

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΜΟΥΣΙΚΗΣ ΚΑΙ ΟΠΤΟΑΚΟΥΣΤΙΚΩΝ ΤΕΧΝΟΛΟΓΙΩΝ

Τμήμα Μουσικής Τεχνολογίας και Ακουστικής



Πτυχιακή εργασία

**Εργαλεία και εφαρμογές μουσικής σημειογραφίας για την
υποβοήθηση της μουσικής εκπαίδευσης**

Αλέξανδρος Νούσιος (ΤΑ2238)

Επιβλέπουσα:

Χρυσούλα Αλεξανδράκη
Επίκουρη Καθηγήτρια

Ρέθυμνο, Σεπτέμβριος 2022

Περίληψη

Η πτυχιακή εργασία στοχεύει στην ανάπτυξη μιας Διαδικτυακής Εφαρμογής (Web Application), η οποία αξιοποιεί τεχνολογίες μουσικής σημειογραφίας προκειμένου να υποβοηθήσει τη μουσική εκπαίδευση από απόσταση.

Ειδικότερα, μελετώνται μορφότυποι αρχείων (file formats) καθώς και εργαλεία γραφικής και ηχητικής απόδοσης μιας παρτιτούρας καθώς και το πώς αυτά μπορούν να καταστούν επεξεργάσιμα, ώστε να είναι χρήσιμα σε ένα σενάριο μουσικής εκπαίδευσης.

Ενδεικτικοί μορφότυποι αφορούν σε αρχεία MIDI, Humdrum, ABC, MusicXML, και MEI. Έπειτα από μια πρώτη διερεύνηση έχει καταστεί σαφές ότι τα αρχεία αυτά δεν είναι ισοδύναμα ως προς την πρωτογενή πληροφορία που περιέχουν, δηλαδή αυτή καθαυτή την περιγραφή μουσικού περιεχομένου. Η πτυχιακή περιλαμβάνει εκτίμηση των ιδιαιτεροτήτων κάθε τέτοιου διαφορετικού μορφότυπο.

Επιπρόσθετα, μελετώνται εργαλεία γραφικής και ηχητική απόδοσης, σε μια προσπάθεια να βρεθούν τα βέλτιστα εργαλεία και δικτυακές εφαρμογές, που μπορούν να υποστηρίξουν τόσο ιδιόζουσες μορφές παρτιτούρας, όσο και μηχανισμούς αλληλεπίδρασης με αρχεία μουσικής σημειογραφίας. Ένα τέτοιο ενδεικτικό εργαλείο είναι το Verovio Humdrum Viewer¹, το οποίο είναι ανοικτού κώδικα και στο οποίο υλοποιήθηκαν ορισμένες προσαρμογές.

Το αποτέλεσμα της εργασίας είναι μια καινοτόμος εφαρμογή για τη μουσική εκπαίδευση η οποία εκτελείται σε περιηγητή (browser), χωρίς να υπάρχουν απαιτήσεις σε ειδικό λογισμικό και συγκεκριμένο λειτουργικό σύστημα. Έτσι, η εφαρμογή μπορεί να ενσωματωθεί σε Συστήματα Διαχείρισης Μάθησης, όπως για παράδειγμα στο Open eClass ή το Moodle.

¹ <https://verovio.humdrum.org/>

Abstract

This work aims at the development of a web application, which investigates the use of music notation technologies, in online music education.

Specifically, the work presents various file formats and score rendering (both audio and graphic rendering) toolkits and investigates their capacity to effectively support the requirements of online music learning.

To date, there are several music notation formats. This study focuses on MIDI, Humdrum, ABC, MusicXML and MEI. Following an initial investigation, it was found out that such file formats are neither equivalent in terms of their information content, nor interchangeable in their graphic or auditory representation. As a distinct contribution, this work provides a comparative presentation of their specificities.

Furthermore, several score rendering toolkits and applications are studied, in an attempt to assess their efficiency in interaction and collaboration capabilities required in the context of online music learning. The study focuses on Verovio Humdrum Viewer (VHV), which is an open-source web application. The implementation of this study includes certain extensions of VHV to improve the efficacy its efficacy for online music learning.

The outcome of this work is a novel web application for music education which is accessible through an Internet browser. Besides not requiring any prior software installation, the developed application may be integrated and used through Learning Management Systems, such as Open eClass or Moodle.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. Εισαγωγή	6
1.1 Στόχος	6
1.2 Μουσική Εκπαίδευση από Απόσταση	6
1.3 Μουσική πληροφορία και εργαλεία λογισμικού	7
1.4 Δομή Εργασίας	9
2. Πρότυπα περιγραφής μουσικού περιεχομένου	11
2.1 Γενικά	11
2.2 MIDI	11
2.3 ABC Notation Format	14
2.4 Kern και Humdrum	15
2.5 MusicXML	18
2.6 MEI	19
2.7 Σύνοψη	20
3. Εργαλεία μουσικής σημειογραφίας	23
3.1 Προγραμματιστικές βιβλιοθήκες (API's)	23
3.1.1 Music21	23
3.1.2 Partitura	24
3.1.3 Jmusic	25
3.1.4 VexFlow	26
3.1.5 Verovio	28
3.2 Πακέτα λογισμικού/εφαρμογές	29
3.2.1 MuseScore	29
3.2.2 Lilypond	30
3.2.3 Verovio Humdrum Viewer	32
3.3 Σύνοψη	33
4. Υλοποίηση	35
4.1 Το VHV στη μουσική εκπαίδευση	35
4.2 Επεξεργαστής Συγχορδιών	35
4.3 Αλλαγή ταχύτητας αναπαραγωγής	41
4.4 Υπολογισμός χρονικής αλληλουχίας	43
4.5 Διάθεση κώδικα στο GitHub	46
5. Συμπεράσματα	47
5.1 Σύνοψη	47

5.2	Μελλοντικές επεκτάσεις	47
5.2.1	Δημιουργία παρτιούρας	47
5.2.2	Sequencer	48
5.2.3	Συνεργατική αλληλεπίδραση με παρτιούρες	48
6.	Αναφορές	49
7.	Παραρτήμα Α	51

1. ΕΙΣΑΓΩΓΗ

1.1 Στόχος

Η εργασία αυτή εστιάζει στη μελέτη προτύπων περιγραφής μουσικού περιεχομένου, όπως αυτό αποτυπώνεται και διανέμεται μέσω τεχνουργημάτων μουσικής σημειογραφίας (παρτιτούρες). Επιπρόσθετα, μελετώνται εργαλεία λογισμικού που επιτρέπουν την επεξεργασία, τη σύνθεση, τη γραφική και την ηχητική απόδοση τέτοιων τεχνουργημάτων.

Η μελέτη αυτών των προτύπων και εργαλείων λογισμικού έχει ως στόχο τη συγκριτική τους αξιολόγηση και την αποτίμηση της καταλληλότητάς τους για την υποβοήθηση της μουσικής εκπαίδευσης από απόσταση.

Ως αποτέλεσμα αυτής της αποτίμησης, η εργασία ασχολείται με την υλοποίηση μιας σειράς πρόσθετων εργαλείων λογισμικού που επεκτείνουν υπάρχουσες λύσεις και μπορούν να χρησιμοποιηθούν για την παροχή εξειδικευμένων λειτουργιών στη μουσική εκπαίδευση.

Τα εργαλεία που αναπτύχθηκαν στο πλαίσιο της εργασίας αυτής φιλοξενούνται στην ανοικτού κώδικα εφαρμογή Veronio Humdrum Viewer και διατίθενται ελεύθερα μέσω του αποθετηρίου GitHub, ενώ έχουν πρόσφατα δημοσιευθεί στο διεθνές συνέδριο Web Audio Conference 2022 (Akoumianakis et al, 2022).

1.2 Μουσική Εκπαίδευση από Απόσταση

Η μάθηση από απόσταση έχει καθιερωθεί εδώ και μεγάλο χρονικό διάστημα στα σύγχρονα εκπαιδευτικά συστήματα, καθώς προσφέρει ευελιξία χρόνου στο μαθητευόμενο, διευκολύνει την επικοινωνία μεταξύ διδασκόμενου-διδάσκοντα και επιτρέπει, μέσω χρήσης πολυμέσων, ποικιλία τρόπων αλληλεπίδρασης κατά τη διάρκεια του μαθήματος. Η εκπαίδευση από απόσταση πραγματοποιείται είτε σύγχρονα με απευθείας αλληλεπίδραση των συμμετεχόντων ή ασύγχρονα με χρήση μαθησιακού υλικού και κατάλληλων εργαλείων λογισμικού από διδάσκοντες και διδασκόμενους.

Το διαδίκτυο παρέχει αφθονία επιλογών για τη μουσική εκμάθηση. Οι ενδιαφερόμενοι μπορούν να αναζητήσουν έτοιμες μουσικές εκτελέσεις σε πλατφόρμες γενικού σκοπού (πχ Youtube), παρτιτούρες σε ψηφιακές συλλογές και βάσεις δεδομένων (πχ MusicBrainz, MuseOpen), καθώς και προηχογραφημένα συνοδεία για μουσικό όργανο (accompaniment ή backing tracks) από παρόχους μουσικού περιεχομένου (πχ Aebersold Spotify, Music Minus One). Επιπρόσθετα, πλατφόρμες ανοιχτής μάθησης (πχ Udemy, Coursera) προσφέρουν πληθώρα μαθημάτων μέσω βιντεοσκοπημένων διαλέξεων και αλληλεπιδραστικών ερωτηματολογίων.

Τέλος, οι πλατφόρμες μουσικής εκπαίδευσης που έχουν δημιουργηθεί από ακαδημαϊκές πρωτοβουλίες (πχ Intermusic²) βασίζονται στη λεγόμενη μεικτή μάθηση που συνδυάζει τη συμβατική διδασκαλία σε αίθουσα, με τη μαθησιακή αλληλεπίδραση μέσω διαδικτύου. Στην τελευταία περίπτωση η εκπαίδευση υποστηρίζεται από Συστήματα Διαχείρισης Μάθησης

² <http://intermusicproject.eu/>

(πχ Moodle, eClass), τα οποία διαδραματίζουν καθοριστικό ρόλο στην οργάνωση του μαθησιακού υλικού, ενώ επιπρόσθετα μπορεί να παρέχουν εργαλεία για την πραγματοποίηση σύγχρονης διδασκαλίας (Akoumianakis et al, 2022).

Συμπληρωματικά με τα παραπάνω, το πρόσφατο ερευνητικό έργο MusiCoLab (Alexandraki et al, 2022) στοχεύει στην ανάπτυξη ενός περιβάλλοντος που θα προσφέρει σύγχρονη και ασύγχρονη μουσική εκπαίδευση, εξ' ολοκλήρου μέσω διαδικτύου. Τα αποτελέσματα της παρούσας πτυχιακής εργασίας θα αξιοποιηθούν από το έργο MusiCoLab το οποίο αποτελεί μια ακαδημαϊκή πρωτοβουλία που συντονίζεται από το Τμήμα Μουσικής Τεχνολογίας και Ακουστικής του Ελληνικού Μεσογειακού Πανεπιστημίου.

1.3 Μουσική πληροφορία και εργαλεία λογισμικού

Ως μουσική πληροφορία αναφέρεται όλη εκείνη η πληροφορία που περιγράφει, προσδιορίζει, ταυτοποιεί κάποιο μουσικό έργο. Η μουσική πληροφορία μπορεί είτε να περιγράφει το μουσικό περιεχόμενο, είτε να αφορά σε μακροσκοπικά μεταδεδομένα.

