds, and determining setups, for which synchronous collaborative music performance may become feasible [1].

However, in a different direction of research and vigorous technological development Voice-over-IP (VoIP) technologies are currently capable of providing video conferencing capabilities through web browsers. Currently, most videoconferencing systems provide a web-based client application implemented on the Web Audio API and the WebRTC architecture. Unluckily, these technologies do not allow configuring client hardware, i.e. the soundcard, to the detail that would permit lower-latency communications, which is the fundamental requirement for NMP systems [11].

The MusiCoLab environment prioritizes collaboration in learning sessions, rather than music performance sessions. In music learning it is rather uncommon for connected peers to simultaneously perform the same piece of music. It is instead more common that a teacher or a student will perform a musical excerpt, and others will be required to imitate or discuss the performance. As the MusiCoLab environment is delivered entirely over the Web, it uses a WebRTC solution, namely Jitsi[7] open-source multiplatform VoIP software to facilitate verbal communication between communicating peers, as well as non-simultaneous music performance. Jitsi provides built-in signaling and messaging protocols (SIP, XMPP), as well as a Videobridge (JVB) allowing to relay audio and video streams through a central network node, hence eliminating the demand for excessive upload bandwidth in multi-peer calls. With respect to audio, Jitsi uses the Opus codec [13] and it permits configuring Opus to disable time consuming audio processes (e.g. Audio Echo Cancellation, transmission of Redundant Audio Data) and also increasing the maximum average bitrate of audio streams, thereby allowing to exchange full band music signals (as opposed to narrowband speech-quality signals) whenever the network infrastructure can provide sufficient resources.

Improved music communication in the MusiCoLab environment is currently under intense investigation and development. Research efforts are oriented towards implementing a client application to permit low-latency soundcard configuration and customize Jitsi to communicate with that application for more efficient audio signal delivery.

2.4 Notation Editor

As previously mentioned, artifacts of symbolic notation are rendered as SVG using a custom implementation of VHV, i.e., the Notation Editor component of Figure 1. Besides SVG rendering, VHV offers a web interface, which includes a text editor that displays symbolic files in Kern, MusicXML or MEI. Editing a file in the text editor modifies the SVG score to reflect the desired modifications. Moreover, VHV allows sound rendering of the score

5 https://wavesurfer-js.org/

6 https://github.com/humdrum-tools/verovio-humdrum-viewer

7 https://jitsi.org/

file, which is achieved by converting symbolic files to MIDI and using MIDI compliant sound fonts. Especially for Kern files, VHV allows graphical editing of note pitches, slurs, and beams.

In MusiCoLab, VHV was extended to support collaborative editing via the Yjs[8] open-source framework, which provides a Conflict-free Replicated Data Types (CRDT) implementation. During a collaborative session of two or more participants, CRDTs allow editing activities performed at one site to be replicated to the remaining participants. The MusiCoLab Notation Editor provides collaborative activities, which include graphic editing of pitches, slurs, beams, highlighting a segment of the graphical score (rubber-banding) and adding comments. Figure 3 provides a screenshot of the Notation Editor in which two users, Dimitris and George, negotiate on the preparation of a score to be used on a synchronous collaborative music lesson.

Figure 3: The Notation Editor.

Moreover, as the Jazz music genre is used as an inspiring case for MusiCoLab investigations, the collaborative score allows selecting a chord symbol and changing it to derive a new chord progression. Figure 4 displays the interface used for chord editing.

The Notation Editor is invoked via hyperlinks provided in courses of the LMS or via the Data Repository, whenever requesting to view or edit a digital artifact of symbolic music.

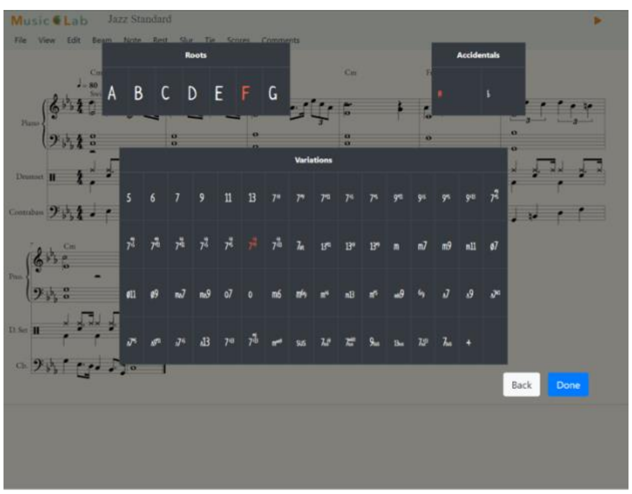

Figure 4: Chord editing functionalities integrated in VHV.