Τα μεταδεδομένα (matadata) αποτελούν πληροφορία που είναι ενιαία για το σύνολο του μουσικού έργου και γενικά διαδραματίζουν αναγνωριστικό-περιγραφικό ρόλο. Πρωτοεμφανίστηκαν ως δεδομένα κειμένου στη μορφή του CD-Text το 1996 και περιείχαν πληροφορία σε σχέση με το συνολικό άλμπουμ (όνομα, καλλιτέχνης, χρονιά) ή σχετικά με το κάθε ξεχωριστό κομμάτι του (όνομα, αριθμός κλπ). Στη συνέχεια λόγω της ευρείας διάδοσης ανταλλαγής μουσικών αρχείων MP3 μέσω δικτύων peer to peer, εξελίχθηκαν σύμφωνα με το πρότυπο ID3v1 ως προκαθορισμένες ετικέτες (artist, album...), καταλαμβάνοντας στο τέλος του MP3 αρχείου ένα πεδίο μεγέθους 128 Bytes. Το πρότυπο εξελίχθηκε ξανά (ID3v2) το 1998 και πλέον ορίζει τα μεταδεδομένα να τοποθετούνται στην αρχή των μουσικών αρχείων για να προεπισκοπούνται στον media player χωρίς να απαιτείται η πλήρης φόρτωση του αρχείου. Καταλαμβάνεται λοιπόν ένας χώρος έως 256MB με ετικέτες-“πλαίσια” μη προκαθορισμένου μεγέθους με μέγιστο όριο τα 16MB. Υπάρχουν 83 τύποι τυποποιημένων πλαισίων στους οποίους περιλαμβάνονται εκτός των κλασικών ετικετών του ID3v1 και πεδία εκτεταμένης περιγραφικότητας του μουσικού αρχείου όπως εξώφυλλο του άλμπουμ, στίχοι, στοιχεία για πνευματικά δικαιώματα και άλλα. Σήμερα χρησιμοποιούνται και άλλα πρότυπα μεταδεδομένων για μορφοποιήσεις αρχείων πέραν της MP3.

Τα μεταδεδομένα εντάσσονται σε τρεις κατηγορίες. Τα περιγραφικά (descriptive metadata) χρησιμεύουν για αναζήτηση, οργάνωση και ταξινόμηση των μουσικών αρχείων σε μουσικές βιβλιοθήκες, για εκπομπή σε ραδιοφωνικούς σταθμούς καθώς και για αναγνώριση του μουσικού έργου. Τα μεταδεδομένα ιδιοκτησίας (ownership metadata) εισήχθησαν για λόγους αναγνώρισης πνευματικών δικαιωμάτων αλλά και για λόγους προώθησης και αναγνωρισιμότητας των συντελεστών που συμμετέχουν σε μια μουσική παραγωγή. Τέλος τα μη προκαθορισμένα μεταδεδομένα σύστασης (recommendation metadata) συνοδεύουν ένα αρχείο ήχου χωρίς να προσαρτώνται σε αυτό, προστίθενται από χρήστες ή παράγονται αυτόματα από αλγόριθμους και παρέχουν πληροφορία σχετικά με χαρακτηριστικά του μουσικού έργου όπως το είδος στο οποίο ανήκει, τί διάθεση δημιουργεί στο άκουσμά του, αν είναι καθαρά οργανικό η αν εμπεριέχονται φωνητικά και άλλα.

Εκτός από τα μεταδεδομένα, η περιγραφή του μουσικού περιεχομένου ενός έργου, μπορεί να αναφέρεται σε διάφορα μουσικά στοιχεία όπως είναι η τονικότητα, το τέμπο, ο ρυθμός,

η μελωδία, η αρμονία, η χροιά ή χρώμα, οι δυναμικές, η μορφή, η δομή και άλλα. Σε αντίθεση με τα μεταδεδομένα, η πληροφορία αυτή μεταβάλλεται κατά τη διάρκεια του μουσικού έργου.

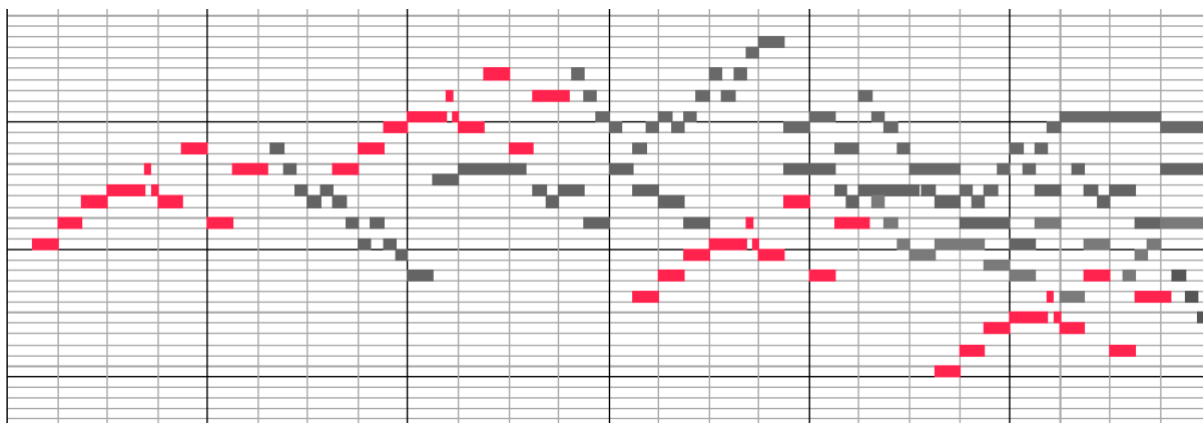
Υπάρχουν διάφορες μορφές αποτύπωσης της πληροφορίας αυτού του τύπου, η παλαιότερη εκ των οποίων είναι η παρτιτούρα. Η μορφή αυτή έχει σαν βάση τις νότες, για καθεμία από τις οποίες παρέχονται οι ιδιότητες του τονικού ύψους, της χροιάς, της ακουστότητας και της διάρκειας. Εκτός των στοιχείων που αφορούν κάθε νότα, η παρτιτούρα εμπεριέχει και πιο γενικές πληροφορίες που αφορούν τη ρυθμική αγωγή, τους χρονοδείκτες, το μουσικό κλειδί κλπ.

Πέρα από την παρτιτούρα, υπάρχουν και εναλλακτικές συμβολικές μορφές με τις οποίες αποτυπώνεται το μουσικό περιεχόμενο ενός έργου. Αυτές εμπεριέχουν οποιαδήποτε αναπαράσταση παρτιτούρας που πραγματοποιείται μέσω ρητής κωδικοποίησης νοτών ή άλλων μουσικών γεγονότων σε σύμβολα. Η πρώτη και αναλογική τέτοια συμβολική αναπαράσταση εντοπίζεται περίπου στο τέλος του 19^{ου} αιώνα, όταν και γίνονται δημοφιλή τα πιάνο χωρίς μουσικό που παίζουν μουσική βάσει της πληροφορίας που αναγιγνώσκουν από ένα ρολό χαρτιού με τρύπες (Müller, 2015). Πρόκειται για το γνωστό piano roll, η σύγχρονη ψηφιακή εκδοχή του οποίου χρησιμοποιείται συχνά σε λογισμικό επεξεργασίας και αναπαραγωγής MIDI αρχείων και αρχείων ήχου. Στο αναλογικό λοιπόν, το ρολό ξετυλίγεται και διέρχεται από έναν μηχανισμό ανάγνωσης, με τις τρύπες στο χαρτί να αναπαριστούν διακριτά μουσικά γεγονότα, δηλαδή μεμονωμένες νότες ή συγχορδίες. Στο σύστημα αυτό, το ύψος και τη διάρκεια της κάθε νότας προσδιορίζονται ανάλογα με τη θέση της στον κάθετο άξονα του ρολού και το μέγεθος της τρύπας αντίστοιχα. Ο οριζόντιος άξονας του ρολού αναπαριστά το χρόνο.

Παρακάτω, στην Εικόνα 1-1 φαίνεται ένα αναλογικό piano roll ενώ στην Εικόνα 1-2 μια ψηφιακή του αναπαράσταση:



Εικόνα 1-1: Αναλογικό piano roll που παίζει



Εικόνα 1-2: Ψηφιακή αναπαράσταση για piano roll. Στον οριζόντιο άξονα αναπαρίσταται ο χρόνος, στον κάθετο το τονικό ύψος. Για κάθε ορθογώνιο-νότα δίνεται η πληροφορία του χρόνου έναρξης, της διάρκειας και του ύψους. Στο περιβάλλον του λογισμικού οι κόκκινες νότες

Στη σύγχρονη εποχή, η παρτιτούρα αποδίδεται ψηφιακά, με το MIDI πρωτόκολλο ή διάφορες κωδικοποιήσεις σε μορφή κειμένου ASCII, όπως είναι οι κωδικοποιήσεις kern, MusicXML, MEI, ABC, κάποιες εκ των οποίων αποτελούν κοινά συστήματα αναπαραστάσεων.

Αυτές οι ψηφιακές αναπαραστάσεις της μουσικής πληροφορίας είναι αντικείμενα επεξεργασίας από κατάλληλο λογισμικό. Για παράδειγμα, υπάρχουν διάφορα εργαλεία και εφαρμογές γραφικής απόδοσης (rendering engines) που μετατρέπουν την κωδικοποιημένη σε αυτές τις μορφές μουσική πληροφορία είτε πίσω σε γραφική μορφή (παρτιτούρα) είτε σε ήχο, συνήθως μέσω του πρωτοκόλλου MIDI. Επιπλέον, καθιερωμένη είναι η χρήση προγραμματιστικών βιβλιοθηκών προς σύνθεση, χειρισμό, επεξεργασία, ανάλυση αρχείων που εμπεριέχουν τέτοιου είδους κωδικοποιήσεις. Οι κωδικοποιήσεις αυτές χρησιμοποιούνται μεταξύ άλλων για μουσικολογική ανάλυση. Ερωτήματα όπως το πόσο συχνά ανευρίσκονται συγκεκριμένα μουσικά διαστήματα σε συνθέσεις της περιόδου του ρομαντισμού ή αν οι διαφωνίες ανευρίσκονται περισσότερο σε ισχυρά παρά σε ασθενή μετρικά σημεία στη μουσική ενός συνθέτη μπορούν να απαντώνται έπειτα από επεξεργασία και ανάλυση τέτοιων συμβολικών αναπαραστάσεων. Πραγματοποιείται στη συνέχεια της εργασίας μια πιο εκτενής ανάλυση των πιο κοινών ψηφιακών μορφών αναπαράστασης μουσικής πληροφορίας καθώς και του σχετικού λογισμικού.

1.4 Δομή Εργασίας

Η εργασία έχει την ακόλουθη δομή.

Το Κεφάλαιο 2 επικεντρώνεται στην περιγραφή κάποιων από τα πιο δημοφιλή πρότυπα περιγραφής μουσικού περιεχομένου. Ενδεικτικά αναφέρονται τα πρότυπα MIDI, Kern, ABC, MusicXML και MEI και παρουσιάζεται μια συγκριτική ανάλυση των προδιαγραφών και των δυνατοτήτων τους

Το Κεφάλαιο 3 περιγράφει εργαλεία λογισμικού που εστιάζουν στη γραφική απόδοση παρτιτούρας (music engraving), καθώς και κάποιες προγραμματιστικές βιβλιοθήκες που χρησιμοποιούνται για τη σύνθεση, το χειρισμό και την ανάλυση αρχείων μουσικής σημειογραφίας.

Στο Κεφάλαιο 4 παρατίθενται οι παρεμβάσεις που πραγματοποιήθηκαν σε ένα διαδικτυακό εργαλείο σύνθεσης, επεξεργασίας και απόδοσης παρτιτούρας. Οι παρεμβάσεις αυτές έγιναν με στόχο την υποστήριξη λειτουργιών για τη μουσική εκπαίδευση από απόσταση.

Τέλος, το Κεφάλαιο 5 συνοψίζει τα αποτελέσματα της εργασίας και καταγράφει πιθανές μελλοντικές επεκτάσεις που θα μπορούσαν βελτιώσουν περαιτέρω την εμπειρία χρήσης τέτοιων εφαρμογών κατά τη μουσική εκπαίδευση.

2. ΠΡΟΤΥΠΑ ΠΕΡΙΓΡΑΦΗΣ ΜΟΥΣΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ

2.1 Γενικά

Καθώς η ψηφιακή κωδικοποίηση της μουσικής πληροφορίας εξυπηρετεί πολλούς σκοπούς, έχουν αναπτυχθεί δεκάδες μορφότυποι και τρόποι αναπαράστασης.

Η ανάγκη για πρότυπα κωδικοποίησης μουσικής πληροφορίας που θα διαδραμάτιζαν υποβοηθητικό ρόλο στον τότε τομέα της Υπολογιστικής Μουσικολογίας, αναδύθηκε τη δεκαετία του '60. Δύο από τις πρώτες και πιο ευρέως χρησιμοποιούμενες γλώσσες κωδικοποίησης ήταν η Ford-Columbia Music Representation, γνωστή και ως DARMs, και ο κώδικας των Plaine και Easie, βάσει του οποίου κωδικοποιήθηκε μουσικό υλικό της συλλογής της Repertoire International des Sources Musicales (RISM). Μια πολύ σημαντική εξέλιξη στον τομέα ήταν η καθιέρωση του πρωτοκόλλου MIDI (Musical Instrument Digital Interface) τη δεκαετία του '80. Αν και αρχικά σχεδιάστηκε για να προτυποποιήσει και να διευκολύνει την επικοινωνία μεταξύ ψηφιακών ηλεκτρονικών μουσικών συσκευών, χρησιμοποιήθηκε ευρέως από μουσικούς ερευνητές. Την ίδια δεκαετία αναδύθηκε και η αναπαράσταση kern ως μέρος του συντακτικού συστήματος Humdrum. Στο kern σύστημα η πληροφορία αναπαρίσταται σε κώδικα γραμμένο στη μορφή κειμένου. Με τον ίδιο τρόπο κωδικοποιείται η μουσική πληροφορία και στη Musedata, η οποία δημιουργήθηκε τη δεκαετία του '90. Η, όχι και τόσο δημοφιλής, NIFF (Notation Interchange File Format) κωδικοποίηση χρησιμοποιήθηκε για την ανταλλαγή μουσικής πληροφορίας ανάμεσα σε προγράμματα δημιουργίας, επεξεργασίας και εκτύπωσης μουσικής παρτιτούρας. Τέλος, δύο άλλες δημοφιλείς σύγχρονες κωδικοποιήσεις που βασίζονται στην γλώσσα σήμανσης XML (eXtensible Markup Language) είναι το MusicXML που παρουσιάστηκε σε ένα συνέδριο της ISMIR (International Society for Music Information Retrieval) στις αρχές του 2000, και το MEI (Music Encoding Initiative) που αποτελεί ένα βασικό σετ κανόνων περιγραφής της μουσικής πληροφορίας το οποίο μπορεί να επεκτείνεται και να προσαρμόζεται στην εκάστοτε περίπτωση χρήσης του. (Devaney, 2020)

Όπως καταγράφεται στα προηγούμενα, οι γλώσσες κωδικοποίησης της μουσικής πληροφορίας παρτιτούρας διακρίνονται ανάλογα με το αν την αναπαριστούν σε δυαδικό κώδικα ή κώδικα σε μορφή κειμένου ASCII. Από τις κωδικοποιήσεις που περιγράφονται πιο κάτω το πρωτόκολλο MIDI ανήκει στην πρώτη κατηγορία, ενώ στη δεύτερη ανήκουν το MusicXML, το MEI, το **kern και το ABC.

2.2 MIDI

Το MIDI (Musical Instrument Digital Interface, δηλαδή Ψηφιακή Διασύνδεση Μουσικών Οργάνων) θεμελιώθηκε ως πρότυπο τον Αύγουστο του 1982 από την κοινοπραξία MMA (Midi Manufacturers' Association) με την ανακοίνωση του κειμένου προδιαγραφών "MIDI Specification 1.0", το οποίο περιέγραφε τις βασικές οδηγίες που μπορούσαν να σταλούν ανάμεσα σε δύο ηλεκτρονικές μουσικές μονάδες. Το MIDI αποτελεί ένα πρωτόκολλο ελέγχου, δηλαδή η πληροφορία μεταφέρεται σε μορφή εντολών και δεν έχει σαν στόχο να αναπαραχθεί αυτούσια στον παραλήπτη, αλλά να ενεργοποιήσει κάποια λειτουργία του. Ο

μουσικός, ενεργώντας στη συσκευή ελέγχου, έχει τη δυνατότητα να ελέγχει το ηλεκτρονικό μουσικό όργανο σε πραγματικό χρόνο με απλές πράξεις όπως το πάτημα ενός πλήκτρου κλαβιέ. Επήλθε, λοιπόν, με την έλευση του MIDI μια ταχεία ανάπτυξη της αγοράς τέτοιων συσκευών.

Πιο κάτω, στην Εικόνα 2-1, βλέπουμε ένα κλασσικό synthesizer της εποχής:



Εικόνα 2-1: Roland Juno-106

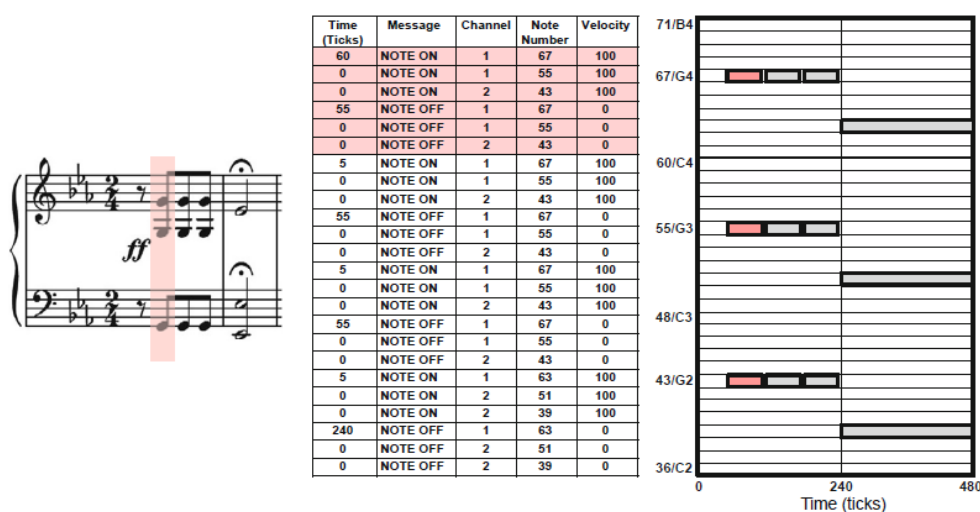
Το σετ προδιαγραφών MIDI Specification 1.0 αποτελούνταν από δύο τμήματα. Το πρώτο (MIDI Cables & Connectors) περιέγραφε τη φυσική διασύνδεση (δηλ. καλώδια και βύσματα) που απαιτούνταν από τις, συμβατές με το MIDI, συσκευές για τη μεταξύ τους επικοινωνία, ενώ το δεύτερο (MIDI Messages) όριζε την κωδικοποίηση των μηνυμάτων, δηλαδή τη σημασιολογία και τη σύνταξη των εντολών η οποία όπως αναφέρθηκε και πιο πάνω υφίσταται σε δυαδική μορφή. Οι αρχικές προδιαγραφές αργότερα επαυξήθηκαν για να συμπεριλάβουν την προτυποποίηση SMF (Standard Midi File), που περιγράφει τον τρόπο αποθήκευσης των MIDI δεδομένων σε ψηφιακό μέσο (πχ στο σκληρό δίσκο ενός υπολογιστή). Το, ανεξάρτητο λειτουργικού συστήματος, πρότυπο SMF επιτρέπει σε μια ευρεία γκάμα χρηστών να ανταλλάζουν MIDI δεδομένα, χτίζοντας τη βάση για ένα αποτελεσματικό σύστημα διαδικτυακής διανομής της μουσικής. Ένα SMF αρχείο αποτελείται από μια λίστα μηνυμάτων MIDI, καταμεμημένων στη διάσταση του χρόνου μέσω μιας “χρονικής σφραγίδας” (timestamp). Εκτός των μηνυμάτων MIDI, σε ένα αρχείο εμπεριέχεται και περαιτέρω πληροφορία που αφορά το λογισμικό που χρησιμοποιήθηκε για την επεξεργασία του (meta messages).

Από άποψη περιγραφής μουσικής πληροφορίας παρτιτούρας, τα πιο σημαντικά MIDI μηνύματα είναι τα note-on/note-off, τα οποία σηματοδοτούν την έναρξη και την παύση, αντίστοιχα, της αναπαραγωγής ενός ηχητικού γεγονότος με συγκεκριμένο ύψος, χροιά και ένταση. Το ουσιαστικό κομμάτι καθενός από αυτά τα μηνύματα αποτελείται από τρεις ακέραιους αριθμούς κωδικοποιημένους δυαδικά. Ο πρώτος στη σειρά μπορεί να λάβει δεκαέξι διαφορετικές τιμές και καθορίζει το κανάλι στο οποίο απευθύνεται το μήνυμα MIDI. Τα κανάλια της ελεγχόμενης συσκευής, της γεννήτριας ήχου (sound module), συνήθως καθορίζουν τη χροιά αναπαραγωγής των ηχητικών γεγονότων. Η προσθήκη “General MIDI

Device Specifications” στις προδιαγραφές του MIDI, μεταξύ άλλων, προτυποποιεί και αριθμοδοτεί κάποιες χροίες που μπορούν να επιλεγούν. Στη συνέχεια, ακολουθεί ο αριθμός της MIDI νότας που πρόκειται να παιχθεί από την ελεγχόμενη συσκευή, ο οποίος βρίσκεται στο κλειστό εύρος 0-127 και αντιστοιχίζεται στο ύψος της νότα. Αυτό το εύρος αριθμών κωδικοποιεί περίπου 9,5 οκτάβες δυτικής μουσικής. Το μήνυμα ολοκληρώνεται με τον τρίτο ακέραιο (εύρος: 0-127) που στα μηνύματα note-on ορίζει την ένταση αναπαραγωγής της νότας και στα μηνύματα note-off ελέγχει την χρονική περίοδο decay της περιβάλλουσας.

Η πληροφορία της χρονικής στιγμής που συνοδεύει ένα ηχητικό γεγονός σε ένα SMF αρχείο είναι ένας ακέραιος που αναπαριστά πόσοι παλμοί του ρολογιού (ticks) της ελεγχόμενης συσκευής πρέπει να περάσουν μέχρι την αναπαραγωγή του. Όπως και στην περίπτωση μιας παρτιτούρας, το MIDI πρότυπο έχει τη δυνατότητα να εκφράζει τη χρονική πληροφορία με ένα σχετικό τρόπο. Δηλαδή, ένας παλμός ρολογιού αποτελεί μια υποδιαίρεση της μουσικής αξίας του τετάρτου. Ο αριθμός των παλμών ανά τέταρτο (PPQN: pulses per quarter note) καθορίζεται στην αρχή (header) του MIDI αρχείου και αφορά όλα τα MIDI μηνύματα που ακολουθούν. Μια κοινή τιμή αυτού του αριθμού είναι 120 PPQN. Εκτός του σχετικού τρόπου, η χρονική πληροφορία μπορεί να εκφραστεί και με απόλυτο τρόπο, μέσω της ενσωμάτωσης μηνυμάτων “tempo” στο αρχείο MIDI, τα οποία καθορίζουν τον αριθμό των milliseconds ανά αξία τετάρτου. Από ένα μήνυμα tempo η ελεγχόμενη συσκευή εξάγει την απόλυτη χρονική διάρκεια ενός παλμού (Müller, 2015). Για παράδειγμα αν το tempo μήνυμα ορίζει ότι κάθε τέταρτο διαρκεί 600000 ms και 120 PPQN τότε ο παλμός αντιστοιχεί σε 5000 ms. Η πληροφορία ενός μηνύματος tempo μεταφράζεται εύκολα σε BPM (Beats Per Minute). Στο πιο πάνω παράδειγμα το τέμπο είναι 120 BPM. Σε αντίθεση με τον αριθμό των παλμών ανά τέταρτο που ορίζεται στην αρχή και αφορά όλο το αρχείο MIDI, τα μηνύματα tempo μπορεί να είναι πολλά και διάσπαρτα μέσα στο αρχείο. Έτσι, μπορούν να αναπαρασταθούν οι τοπικές μεταβολές του τέμπο μιας παρτιτούρας όπως επίσης και να κωδικοποιηθούν στο αρχείο οι αυξομειώσεις της ταχύτητας με την οποία παίζεται ένα μουσικό κομμάτι (accelerando, ritardando κλπ).