2.5 Music Generation Server

The aim of the Music Generation Server (MGS) is to provide students and teachers with the ability to have new, original music available as an accompanying backing track, based on the skills of the student. The (MGS) achieves this by combining two tools that generate musical material on different levels: i) the Chameleon[9] melodic harmonization assistant provides alternative re-harmonisations of user-selected initial chords (e.g., chords of a specific jazz standard); and ii) the Genius Jam Tracks[10] renders user-provided or Chameleon-generated harmonies into jazz trio music (i.e., drums, bass and piano parts). Both tools are invoked by the Notation Editor of the Collaboration Server to receive information from the user about the chords that need to be processed (please refer to Figure 4) and return a new symbolic file to be rendered.

The Chameleon melodic harmonization assistant [7] learns harmonies from diverse musical styles and generate novel harmonisations of given melodies. The user may select the style of those harmonisations or select a 'blended' style that Chameleon creates, based on two styles selected by the user. Blended styles incorporate the general principles of Conceptual Blended theory, whereby a new, blended style encompasses characteristics from both inputs but can also be identified as an altogether new style [14]. In the MusiCoLab version of Chameleon, the system learns individual jazz standards (instead of styles comprising multiple pieces) and can, therefore, either reharmonize melodies in the 'style' of individual, user selected, jazz standards, or blend pairs of user selected jazz standards. Chameleon gives interesting possibilities for learning: students can focus their study on variations or combinations of harmonic phenomena that are currently being studying. This possibility makes studying jazz improvisation more engaging and diversified, allowing harmonic exploration and experimentation.

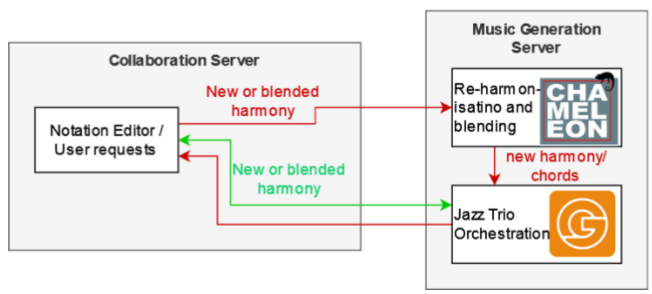

Figure 5: Overview of Music Generation Server

Genius Jam Tracks (GJT) is a commercial iOS application that complements the functionality of Chameleon. While Chameleon employs rudimentary algorithms for rendering harmonies, in the form of chord sequences, GJT has sophisticated AI algorithms that involve large amounts of human-played music for rendering chords in the style of a jazz standard drums-bass-piano trio. An implementation-level challenge of the MGS is the fact that GJT algorithms and data are proprietary and, therefore, Chameleon cannot be integrated directly with the GJT application. To this end, data exchange and communication between all involved components must be performed through API calls between independent servers; this architecture is abstractly shown in Figure 5. The green line indicates direct music rendering of the current chords that the user sees in the Notation Editor (only GJT is

8 https://github.com/yjs/

9 http://ccm.web.auth.gr/chameleonmain.html

10 https://geniusjamtracks.com/

involved in this process), while the path shown with the red lines involves re-harmonization and/or blending from Chameleon.

3. Implementation and interoperability

The implementation of the MusiCoLab environment uses several open-source tools and APIs, which have been customized and integrated to better address the requirements of the foreseen collaborative learning scenarios. The authors of this paper would like to express their sincere appreciation to the developers of these tools as well as the wider open-source developer community, without the support of which, the MusiCoLab environment would not have been realized. Table 2 provides a list of the open-source tools that were used as the baseline modules of the MusiCoLab architecture. Collectively, these modules provide a suite of tools that are capable of exchanging data and may be combined in various setups to facilitate synchronous and asynchronous teaching and learning.