Στην Εικόνα 2-2 απεικονίζεται ένα παράδειγμα μετάφρασης μουσικής πληροφορίας παρτιτούρας σε MIDI μηνύματα και αναπαράστασης της σε piano-roll:



Εικόνα 2-2: Στη μέση παράδειγμα κωδικοποίησης MIDI, αριστερά η απόδοσή του σε παρτιτούρα, δεξιά η αναπαράσταση του σε piano-roll

2.3 ABC Notation Format

Το σύστημα μουσικής σημειογραφίας ABC, στη βασική του μορφή, χρησιμοποιεί λατινικούς χαρακτήρες για να κωδικοποιήσει μουσικές νότες (a-g, A-G) και παύσεις (z). Στους χαρακτήρες αυτούς προστίθενται σύμβολα που αποδίδουν τη διάρκεια της νότας, την μουσική κλίμακα, αλλοιώσεις και καλλωπισμούς. Αυτή η μορφή σημειογραφίας βασίζεται στο σύστημα σημειογραφίας ύψους του Helmholtz (χρήση λατινικών χαρακτήρων για αναπαράσταση των μουσικών υψών), και αποτελεί, έπειτα και από κάποιες προσθήκες ASCII χαρακτήρων που αποδίδουν περαιτέρω στοιχεία (διαστολές, τέμπο κλπ), ένα απλό και χρηστικό σύστημα αναπαράστασης.

Αρχικά, αυτό το σύστημα κωδικοποίησης μουσικής σημειογραφίας σχεδιάστηκε προς αναπαράσταση παραδοσιακής Δυτικοευρωπαϊκής μουσικής, που αποτελείται κυρίως από μονοφωνικές μελωδίες. Η κωδικοποίηση ABC επεκτάθηκε και προτυποποιήθηκε με τη συμβολή του Chris Walshaw, που πρόσθεσε μια σειρά κωδικοποιητικών χαρακτήρων και πρόσφερε ένα συντακτικό σύστημα που μπορεί να υποστηρίξει την κωδικοποίηση μεταδεδομένων. Αργότερα, εμπορικά πακέτα λογισμικού επέκτειναν περαιτέρω το συντακτικό σύστημα, ώστε προστέθηκε και η δυνατότητα κωδικοποίησης στίχων.

Εφόσον το σύστημα μουσικής σημειογραφίας ABC είναι βασισμένο σε χαρακτήρες ASCII, οποιοσδήποτε επεξεργαστής κειμένου μπορεί να χρησιμοποιηθεί για τη δημιουργία και την επεξεργασία τέτοιου κώδικα. Παρόλα αυτά, υπάρχουν αρκετά πακέτα λογισμικού που εξυπηρετούν αυτούς τους σκοπούς και παρέχουν πρόσθετες δυνατότητες, όπως η μετατροπή του ABC κώδικα σε αρχείο MIDI³.

Στην Εικόνα 2-3 φαίνεται παράδειγμα ABC κώδικα και στην Εικόνα 2-4 η παρτιτούρα που αναπαριστά:

```
<score lang="ABC">
X:1
T:The Legacy Jig
M:6/8
L:1/8
R:jig
K:G
GFG BAB | gfg gab | GFG BAB | d2A AFD |
GFG BAB | gfg gab | age edB |1 dBA AFD :|2 dBA ABd |:
efe edB | dBA ABd | efe edB | gdB ABd |
efe edB | d2d def | gfe edB |1 dBA ABd :|2 dBA AFD |]
</score>
```

Εικόνα 2-3: Παράδειγμα ABC κώδικα

³ <https://abcnotation.com/>

The Legacy Jig



Εικόνα 2-4: Η αναπαράσταση σε παρτιτούρα του κώδικα της Εικόνας 2-3

2.4 Kern και Humdrum

Το Humdrum αρχικά δημιουργήθηκε από τον David Huron τη δεκαετία του 1980 και χρησιμοποιείται σταθερά έως και σήμερα. Το σύστημα αυτό αποτελείται από δύο τμήματα: ένα τμήμα που ορίζει διάφορες αναπαραστάσεις (εκ των οποίων η πιο δημοφιλής είναι η kern) και ένα άλλο που αποτελείται από λογισμικό ανάλυσης της πληροφορίας που εμπεριέχεται στις αναπαραστάσεις, υλοποιημένο σε μορφή εντολών για προγράμματα shell (διερμηνείς γραμμής εντολών). Η kern σύνταξη σχεδιάστηκε για να διευκολύνει, μάλλον την ανάλυση της μουσικής πληροφορίας, παρά την απόδοσή της σε παρτιτούρα ή ήχο⁴.

Στο σύστημα Humdrum δίνεται η δυνατότητα στους χρήστες να δημιουργούν εύκολα νέες μουσικές αναπαραστάσεις για ειδικούς σκοπούς. Έτσι, βάσει προσαρμοσμένων, αλλά βασισμένων στο Humdrum, συντακτικών, έχει κωδικοποιηθεί μια ευρεία γκάμα μουσικής πληροφορίας όπως μουσική ξυλόφωνου από την Ουγκάντα, ταμπλατούρες για ακορντεόν τύπου Cajun ή για περσικό νέυ και τετράγωνη σημειογραφία (πρόγονος του σημερινού συστήματος σημειογραφίας Δυτικής Μουσικής).

Όλες οι μορφοποιήσεις αρχείων Humdrum είναι γραμμένες σε ASCII κείμενο. Το γεγονός αυτό διευκολύνει την συμβατότητα και φορητότητα όσον αφορά την πλατφόρμα, καθώς και τη συγγραφή συντακτικών ειδικού σκοπού από χρήστες. Επίσης, καθιστά δυνατή την απευθείας ανθρώπινη ανάγνωση της μουσικής πληροφορίας.

Τα Humdrum δεδομένα οργανώνονται σε πίνακες δύο διαστάσεων, όπως ένα λογιστικό φύλλο. Οι στήλες ορίζουν τον τύπο της κωδικοποιημένης πληροφορίας, ενώ οι γραμμές αναπαριστούν διαδοχικές χρονικές στιγμές. Όσο, λοιπόν, πιο κάτω στη σελίδα βρίσκονται τα δεδομένα, τόσο περισσότερος χρόνος έχει περάσει από την έναρξη του μουσικού έργου.

Ένα παράδειγμα φαίνεται παρακάτω στην Εικόνα 2-5, όπου στην αριστερή στήλη αναπαρίστανται τα μουσικά ύψη κωδικοποιημένα βάσει του συστήματος ABC, και στη δεξιά στήλη προτείνονται τα δάκτυλα που ενδείκνυται να χρησιμοποιηθούν για το παίξιμο των νωτών από μουσικό:

⁴ <http://www.humdrum.org>

**pitch	**fingering
C4	R1
D4	R2
E4	R3
F4	R1
G4	R2
A4	R3
B4	R4
C5	R5
* _	* _

Εικόνα 2-5: Παράδειγμα κωδικοποίησης Humdrum

Σε αντίθεση με ένα λογιστικό φύλλο, οι στήλες σε ένα αρχείο Humdrum (spines) παρουσιάζουν μεγαλύτερη ευελιξία. Μπορούν να ενωθούν ή να χωριστούν, να αλλάξουν θέσεις ή να διακοπούν, καθώς και να εισαχθούν σε οποιοδήποτε σημείο του αρχείου. Ο βασικός τύπος της πληροφορίας που αναπαρίσταται σε μία στήλη Humdrum καθορίζεται με μία ετικέτα (διπλός αστερίσκος), ενώ συμπληρωματικές (μονός αστερίσκος) ετικέτες χρησιμοποιούνται για περαιτέρω αποσαφήνιση του τύπου. Επιπρόσθετα, τα σχόλια σε ένα αρχείο Humdrum, τα οποία ξεκινούν με ένα θαυμαστικό, μπορεί να αφορούν όλο το αρχείο, μια στήλη ή γραμμή, ένα «κελί» δεδομένων ή ακόμα και μέρος του κελιού.

Συνήθως, τα αρχεία Humdrum κωδικοποιούν την παρτιτούρα ενός μουσικού έργου. Εντός ενός τέτοιου αρχείου, τα κελιά δεδομένων εμπεριέχουν διακριτά μουσικά γεγονότα (πχ νότες) και η καθεμία στήλη αφορά ένα μουσικό όργανο. Εκτός του σχήματος που αναπαριστά μουσικά ύψη, στο Humdrum υπάρχουν περίπου 30 προ-ορισμένα αναπαραστατικά σχήματα που κωδικοποιούν πληροφορία σχετική με αρμονικά/μελωδικά διαστήματα, διάρκεια, δυναμική, βαθμίδες κλίμακας, ακόμα και στίχους (Huron, 2015).

Από τα προ-ορισμένα σχήματα, το πιο δημοφιλές είναι το ****kern** (η γερμανική λέξη για τον πυρήνα), που χρησιμοποιείται για την αναπαράσταση βασικής (πυρηνικής) μουσικής πληροφορίας παρτιτούρας σχετικά με το μουσικό ύψος, τις αλλοιώσεις και τη διάρκεια μιας νότας, τις παύσεις, τις γραμμές έναρξης και λήξης ενός μουσικού μέτρου, ζεύξεις και συζεύξεις διάρκειας, σημεία δυναμικής και προσωδίας, καλλωπισμούς κλπ.

Στην Εικόνα 2-6 φαίνεται τμήμα μιας παρτιτούρας και στην Εικόνα 2-7 η κωδικοποίησή της σε αναπαράσταση kern:

2. Rondo - Dance

Dmitri Kabalevsky



Εικόνα 2-6: Τμήμα ενός μουσικού έργου

```
!!!COM: Kabalevsky, Dmitri
!!!OTL: 2. Rondo-Dance
**kern          **kern
*staff2        *staff1
*clefF4        *clefG2
*k[f#c#]       *k[f#c#]
*M3/8          *M3/8
8r             (16dd
.              16cc#
=              =
8D'           8dd')
8A'           8f#' 8a'
8r            (16b
.             16a#
=             =
8BB'         8b')
8F#'         8d' 8f#'
8r           (16g
.            16f#
=            =
8BB-        8g')
8G'         8c#' 8e'
8F#'        8a'
=           =
8D          (8f#
8A          8d)
*-          *-
```

Εικόνα 2-7: Η kern κωδικοποίηση της παρτιτούρας της Εικόνας 2-6

Αναπαρίσταται, λοιπόν, η κωδικοποιημένη παρτιτούρα σε δισδιάσταση μορφή σαν να έχει στραφεί κατά 90 μοίρες. Οι αρχικές ετικέτες αριθμοδοτούν τα υπάρχοντα πεντάγραμμα και εμπεριέχουν πληροφορία σχετικά με τα μουσικά κλειδιά, τις μουσικές κλίμακες και τους χρονοδείκτες. Ένα μουσικό γεγονός ξεκινά με έναν αριθμό που κωδικοποιεί τη διάρκειά του (πχ 8 για όγδοο), συνεχίζει με ένα γράμμα που αναπαριστά το μουσικό ύψος (βάσει του ABC συστήματος), και τελειώνει πιθανώς με σύμβολα που δηλώνουν είτε ζεύξεις προσωδίας/δυναμικής (παρενθέσεις) είτε παρεσιγμένη χρονική διάρκεια (τόνοι). Για να αναπαρασταθούν δύο νότες που παίζονται την ίδια χρονική στιγμή, χρησιμοποιείται ο κενός χαρακτήρας ενδιάμεσά τους, ενώ τα μουσικά μέτρα χωρίζονται με το σύμβολο “=”.

2.5 MusicXML

Το MusicXML δημιουργήθηκε, όπως αναφέρθηκε πιο πάνω, στις αρχές του 21^{ου} αιώνα με σκοπό να χρησιμοποιηθεί ως ενδιάμεση μορφοποίηση για ανταλλαγή μουσικής πληροφορίας μεταξύ εφαρμογών μουσικής σημειογραφίας, μουσικής ανάλυσης, ανάκτησης μουσικής πληροφορίας και μουσικής εκτέλεσης. Στοχεύοντας στην επάρκεια και όχι στη βελτιστότητα, επαύξησε χωρίς να αντικαταστήσει τις προϋπάρχουσες εξειδικευμένες μορφοποιήσεις των εφαρμογών. Το MusicXML βασίστηκε και επέκτεινε δύο προϋπάρχουσες μορφοποιήσεις, το MuseData και το Humdrum. Το σχήμα ορισμού του είναι διαθέσιμο ελεύθερα στο διαδίκτυο.