Table 2. Open-source tools used for the realization of the MusiCoLab environment.

SW Tool	MusiCoLab Module
Moodle	LMS
Jitsi	AV Conferencing
VHV	Symbolic artifact manipulation
Wavesurfer	Waveform visualisation, editing and annotation
Essentia	Automatic audio analysis and tagging
Music21	Automatic tagging of symbolic artifacts
Yjs	Collaborative manipulation of digital artifacts over the web

In any such setup, a user logs on to the LMS. The LMS provides users with hyperlinks pointing to the Data Repository and to the Conferencing Server. These links open a new tab of the Internet Browser. The former provides affordances supporting practices employed in asynchronous learning, while the latter offers collaborative affordances to support synchronous music lessons. Modules focusing on musical artifact creation, generation and processing (i.e. the Sound Editor, the Notation Editor) are represented as iFrames either to the user interface of the Data Repository or to that of the Video Conferencing Server. These iFrames may be activated or deactivated on demand. For example, Figure 2 shows the sound editor component used within the Data repository, to render, record or annotate an audio artifact stored in the repository. Figure 6 shows the Notation Editor collaboratively manipulated by two users participating in a synchronous videoconference session hosted by Jitsi. Artifacts that are manipulated either through the repository or through the videoconferencing server may be stored back in the Data Repository for preservation and community sharing. Community sharing is achieved either by moving an artifact to the public collection of the Data Repository, or by exporting that artifact from the Data Repository to the LMS files and using it in the context of a specific course offered by the LMS.

The following subsections present two example use cases, one for a synchronous collaborative lesson and one for asynchronous student practicing, to illustrate the intended purpose of each module of the MusiCoLab architecture.

3.1 Synchronous Lesson

Antonis, a competent jazz saxophonist and improvisation instructor, creates on MusiCoLab LMS a course entitled 'Jazz Improvisation' and posts an announcement to the MusiCoLab community about admitting student enrollments until July 10. He schedules the first introductory course on Monday July 12 at 11:00am. He posts a second announcement to provide a hyperlink to the room of the videoconferencing server, where the lesson is about to take place.

During the lesson, participants use headphones to isolate their microphone input from audio output, not only to avoid feedback (please note that as mentioned in section 2.3 Audio Echo Cancellation is disabled), but also to allow making recordings of the participants' isolated performance. Lesson participants use the interface depicted on Figure 6, which comprises two iFrames. The top iFrame accommodates the videoconferencing session, while bottom one accommodates the collaborative Notation Editor. Antonis uses the Notation Editor to open a jazz standard accompaniment, e.g. the piece 'Autumn Leaves', in the Notation Editor, which is rendered in jazz trio format by the MGS. Then, he improvises on the accompaniment to provide students with a demo solo performance. While Antonis improvises on top of the MIDI-rendered accompaniment, he also records his performance to make it available to the students for future reference. Afterwards, Antonis asks each student to improvise on the jazz accompaniment and provides oral feedback. He adds comments about each student performance on the score and highlights certain musical passages that need to be practiced repeatedly (please also refer to Figure 3).

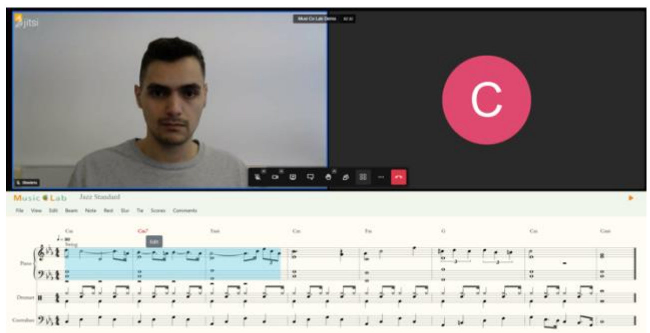

Figure 6: The user interface of a synchronous music lesson.