Το MusicXML είναι ένα σχήμα XML. Το XML προτυποποιήθηκε το 1998 ως μια γλώσσα σήμανσης και είχε ως στόχο τη διευκόλυνση της διαδικτυακής κυκλοφορίας εγγράφων. Ορίζει ένα συντακτικό σύστημα που χρησιμοποιείται για να «σημάνει» ένα έγγραφο, να καθορίσει δηλαδή πώς αυτό εμφανίζεται εκτυπωμένο ή σε μια οθόνη. Το σύστημα αποτελείται από στοιχεία (elements), δηλαδή κείμενο εγκλεισμένο σε “ετικέτες”. Ένα αρχείο XML δομείται βάσει συγκεκριμένης ιεραρχίας, κατά την οποία στοιχεία εμπεριέχουν άλλα στοιχεία. Για τη συγγραφή κώδικα σε μια XML γλώσσα, πέρα από το συντακτικό σύστημα απαιτείται και ένα σετ νοηματικών κανόνων, το σχήμα, που καθορίζει τα στοιχεία που χρησιμοποιούνται καθώς και τις επιτρεπτές δομικές τους σχέσεις.

Γενικά, για την ψηφιακή κωδικοποίηση μιας παρτιτούρας λαμβάνονται υπόψη είτε στοιχεία που μπορεί να αφορούν την αναπαράσταση καθαρά μουσική πληροφορίας (μελωδίες, ρυθμοί κλπ), είτε στοιχεία που αφορούν την απεικόνιση στο πεντάγραμμο όπως η κατεύθυνση του στελέχους ενός φθογγόσημου και η γραμματοσειρά γραφής, είτε στοιχεία που αφορούν τη μουσική εκτέλεση (τέμπο, δυναμικές κλπ). Σε ένα αρχείο MusicXML, οι πληροφορίες που ανήκουν στην πρώτη κατηγορία υποδηλώνονται με διαφορετικά στοιχεία από εκείνες της δεύτερης και τρίτης κατηγορίας και έτσι επιτυγχάνεται μια διακριτοποίηση των, κατά τα άλλα ενοποιημένων σε ένα κοινό κώδικα, κατηγοριών.



Στο MusicXML υπάρχουν δύο βασικές δομές για την αναπαράσταση μιας μουσικής παρτιτούρας. Στην πρώτη η βασική μονάδα οργάνωσης του κώδικα είναι το μουσικό μέτρο και αυτό εμπεριέχει όλα τα ξεχωριστά μέρη (“πάρτες”) μιας μουσικής σύνθεσης. Αντίστροφα, στη δεύτερη δομή, η μονάδα οργάνωσης είναι η πάρτη, στην οποία εμπεριέχονται τα μουσικά της μέτρα. Πέρα από τη διαφορά στη βασική μονάδα οργάνωσης, οι δύο μορφές χρησιμοποιούν κοινά XML στοιχεία. Η μετατροπή ανάμεσα στις δύο μορφές πραγματοποιείται αυτόματα (Khan et al, 2014).

Σε μια παρτιτούρα που δομείται βάσει του δεύτερου τρόπου οργάνωσης, το αρχικό στοιχείο είναι το <score-partwise>, το οποίο εμπεριέχει όλα τα υπόλοιπα στοιχεία, μεταδεδομένα διαφόρων τύπων (πχ τίτλος, λογισμικό που χρησιμοποιήθηκε για τη δημιουργία της κλπ.), καθώς και πληροφορίες για την απεικόνισή της παρτιτούρας (μέγεθος πενταγράμμου, γραμματοσειρές κλπ.). Άλλο βασικό στοιχείο είναι το <partlist> που αποτελεί μια λίστα όλων των επακόλουθων <part> στοιχείων, το καθένα εκ των οποίων περιέχει κωδικοποιημένη τη μουσική πληροφορία μιας πάρτης. Εντός ενός <part> στοιχείου παρατίθενται το ένα μετά το άλλο τα μουσικά μέτρα κωδικοποιημένα σε στοιχεία <measure>. Το πιο κοινό στοιχείο που χρησιμοποιείται εντός ενός <measure> στοιχείου είναι το <note> που αναπαριστά ένα φθογγόσημο παρτιτούρας. Ένα <note> στοιχείο εγκλείει στοιχεία για την κωδικοποίηση του μουσικού ύψους ή της παύσης, της διάρκειας, των αλλοιώσεων και γραφικών

χαρακτηριστικών του φθογγόσημου (πχ καλλωπισμοί, ζεύξεις, συζεύξεις διαρκείας κλπ.). Το <attribute> στοιχείο του <measure> περιέχει πληροφορία συνήθως σχετικά με το χρονοδείκτη, την κλίμακα ή το κλειδί. Επίσης ορίζει τη χρονική διάρκεια ενός τετάρτου με σχετικό τρόπο (Good, 2001).

Ένα παράδειγμα MusicXML κώδικα απεικονίζεται πιο κάτω στην Εικόνα 2-8:

```
<note>
  <pitch>
    <step>E</step>
    <alter>-1</alter>
    <octave>4</octave>
  </pitch>
  <duration>2</duration>
  <type>half</type>
</note>
```



Εικόνα 2-8: Αριστερά το παράδειγμα κώδικα και δεξιά η αποκωδικοποίηση σε παρτιτούρα

Τα αρχεία MusicXML είναι συνήθως μεγάλου μεγέθους. Αρκετά μεγαλύτερου από αρχεία εφαρμογών μουσικής σημειογραφίας, από τη μεταγραφή των οποίων πιθανώς προέκυψαν. Το γεγονός αυτό αρχικά ανέστειλε τη χρήση της MusicXML κωδικοποίησης για διαδικτυακή διανομή. Μια αναθεωρημένη έκδοση του προτύπου, το MusicXML 2.0, εισήγαγε ένα νέο συμπιεσμένο μορφότυπο με την επέκταση .mxl. Η συμπίεση που πραγματοποιείται μπορεί να μειώσει το μέγεθος του MusicXML αρχείου στα επίπεδα μιας, αντίστοιχου μουσικού περιεχομένου, MIDI κωδικοποίησης⁵.

2.6 MEI

Το Music Encoding Initiative (MEI) είναι ένα σετ κανόνων για την κωδικοποίηση της μουσικής πληροφορίας μιας παρτιτούρας, μια γλώσσα σήμανσης βασισμένη στο XML συντακτικό σύστημα. Το σχήμα ορισμού του MEI είναι, όπως και του MusicXML, διαθέσιμο στο διαδίκτυο και υπάρχει δυνατότητα επέκτασής του από τους χρήστες. Το MEI χρησιμοποιεί κείμενο ASCII, άρα τα αρχεία τέτοιου τύπου είναι ανθρωπίνως αναγνώσιμα.

Ένα έγγραφο MEI κώδικα μπορεί να κωδικοποιεί μουσική πληροφορία που προέρχεται από διάφορες πηγές. Για παράδειγμα, η πληροφορία μπορεί να προκύπτει από την ανθρώπινη ή τεχνητή (OMR: Optical Music Recognition) ερμηνεία μιας παρτιτούρας, ακόμα και από έναν αλγόριθμο μουσικής ανάλυσης.

Στο MEI υπάρχει ένα πυρηνικό σετ με προτυποποιημένες ενότητες κώδικα (modules), τις οποίες ο/η χρήστης-ρια μπορεί να επαυξήσει ή προσαρμόσει ανάλογα με τις ανάγκες του. Δύναται επίσης να δημιουργήσει εντελώς καινούργιες. Γενικά αυτές οι ενότητες ορίζουν διαφορετικού τύπου στοιχεία σήμανσης (elements). Μπορούν, ανάλογα με την εκάστοτε εργασία που πρόκειται να επιτελεστεί, να χρησιμοποιηθούν ανεξάρτητα η μία από την άλλη.

⁵ <https://www.w3.org/2021/06/musicxml40/tutorial/compressed-mxl-files/>

Μία από τις πυρηνικές ενότητες εμπεριέχει στοιχεία χρήσιμα για την αναπαράσταση του βασικού σκελετού μιας παρτιτούρας, δηλαδή των σελίδων, των πενταγράμμων ή των διαφόρων μερών μιας μουσικής σύνθεσης (πάρτες). Άλλη (Common Western notation module) παρέχει στοιχεία για κωδικοποίηση νοτών, μουσικών μέτρων, παύσεων, δυναμικών κλπ. Επιπρόσθετες ενότητες κωδικοποιούν καλλωπισμούς ή σύμβολα συγχορδίων καθώς και παρτιτούρες γραμμένες σε μετρική ή νευματική σημειογραφία.

Όπως και στο MusicXML, στο MEI μπορεί να κωδικοποιηθεί πληροφορία που αφορά είτε αυτό καθαυτό το μουσικό περιεχόμενο ή την εκτέλεσή του, είτε την οπτική του απεικόνιση, είτε επιπρόσθετα την περαιτέρω ανάλυση και επεξεργασία του από μουσικούς ερευνητές. Ο τελευταίος τομέας πληροφορίας εμπεριέχει λεπτομέρειες χαμηλού επιπέδου που ενδιαφέρουν συνήθως μουσικούς αναλυτές, όπως τις σχέσεις μεταξύ νοτών, συγχορδίων ή μέτρων, την αρμονικότητα μιας μελωδίας ή τη βαθμίδα ενός φθογγόσημου μέσα στην χρησιμοποιούμενη κλίμακα.

Όπως και στον MusicXML μορφότυπο, στα αρχεία MEI χρησιμοποιείται ένα προτυποποιημένο συντακτικό (XML) και ένα σχήμα ανοιχτού κώδικα, γεγονός που συμβάλλει στην ελεύθερη διαδικτυακή διανομή τους. Ευνοείται έτσι η δημιουργία αρχειοθετημένου μουσικού υλικού, στο οποίο κάποιος-α μπορεί να έχει εύκολη πρόσβαση και δυνατότητα επεξεργασίας. Προς υποστήριξη αυτής της χρήσης του MEI, το σχήμα παρέχει μια μεγάλη συλλογή στοιχείων που κωδικοποιούν μεταδεδομένα. Το γεγονός αυτό συμβάλλει επίσης στην καταλληλόλητα χρήσης του MEI από λογισμικό Ανάκτησης Μουσικής Πληροφορίας. Για παράδειγμα, τα μεταδεδομένα μπορεί να αφορούν πληροφορία για τα μουσικά έργα που αναπαρίστανται, τις πηγές από τις οποίες εξήχθη το μουσικό περιεχόμενο, τη διαδικασία με την οποία δημιουργήθηκε ένα MEI έγγραφο, ή τη σχέση του με άλλα έγγραφα (Crawford et al, 2016).

Παρακάτω στην Εικόνα 2-9 απεικονίζεται παράδειγμα MEI κώδικα:

Εικόνα 2-9: Παράδειγμα παρτιτούρας με τη MEI κωδικοποίησή της

2.7 Σύνοψη

Η ύπαρξη πληθώρας μορφότυπων αρχείων μπορεί να αποδοθεί στην πολυπλοκότητα της μουσικής σημειογραφίας προς κωδικοποίηση. Όπως αναφέρθηκε και πιο πάνω, μια παρτιτούρα εμπεριέχει μουσική πληροφορία σε πολλές διαστάσεις αφού καθορίζει μια σειρά παραμέτρων για την εκτέλεση του μουσικού έργου. Η πληροφορία που αναπαρίσταται

από ένα στοιχείο μιας παρτιτούρας όπως και η κατηγορία στην οποία ανήκει, καθορίζονται από πολλούς παράγοντες μεταξύ των οποίων η επιλογή της θέσης στον οριζόντιο/κάθετο άξονα της παρτιτούρας, των αποστάσεων του από διπλανά στοιχεία, καθώς επίσης και του χρώματος/μεγέθους/σχήματός του (Hankinson et al, 2011). Επιπρόσθετα, ανάλογα με το ιστορικό και πολιτισμικό πλαίσιο στο οποίο αναπτύχθηκε, κάθε μορφή μουσικής σημειογραφίας (μετρική, νευματική, Δυτικής Μουσικής κλπ.) απεικονίζει διαφορετικές κατηγορίες παραμέτρων μουσικής πληροφορίας, με κάθε ξεχωριστή παράμετρο να αναπαρίσταται με διαφορετικούς τρόπους. Δεν αποτελεί, λοιπόν, έκπληξη το γεγονός ότι οι διάφοροι μορφότυποι αρχείων μουσικής σημειογραφίας που διατίθενται δεν είναι ισοδύναμοι, δηλαδή δεν κωδικοποιούν τις ίδιες κατηγορίες μουσικής πληροφορίας. Κατ' επέκταση, ένα αρχείο συγκεκριμένου μορφότυπου, στις περισσότερες περιπτώσεις, δεν μπορεί να μεταγραφεί σε κάποιον άλλο χωρίς απώλεια πληροφορίας.