At the end of the lesson, Antonis uploads the annotated score on the public collection of the repository. The annotations are stored in JSON Strings which are inserted in the database of the Data Repository. He also uploads his solo recording on his private collection on the repository and from there he exports it to the LMS as a resource of the 'Jazz Improvisation' course (e.g. Lesson1_DemoSolo.mp3). Finally, he creates an LMS student assignment for the second lesson by providing hyperlinks for: a) the annotated 'Autumn Leaves' score residing in the public collection of the Data Repository and b) his solo recording residing in the collection of course files of the LMS.

3.2 Asynchronous Student Practice

Mary, a piano student who has enrolled to Antonis' course on Jazz Improvisation wishes to practice the piece performed on the first lesson. She logs on to the LMS, finds the assignment created by Antonis. She opens the annotated 'Autumn Leaves' score using the Notation Editor (Figure 3) and listens the MIDI-rendered accompaniment. Then, she opens the Sound Editor and listens to the reference improvisation provided by Antonis during the lesson. She practices with the accompaniment repeatedly. Since practicing repeatedly on the same chords can become tedious, Mary may also request re-harmonisation for some chords (Figure 4), or replacement of some chords by blending with another jazz standard. When Mary feels confident enough with her improvisation, she records her performance and uploads the recording to her private files collection of the Data Repository. Finally, that recording is exported from the repository to the LMS assignment that was created by Antonis.

4. CONSOLIDATION

This article presents research and development efforts on the implementation of a modular architecture for music education. The theoretical motivations of this effort have emerged from the theories of social constructionism and social constructivism, which state that human development and knowledge is constructed through the interaction with others to create artifacts. The main artifacts of distant music learning are digital sound recordings and resources of symbolic music that may be composed or manipulated to reflect the shared understanding of what is correctly interpreted by the group and what may be incorrectly understood by individuals. To this end, MusiCoLab provides the tools to create, automatically generate, comment, annotate, process, and manipulate digital music artifacts.

The MusiCoLab project commenced in June 2021. At the time of this writing, the project is going through the first year of implementation. The presented ideas and software modules are under vigorous investigation and development. A number of modules (e.g. the Sound Editor) need further development to address their intended purpose. Due to developments being in preliminary state, the paper does not present a user evaluation. It is however acknowledged that, to better understand the requirements of teachers and students, several pilot experiments need to take place. These experiments have been scheduled to take place in the forthcoming year and motivate further ongoing development.

Besides adhering to theoretical motivations of artifact construction and addressing the requirements of distant music education, a further scientific priority of the MusiCoLab architecture, concerns the perspective of offering a testbed allowing to assess the usability of state-of-the-art advances in Music Information Retrieval (MIR) in the context of music education. In this direction, the Data Repository currently facilitates algorithms for the automatic inference of descriptive metadata of music artifacts and for content-based search, i.e. providing users with a facility for searching artifacts by providing an excerpt of their musical performance. An additional MIR task being currently considered concerns the detection of structural music components, such as chords and beats to be provided as suggestions for the analyses of sound files.

It is widely acknowledged that in MIR there is no there is no one-algorithm-fits-all solution for every type of music content. For example, genre, tempo and instrumentation severely affect the performance of detection algorithms. For this reason, a plausible criticism to incorporating such functionalities in music education systems, is the possibility of providing false or inaccurate information to students. To account for this problem, the MusiCoLab environments adopts a suggestion-based approach to allow users requesting automatically retrieved information that they may accept, reject, or edit after retrieval. It is expected that these suggestions, not only improve the efficacy of managing music content, but also enhance user engagement by promoting a curator rather than a manual annotator, often labor-intensive, role.

The implementation of such affordances is also foreseen in the domain of symbolic music analysis. To date, software packages for musicological analysis, such as the music21 [3] python package may significantly elevate the level of musical understanding. Furthermore, we expect to integrate methodologies of automatic score following by presenting MusiCoLab users with the possibility of monitoring their instant performance on the score or by navigating within their audio recording to a structural element indicated on the score (e.g. go to the 2nd beat of the 3rd measure) using an interface that encompasses the Notation Editor and the Sound Editor.