Ο μορφότυπος MIDI, για παράδειγμα, μπορεί να αναπαραστήσει λιγότερες κατηγορίες μουσικής πληροφορίας σχετικά με τους Humdrum/MusicXML/MEI, αφού το MIDI αρχικά σχεδιάστηκε ως πρωτόκολλο επικοινωνίας και η κωδικοποίηση προέκυψε στη συνέχεια για λόγους ηχητικής και όχι γραφικής (παρτιτούρα) αναπαράστασης. Για αυτό και συνηθέστερα, τα αρχεία MIDI αποδίδονται γραφικά σε μορφή `rianoroll` και όχι σε παρτιτούρα. Επίσης, οι δυνατότητες ενσωμάτωσης μεταδεδομένων ή σχολίων σε ένα τέτοιου τύπου αρχείο είναι φτωχές. Πρόκειται, όμως, για ένα δημοφιλές σύστημα αναπαράστασης μουσικής πληροφορίας, γεγονός που καταδεικνύεται και από πιο πρόσφατα διεθνή πρότυπα όπως είναι το Web MIDI API⁶, μια προγραμματιστική βιβλιοθήκη που αναπτύσσεται από το διεθνή οργανισμό W3C και στοχεύει στην χρήση του MIDI σε περιβάλλοντα διαδικτυακών περιηγητών. Τελικά, τα αρχεία MIDI είναι κατά κανόνα αρκετά μικρότερα από αρχεία άλλων μορφοτύπων, αφού εμπεριέχουν εντολές αναπαραγωγής σε δυαδική μορφή.

Από την άλλη, το σύστημα μουσικής σημειογραφίας ABC μπορεί να κωδικοποιήσει περισσότερες κατηγορίες μουσικής πληροφορίας, αν και όχι τόσες όσες οι υπόλοιποι μορφότυποι (για παράδειγμα δεν κωδικοποιεί καλλωπισμούς), αφού είναι σχετικά απλά δομημένο. Ωστόσο, λόγω αυτού του γεγονότος μια παρτιτούρα κωδικοποιημένη στο ABC σύστημα είναι ανθρωπίνως αρκετά ευανάγνωστη, τόσο ώστε με κάποια εξάσκηση να μπορεί να εκτελεστεί αυτούσια, χωρίς απεικόνιση στην κλασική της μορφή. Το ABC σύστημα, όντας απλό και εύκολα κατανοητό, χρησιμοποιήθηκε ως βάση σε πιο σύνθετες κωδικοποιήσεις (πχ Humdrum).

Ο μορφότυπος Humdrum δε σχεδιάστηκε με κύριο σκοπό τη γραφική απεικόνιση παρτιτούρας, ωστόσο χρησιμοποιείται σε σχετικές εργασίες. Είναι, όμως, εξ' ορισμού κατάλληλος για ανάλυση μουσικής πληροφορίας, αφού το σύστημα Humdrum παρέχει αρκετά σχετικά εργαλεία.

Οι κωδικοποιήσεις MusicXML και MEI μπορούν να αναπαραστήσουν πολλών διαφορετικών κατηγοριών πληροφορίες που αφορούν όχι μόνο το μουσικό περιεχόμενο ή την εκτέλεση του μουσικού έργου, αλλά και τη γραφική του απεικόνιση σε παρτιτούρα. Μάλιστα, όπως προαναφέρθηκε, οι διαφορετικές κατηγορίες αναπαρίστανται με διακριτοποιημένο τρόπο, γεγονός που καθιστά τις κωδικοποιήσεις αυτές κατάλληλες για χρήση σε εργασίες του ερευνητικού χώρου της Ανάκτησης Μουσικής Πληροφορίας. Ο μορφότυπος MEI δίνει

⁶ <https://www.w3.org/TR/webmidi/>

επιπλέον δυνατότητες επέκτασης (χρήση σε ειδικές περιπτώσεις μουσικής σημειογραφίας) καθώς και κωδικοποίησης ευρείας γκάμας μεταδεδομένων (χρήση για δημιουργία βάσεων δεδομένων). Τείνει να γίνει το πιο διαδεδομένο πρότυπο κωδικοποίησης. Συνήθως, όμως, αρχεία MEI ή MusicXML περιέχουν εκτενείς περιγραφές (verbose descriptions), κι επομένως έχουν μεγαλύτερο μέγεθος αρχείου, συγκριτικά με άλλους μορφότυπους, στην ασυμπίεστη μορφή τους.

Παρακάτω, στον Πίνακα 2-1, απεικονίζεται συνοπτικά αν η μορφή της κάθε αναλυθείσας κωδικοποίησης είναι δυαδική ή τύπου ASCII:

	Δυαδική	ASCII
MIDI	✓	
ABC		✓
Humdrum		✓
MusicXML		✓
MEI		✓

Πίνακας 2-1: Μορφή κωδικοποίησης των μορφότυπων

3. ΕΡΓΑΛΕΙΑ ΜΟΥΣΙΚΗΣ ΣΗΜΕΙΟΓΡΑΦΙΑΣ

Η απόδοση (rendering) μουσικής σημειογραφίας ασχολείται με την γραφική/ηχητική απόδοση μουσικής πληροφορίας που αναπαρίσταται σε ψηφιακούς μορφότευπους. Η γραφική απόδοση αρχείων μουσικής σημειογραφίας έγκειται στην (ανα)δημιουργία μιας παρτιτούρας βάσει της κωδικοποιημένης πληροφορίας, ενώ η ηχητική απόδοση συνήθως πραγματοποιείται μέσω εξαγωγής της σε μορφή μηνυμάτων MIDI. Τελικά βάσει των μηνυμάτων MIDI, είτε δημιουργείται απευθείας αρχείο μορφότευπου SMF, είτε εγγράφεται το αποτέλεσμα της εκτέλεσής τους από γεννήτρια ήχου σε αρχείο τύπου audio (MP3, FLAC κλπ.).

Στο κεφάλαιο αυτό παρουσιάζονται κάποιες προγραμματιστικές βιβλιοθήκες (API's: Application Programming Interfaces) καθώς και πακέτα λογισμικού/εφαρμογές, που χρησιμοποιούνται ευρέως στον τομέα της απόδοσης μουσικής σημειογραφίας. Τελικά, πραγματοποιείται μια σύγκριση αυτών ως προς συγκεκριμένες παραμέτρους.

3.1 Προγραμματιστικές βιβλιοθήκες (API's)

Σε ένα γενικό πλαίσιο, οι βιβλιοθήκες στοχεύουν στην υλοποίηση εξειδικευμένων εφαρμογών που συνθέτουν, επεξεργάζονται, αναλύουν και αποδίδουν με γραφικό/ηχητικό τρόπο τεχνουργήματα μουσικής σημειογραφίας. Σε αυτή την ενότητα παρατίθενται πέντε API's και αναλύονται οι χρήσεις τους.

3.1.1 Music21

Το Music21 είναι μια, ανοιχτού κώδικα και ανεξάρτητη πλατφόρμας λειτουργικού συστήματος, προγραμματιστική βιβλιοθήκη υλοποιημένη στην Python, μια δημοφιλή γλώσσα προγραμματισμού υψηλού επιπέδου. Δημιουργήθηκε το 2008 από την ακαδημαϊκή κοινότητα του Μουσικολογικού τμήματος του MIT και στη συνέχεια στην ανάπτυξη της συνέβαλαν και άλλα πανεπιστήμια των Η.Π.Α. Πιο πρόσφατα δημοσιεύτηκε η Music21j, μια έκδοση της Music21 σε γλώσσα Javascript. Πρόκειται στην ουσία για ένα αντικειμενοστραφές σετ εργαλείων προς ανάλυση και αναζήτηση μουσικής πληροφορίας καθώς και προς γραφική της απόδοση σε μορφή παρτιτούρας. Ο αρθρωτός (modular) χαρακτήρας του Music21 επιτρέπει σε μουσικούς και ερευνητές να δημιουργούν μικρά και επαναχρησιμοποιήσιμα προγράμματα με χρήση μόνο τμημάτων της βιβλιοθήκης.

Η μουσική πληροφορία μπορεί να εισαχθεί κωδικοποιημένη στις διάφορες δημοφιλείς μορφοποιήσεις -MIDI, Humdrum, MusicXML, MEI- όπως και σε μορφοποιήσεις που χρησιμοποιούνται από λογισμικά γραφικής απόδοσης μουσικής σημειογραφίας (πχ MuseScore). Λόγω της ολοκληρωτικής του υλοποίησης σε Python, το Music21 μπορεί να συνδυαστεί με πολλές άλλες βιβλιοθήκες (υλοποιημένες επίσης σε Python) και να ενσωματωθεί σε διαφόρων ειδών λογισμικά πακέτα όπως προγράμματα οπτικοποίησης της μουσικής πληροφορίας, προγράμματα αναζήτησης και ανάλυσης σε μουσικές βάσεις δεδομένων κλπ. Τελικά, το εξαγόμενο προϊόν δύναται να είναι κωδικοποιημένο σε όλες τις

μορφοποιήσεις που προαναφέρθηκαν. Δίνεται, επίσης, και δυνατότητα απευθείας γραφικής απόδοσης μικρών τμημάτων μουσικής.

Κάποιες απλές διεργασίες που μπορούν να πραγματοποιηθούν με χρήση του Music21 φαίνονται στην Εικόνα 3-1 και στην Εικόνα 3-2:

```
Convert a file from Humdrum's **kern data format to MusicXML for editing in Finale or Sibelius:  
parse('/users/documents/composition.krn').  
write('xml')
```

Εικόνα 3-1: Μετατροπή ενός αρχείου Humdrum σε MusicXML

```
Display a simple melody in musical notation:  
tinyNotation.TinyNotationStream(  
    "c4 d8 f g16 a g f#", "3/4").show()
```



Εικόνα 3-2: Γραφική απόδοση μιας μουσικής φράσης

Ο σκελετός αυτού του API περιλαμβάνει πολλές κλάσεις αντικειμένων (Pitch, Chord, Duration, Time Signature, Instrument...) που περιγράφουν ολοκληρωμένα τα μουσικά σύμβολα που συναντά κανείς σε παρτιτούρα. Μέσω κλήσης μεθόδων αυτών των κλάσεων, τα αντικείμενα μπορούν να εκτελέσουν διεργασίες ανάλυσης και μετασχηματισμού. Για παράδειγμα στην κλάση Chord υπάρχουν μέθοδοι που χρησιμοποιούνται για την εύρεση της νότας ρίζας της συγχορδίας (η νότα με το χαμηλότερο ύψος) ή για τον υπολογισμό της ανάλογης συγχορδίας στενής θέσης (τα ύψη των νοτών που συμμετέχουν είναι όσο το δυνατόν πιο κοντά). Ο αντικειμενοστραφής χαρακτήρας της βιβλιοθήκης διευκολύνει, επίσης, την επέκταση των δυνατοτήτων των υπάρχουσών κλάσεων καθώς και την κατασκευή νέων, με βάση τις ανάγκες της εκάστοτε εργασίας.