Further to content-based retrieval of music information from artifacts, the Data Repository provides the possibility of creating hashtags to allow for user-contributed metadata and enable cross-referencing artifacts by topic or by learning context. Collectively, these affordances make the MusiCoLab environment an ideal platform for music pedagogy offered in the context of the Internet of Musical Things (IoMusT). The IoMusT is an emerging field of research dedicated to providing ubiquitous access to music content trough *musical things*, which may be smart musical instruments, smartphones, and wearables [12]. A distinct priority of IoMusT relates to offering improved services for music pedagogy and in fact the MusiCoLab architecture can be tailored to address this near-future perspective.

To summarise, MusiCoLab envisages to provide a comprehensive solution for distant music education and through this endeavor manifest new scientific challenges and innovative research possibilities that are yet to be realised.

5. ACKNOWLEDGMENTS

This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE. Project Acronym: MusiCoLab, Project Code: T2EDK-00353.

6. REFERENCES

[1] Alexandraki, C., 2019. Experimental Investigations and Future Possibilities in Network-Mediated Folk Music Performance. In: Bader R. (eds) *Computational Phonogram Archiving. Current Research in Systematic Musicology, vol 5. Springer*, Springer International Publishing pp. 207–228. doi:10.1007/978-3-030-02695-0_10

[2] Alonso-Jiménez, P., Bogdanov, D., ... Serra, X., 2020. Tensorflowaudio models in essentia, in: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings.* pp. 266–270. doi:10.1109/ICASSP40776.2020.9054688

[3] Ariza, C., Cuthbert, M.S., 2011. the Music21 Stream: a New Object Model for Representing, Filtering, and Transforming Symbolic Musical Structures. *Proceedings of the International Computer Music Conference 2011*, 61–68.

[4] Barate, A., Ludovico, L.A., 2020. An open and multi-layer web platform for higher music education. *Journal of E-Learning and Knowledge Society* 16, 29–37. doi:10.20368/1971-8829/1135356

[5] Blum, N., Lachapelle, S., Alvestrand, H., 2021. WebRTC. Communications of the ACM 64, 50–54. doi:10.1145/3453182

[6] Jiemsak, R. 2021. The Efficiency of Music Learning in a Blended Learning Environment of Undergraduate Students In Phranakhon Si Ayutthaya Rajabhat University. 2021 *6th*

International STEM Education Conference (iSTEM-Ed), 2021, pp. 1-4, doi: 10.1109/iSTEM-Ed52129.2021.9625071.

[7] Kaliakatsos-Papakostas, M., Queiroz, M., Tsougras, C. and Cambouropoulos, E., 2017. Conceptual blending of harmonic spaces for creative melodic harmonisation. *Journal of New Music Research*, *46*(4), pp.305-328.

[8] Kelkar, T., Ray, A., Choppella, V., 2015. SangeetKosh: An open web platform for music education, in: *Proceedings - IEEE 15th International Conference on Advanced Learning Technologies*, pp. 5–9. doi:10.1109/ICALT.2015.102

[9] Knapp, N.F., 2019. The Shape Activity: Social Constructivism in the Psychology Classroom. *Teaching of Psychology* 46, 87–91. doi:10.1177/0098628318816181

[10] Pugin, L., Zitellini, R., Roland, P., 2014. Verovio: A library for engraving MEI music notation into SVG, in: *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014*. pp. 107–112.

[11] Sacchetto M., Servetti A. and Chafe C. 2021. JackTrip-WebRTC: Networked music experiments with PCM stereo audio in a Web browser, in: *Proceedings of the Web Audio Conference 2022.*

[12] Turchet, L., Fischione, C., Essl, G., Keller, D., & Barthet, M. (2018). Internet of Musical Things: Vision and Challenges. IEEE Access, 6, 61994–62017. https://doi.org/10.1109/ACCESS.2018.2872625

[13] Valin, J.M., Maxwell, G., ... Vos, K., 2013. High-quality, low-delay music coding in the opus codec, in: 135th Audio Engineering Society Convention 2013. Audio Engineering Society, pp. 73–82.

[14] Zacharakis, A., Kaliakatsos-Papakostas, M., Tsougras, C. and Cambouropoulos, E., 2018. Musical blending and creativity: An empirical evaluation of the CHAMELEON melodic harmonisation assistant. *Musicae Scientiae*, *22*(1), pp.119-144.