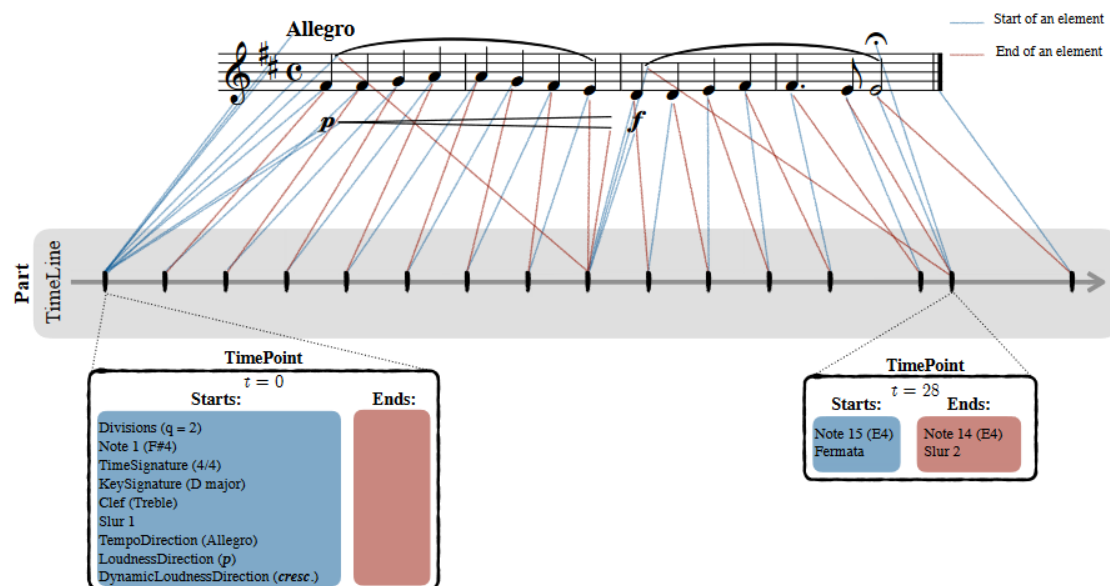
Η καινοτομία, όμως, του Music21 έγκειται σε ένα βασικό χαρακτηριστικό του πυρήνα του, τις ομαδοποιήσεις (Streams) της μουσικής πληροφορίας σε περιέκτες που μπορούν οι ίδιοι να εμφωλεαστούν. Έτσι, δημιουργείται ένα ιεραρχικό δέντρο ομαδοποιήσεων που προάγει την οργάνωση των ξεχωριστών μουσικών γεγονότων βάσει, για παράδειγμα, της φωνής ή της πάρτης στην οποία ανήκουν. Τα στοιχεία των ομαδοποιήσεων είναι προσιτά μέσω μεθόδων όπως η “getElementById”, παρομοίως με το μοντέλο πρόσβασης σε στοιχεία (elements) XML ή HTML εγγράφων. Οι ομαδοποιήσεις πληροφορίας μπορούν οποιαδήποτε στιγμή να μεταφραστούν στην επιθυμητή μορφοποίηση (πχ MusicXML ή Lilypond) και δυναμικά να εξαχθούν σε ένα πρόγραμμα απόδοσης μουσικής σημειογραφίας (Musescore, Lilypond) και να αποδοθούν γραφικά (Kuthbert et al, 2010).

3.1.2 Partitura

Το Partitura είναι μια προγραμματιστική βιβλιοθήκη ανοιχτού κώδικα που πρωτοεκδόθηκε πρόσφατα, το 2019. Είναι υλοποιημένη εξ'ολοκλήρου στην γλώσσα Python. Στόχος της δημιουργίας της είναι η διευκόλυνση διαφόρων εργασιών του επιστημονικού τομέα της Ανάκτησης Μουσικής Πληροφορίας (Music Information Retrieval) μέσω παραγωγής μουσικής πληροφορίας εκφρασμένης με κατάλληλο τρόπο, εύκολα και γρήγορα αναγνώσιμο από μηχανές. Ασχολείται, λοιπόν, με τον μετασχηματισμό των διαφόρων μορφότυπων κωδικοποίησης μουσικής πληροφορίας (MEI, MusicXML, humdrum, MIDI) σε μια απλή αντικειμενοστραφή δομή και αντίστροφα.

Στο Partitura, το υψηλότερο επίπεδο στην ιεραρχία της οργάνωσης ορίζεται από ένα ή περισσότερα αντικείμενα της κλάσης “Part” που αντιπροσωπεύει μια πάρτη, δηλαδή τη μουσική ενός οργάνου. Κάθε τέτοιο αντικείμενο εμπεριέχει ένα αντικείμενο τύπου

“TimeLine” που αποτελεί το χρονοδιαγραμματικό σκελετό πάνω στον οποίο δομείται μια αλληλουχία από αντικείμενα τύπου “TimePoint”. Το καθένα από αυτά δηλώνει μια χρονική θέση στην παρτιτούρα και μπορεί να συσχετιστεί με αντικείμενα άλλων κλάσεων, για παράδειγμα “Note” ή “Slur” που υποδηλώνουν φθογγόσημα και σημεία legato αντίστοιχα, ορίζοντας την έναρξη ή τη λήξη τους (Grachten et al, 2022). Η Εικόνα 3-3 επεξηγεί σχηματικά τη μετάφραση μια παρτιτούρας:



Εικόνα 3-3: Σχηματική επεξήγηση της δομής οργάνωσης του Partitura

3.1.3 Jmusic

Το JMusic είναι ένα πακέτο κλάσεων υλοποιημένων στη γλώσσα προγραμματισμού Java, με κύριο στόχο τη διαμόρφωση ενός προγραμματιστικού περιβάλλοντος κατάλληλου για μουσική σύνθεση καθώς και για σύνθεση ήχου σε μη πραγματικό χρόνο. Έχει ωστόσο χρησιμοποιηθεί και για άλλους σκοπούς μεταξύ των οποίων η ανάλυση μουσικών δεδομένων και η δημιουργία/επεξεργασία/απόδοση μουσικής σημειογραφίας. Ξεκίνησε να αναπτύσσεται το 1987 στα πλαίσια ενός μεταπτυχιακού προγράμματος σπουδών του Queensland University of Technology στην Αυστραλία και έκτοτε έχει επανασχεδιαστεί και επεκταθεί για να αρχίσει να διανέμεται το 1998 σύμφωνα με τους όρους της Γενικής Άδειας Δημόσιας Χρήσης GNU (ανοιχτού κώδικα). Η βιβλιοθήκη δίνει δυνατότητα εισαγωγής/εξαγωγής αρχείων MIDI μορφοποίησης καθώς και εξαγωγής audio αρχείων (συγκεκριμένα μορφοποιήσεων .au). Από το 2001 και μετά μπορεί επίσης να επεξεργαστεί δεδομένα MusicXML αναπαραστάσεων.

Στη JMusic βασικό αντικείμενο είναι το “Note” που υποδηλώνει ένα μουσικό γεγονός και δέχεται τιμές στις παραμέτρους του που αντιπροσωπεύουν ιδιότητες όπως ύψος, διάρκεια ή ακουστότητα. Η διάρκεια του γεγονότος μπορεί να καταχωρηθεί με σχετικό και απόλυτο τρόπο με τις παραμέτρους “RhythmValue” και “Duration” αντίστοιχα. Το ύψος εκφράζεται είτε ως MIDI νότα (μεσαίο NTO=60) είτε ως συχνότητα σε Hz, ενώ η ένταση ως ένας ακέραιος από το 0 έως το 127. Ο σκελετός για την οργάνωση αυτών των αντικειμένων σε μια αλληλουχία μουσικών γεγονότων, μια μουσική φράση, παρέχεται από την κλάση “Phrase” που αντιπροσωπεύει μια μονοφωνική μελωδία. Μουσικές φράσεις που παίζονται με το ίδιο

όργανο μπορούν να συλλεχθούν σε μια πάρτη, με το “Part”, και πολλές πάρτες δομούν ένα “Score”, το υψηλότερο στην ιεραρχία αντικείμενο.

Υπάρχουν δύο τρόποι για να προσδιορίσει καμία τη χροιά με την οποία θα αναπαραχθούν οι πάρτες των συνθέσεων. Είτε επιλέγει ένα αντικείμενο κλάσης Inst (Instrument) που έχει αποθηκευμένα όλα τα όργανα του General Midi Specification ως σταθερές είτε κατασκευάζει μέσω μιας διαδικασίας επεξεργασίας σήματος μια καινούργια χροιά (Sorensen et al, 2000).

Ένα παράδειγμα κώδικα που χρησιμοποιεί JMusic φαίνεται στην Εικόνα 3-4:

```
import jm.JMC;
import jm.music.data.*;
import jm.midi.*;
import jm.audio.*;
import jm.audio.synth.*;
import jmInst.*;

public final class Scale implements JMC{
    public static void main(String[] args){
        Score s = new Score("JMDemo - Audio Scale");
        Part p = new Part("Flute", 1, 0);
        Phrase phr = new Phrase(0.0);

        TriangleInst ti = new TriangleInst(44100);
        Instrument[] ensemble = {ti};

        for(short i=0;i<12;i++){
            Note n = new Note(i+60, Q, 127, 0.5);
            phr.addNote(n);
        }

        p.addPhrase(phr);
        s.addPart(p);
        s.writeMIDI("Scale.mid");
        s.writeAU("Scale.au", ensemble);
    }
}
```

Εικόνα 3-4: Παράδειγμα κώδικα με χρήση της JMusic

Στον κώδικα του παραδείγματος αφού εισαχθούν τα επιθυμητά τμήματα της JMusic, δημιουργείται ένα αντικείμενο παρτιτούρας “s” με τίτλο “JMDemo-Audio Scale”, μια πάρτη “p” με τίτλο “Flute” και μια μουσική φράση “phr”, στην οποία προστίθενται σειριακά οι νότες της οκτάβας που ξεκινά από το μεσαίο Ντο. Τέλος, η σύνθεση εξάγεται σε μορφοποίηση MIDI και σε .au και συσχετίζεται με τη χροιά ενός τριγωνικού ταλαντωτή “ti”.

3.1.4 VexFlow

Το VexFlow είναι μια προγραμματιστική βιβλιοθήκη προς γραφική απόδοση μουσικής σημειογραφίας σε μορφή παρτιτούρας ή ταμπλατούρας. Είναι υλοποιημένη εξ’ ολοκλήρου στη γλώσσα προγραμματισμού JavaScript και άρα κατάλληλη για χρήση σε διαδικτυακές εφαρμογές. Το τελικό προϊόν, η ψηφιακή παρτιτούρα, παράγεται από κώδικα JavaScript και είτε σχεδιάζεται απευθείας σε ένα στοιχείο (element) της HTML5 (5^η προτυποποιημένη έκδοση της HTML), το <canvas>, είτε αποτελείται από γραφικά τύπου SVG (Scalable Vector Graphics) που εισάγονται σε ένα γενικού σκοπού στοιχείο HTML, το <div>. Έχει ελάχιστες εξαρτήσεις από άλλα API’s, γεγονός που αυξάνει τη συμβατότητά της σε διάφορες περιστάσεις. Διατίθεται ελεύθερα κάτω από την άδεια MIT για το λογισμικό ανοικτού κώδικα (Slovak et al, 2022).

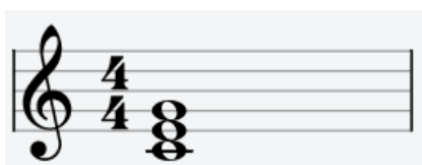
Στην Εικόνα 3-5 απεικονίζεται ένα παράδειγμα κώδικα JavaScript που χρησιμοποιεί τη VexFlow για να αποδώσει τελικά τη συγχορδία Ντο Μείζονα που φαίνεται στην Εικόνα 3-6:

```

var VF = Vex.Flow,
div = document.getElementById("staff"),
renderer = new VF.Renderer(div, VF.Renderer.Backends.SVG);
renderer.resize(500, 500);
var context = renderer.getContext();
context.setFont("Arial", 10, "").setBackgroundFillStyle("#eed");
var stave = new VF.Stave(10, 40, 200);
stave.addClef("treble").addTimeSignature("4/4");
stave.setContext(context).draw();
var notes = [new VF.StaveNote({ keys: ["c/4", "e/4", "g/4"], duration: "w" })
];
var voice = new VF.Voice({num_beats: 4, beat_value: 4});
voice.addTickables(notes);
var formatter = new VF.Formatter().joinVoices([voice]).format([voice], 200);
voice.draw(context, stave);

```

Εικόνα 3-5: Απόσπασμα κώδικα Javascript, στον οποίο γίνεται χρήση της Vexflow

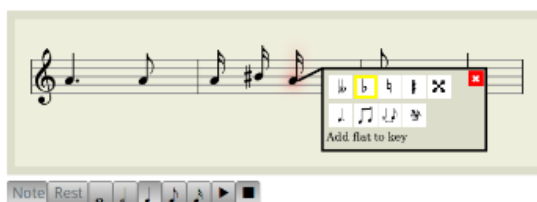


Εικόνα 3-6: Η γραφική απόδοση του κώδικα της Εικόνας 3-5

Στις παραπάνω γραμμές κώδικα καθορίζονται το τμήμα της ιστοσελίδας, μέσα στο οποίο θα σχεδιαστεί το πεντάγραμμα (<div id="staff"></div>), ο τύπος της επιφάνειας (CANVAS ή SVG) και το μέγεθος του πλαισίου, η διαμόρφωση του περιεχομένου (context), το μέγεθος και η θέση του πενταγράμμου, το κλειδί και ο ρυθμός. Καλείται η μέθοδος draw για τη σχεδίαση του πενταγράμμου και στη συνέχεια ορίζονται τα φθογγόσημα

με τη διάρκειά τους, η φωνή που θα τα περιέχει και η θέση τους μέσα στο πεντάγραμμα. Όλα τα παραπάνω αντικείμενα καθορίζουν με μεγάλη λεπτομέρεια και ακρίβεια τα προς απόδοση μουσικά σύμβολα.

Βάσει της VexFlow, λειτουργούν διάφορες εφαρμογές επεξεργασίας περιεχομένου μουσικής σημειογραφίας. Το VexUI και το VexFlow Notation Editor είναι δύο τέτοιες απλές και εύχρηστες εφαρμογές που παρέχουν μια βάση πενταγράμμου, στο οποίο ο χρήστης πραγματοποιεί με το ποντίκι βασικές λειτουργίες, όπως η προσθήκη/αφαίρεση στοιχείων (φθογγόσημα, συγχορδίες, παύσεις), η επισήμανση ενός επιλεγμένου στοιχείου, ο ορισμός του κλειδιού και του χρονοδείκτη, ο καθορισμός μουσικών μέτρων κλπ. Για τους χρήστες που δεν γνωρίζουν προγραμματισμό, προσφέρεται το VexTab, ένα περιβάλλον για δημιουργία μουσικού κειμένου και ταμπλατούρας κιθάρας με χρήση κειμένου. Η απεικόνιση των μουσικών συμβόλων γίνεται με τη μορφή SVG, τα οποία αποδίδονται στην προκαθορισμένη ετικέτα div της ιστοσελίδας και είναι αρκετά ικανοποιητική. Στην Εικόνα 3-7, στην Εικόνα 3-8, καθώς και στην Εικόνα 3-9 απεικονίζονται στιγμιότυπα από τις παραπάνω εφαρμογές



Εικόνα 3-7: Στιγμιότυπο από την εφαρμογή VexUI



Εικόνα 3-8: Στιγμιότυπο από την εφαρμογή VexFlow Notation Editor



Εικόνα 3-9: Στιγμιότυπο από την εφαρμογή VexTab

3.1.5 Verovio

Το 2013, το RISM (Répertoire International des Sources Musicales) Digital Center ξεκίνησε να αναπτύσσει το Verovio με στόχο την αποκωδικοποίηση αρχείων MEI σε ψηφιακή παρτιτούρα χωρίς ενδιάμεσες μετατροπές. Το Verovio χρησιμοποιήθηκε για την γραφική απόδοση μουσικών έργων της βιβλιοθήκης του RISM, κωδικοποιημένων σε κώδικα Plain και Easy καθώς και σε κώδικα DARMS. Η απόδοση των έργων σε παρτιτούρα πραγματοποιήθηκε μέσω ενδιάμεσης μετατροπής σε κωδικοποίηση MEI.

Το Verovio σχεδιάστηκε ως ένα γρήγορο και ελαφρύ σετ εργαλείων, υλοποιημένο αρχικά σε γλώσσα C++. Ο κώδικας του είναι απόλυτα φορητός και συμβατός με διαφορετικών τύπων χρήσεις, αφού δεν εξαρτάται από άλλες προγραμματιστικές βιβλιοθήκες για τη λειτουργία του. Μπορεί να χρησιμοποιηθεί απευθείας μέσω ενός διερμηνέα γραμμής εντολών ή ενσωματωμένο σε διάφορες προγραμματιστικές εργασίες ως βιβλιοθήκη. Επίσης στον επίσημο ιστότοπό του προσφέρεται έκδοση υλοποιημένη σε γλώσσα Javascript⁷. Η τελευταία περίπτωση παρουσιάζει ιδιαίτερο ενδιαφέρον καθώς προσφέρει τη δυνατότητα γρήγορης απόδοσης κώδικα μουσικής πληροφορίας σε ψηφιακή παρτιτούρα εντός περιβάλλοντος περιηγητή (Pugin et al, 2014).

Το Verovio, λοιπόν, δέχεται σαν είσοδο ένα αρχείο MEI (ή κώδικα Plain και Easy ή DARMS κώδικα) και παράγει στην έξοδο μορφοποιήσεις SVG, οι οποίες στη συνέχεια αποδίδονται

⁷ <https://book.verovio.org/installing-or-building-from-sources/javascript-and-webassembly.html>

(από το περιβάλλον εντός του οποίου εκτελείται) σε γραφικά SVG. Τα μουσικά σύμβολα που απαιτούνται για την απόδοση σημειογραφίας Δυτικής Μουσικής ή μετρικής σημειογραφίας παρέχονται στο πακέτο του Verovio. Τα πλεονεκτήματα της εξαγωγής μορφοποιήσεων SVG είναι πολλά. Αρχικά, διευκολύνεται η χρήση του Verovio σε διαδικτυακές εφαρμογές καθώς οι SVG μορφοποιήσεις μπορούν εγγενώς να αποδοθούν σε γραφικά από τους περισσότερους σύγχρονους περιηγητές. Επιπρόσθετα, αφού μια εικόνα SVG βασίζεται σε διανυσματικά γραφικά, η ψηφιακή παρτιτούρα μπορεί να τυπωθεί με υψηλή ποιότητα. Λειτουργικώς, κάθε παραγόμενο γραφικό στοιχείο είναι προγραμματιστικά προσβάσιμο, αφού η μορφοποίηση SVG είναι ένα σχήμα XML. Το γεγονός αυτό καθιστά την ψηφιακή παρτιτούρα αλληλεπιδραστική (για παράδειγμα η χρήστρια μπορεί να επισημάνει νότες ή άλλα μουσικά σύμβολα). Πέραν της προγραμματιστικής προσβασιμότητας, τα γραφικά SVG αντιστοιχίζονται απευθείας με στοιχεία ενός αρχείου MEI, αφού η δομή XML του μορφότυπου MEI μεταφέρεται και διατηρείται στην παραγόμενη μορφοποίηση SVG. Σαν αποτέλεσμα, η διαδικασία απόδοσης αρχείων MEI σε γραφικά SVG μέσω του Verovio δεν είναι μονόδρομος, αλλά μάλλον η σχέση μεταξύ των δύο μορφών μπορεί να ειπωθεί ως αμφίδρομη. Είναι, λοιπόν, δυνατή όχι μόνο η απόδοση παρτιτούρας από MEI κώδικα, αλλά και η επεξεργασία του κώδικα μέσω της παρτιτούρας⁸.

3.2 Πακέτα λογισμικού/εφαρμογές

Υπάρχει πληθώρα εφαρμογών που υλοποιείται με καθένα από αυτά τα API's, οι οποίες μπορούν να χρησιμοποιηθούν και σε πλαίσιο μουσικής εκπαίδευσης. Γενικά, οι εφαρμογές γραφικής απόδοσης μουσικής σημειογραφίας διανέμονται είτε ελεύθερα ή επί αντίτιμου. Στην ενότητα αυτή παρουσιάζονται τρεις εφαρμογές που διατίθενται δωρεάν και είναι ανοιχτού κώδικα.

3.2.1 MuseScore

Το MuseScore είναι ένα ελεύθερο και ανοιχτού κώδικα πακέτο λογισμικού μουσικής σημειογραφίας, διαθέσιμο για πολλά λειτουργικά συστήματα. Ο κύριος σκοπός του είναι η δημιουργία υψηλής ποιότητας ψηφιακών μουσικών παρτιτούρων.

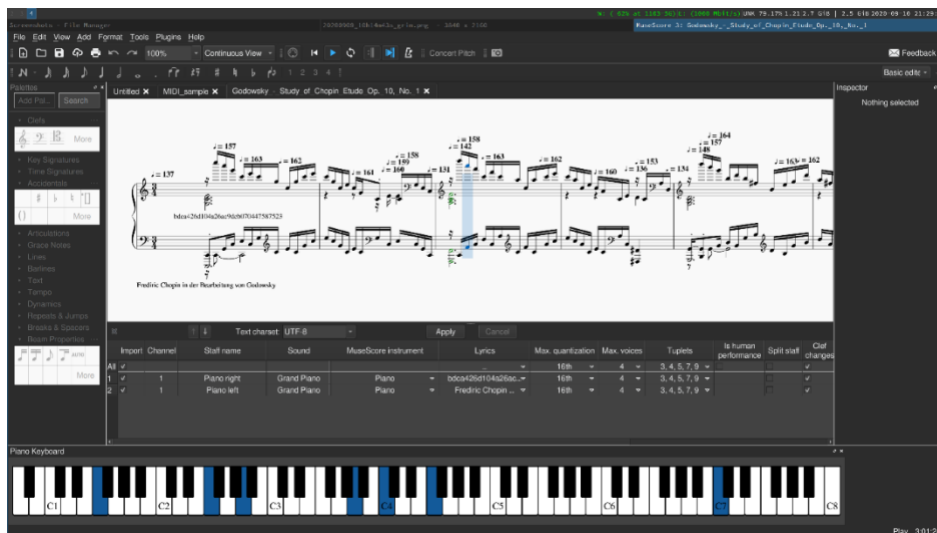
Η αρχική του έκδοση (2008) βασίστηκε στον κώδικα του MusE, ενός MIDI sequencer προγράμματος (Watson, 2018). Υποστηρίζεται από μια προοδευτικά αυξανόμενη κοινότητα προγραμματιστών, γεγονός που προωθεί την ευρεία χρήση του. Στην ιστοσελίδα του υπάρχει μια συνεχώς επεκτεινόμενη διαδικτυακή συλλογή στην οποία μπορεί κανείς να αναζητήσει ή να αποθηκεύσει παρτιτούρες. Έτσι συμβάλλει ενεργά στην ελεύθερη διαδικτυακή διακίνηση μουσικής πληροφορίας.

Στο περιβάλλον του MuseScore η μουσική πληροφορία απεικονίζεται με τη μορφή ψηφιακής παρτιτούρας. Έτσι, κατά τη διάρκεια της επεξεργασίας της, στην οθόνη φαίνεται απευθείας

⁸ <https://book.verovio.org/verovio-reference-book.pdf>

το τελικό προϊόν γραφικής απόδοσης. Αυτή η εύχρηστη και πλήρως αλληλεπιδραστική γραφική διεπαφή καθιστά δυνατή τη χρήση του λογισμικού από χρήστες χωρίς εμπειρία στον τομέα του προγραμματισμού. Κατά τα άλλα, το MuseScore προσφέρει όλες τις λειτουργίες που θα περίμενε κανείς από ένα χρηστικό πρόγραμμα απόδοσης μουσικής σημειογραφίας. Υποστηρίζει την ύπαρξη απεριόριστου αριθμού παρτών, τη σύνδεση συναφών παρτών μεταξύ τους (όπως στην έντυπη μορφή παρτιτούρων για ορχήστρα) καθώς και την εξαγωγή κάποιων πάρτης από το σύνολο του μουσικού έργου. Επίσης, πέρα από τη βασική πληροφορία του ύψους και της χρονικής διάρκειας του φθογγόσημου (ή της παύσης), του μουσικού κλειδιού και του χρονοδείκτη, δίνεται δυνατότητα κωδικοποίησης δακτυλισμών, δυναμικών, ζεύξεων, σημείων προσωδίας, σημείων επανάληψης, καλλωπισμών, στίχων και γενικά διαφόρων ειδών πληροφοριών που απεικονίζονται σε μια έντυπη παρτιτούρα. Δίνονται επίσης δυνατότητες αυτόματης μετατροπίας, καθώς και ψηφιακής αναπαράστασης πρώιμων μορφών σημειογραφίας και σημειογραφίας κρουστών. Υποστηρίζεται η εισαγωγή/εξαγωγή πληροφορίας MIDI και η παρτιτούρα μπορεί να αναπαραχθεί, αφού μετατραπεί αυτόματα σε αλληλουχία MIDI, από τον ενσωματωμένο sequencer. Τέλος, ο χρήστης μπορεί να προσθέσει βασικά εφέ ή να προσαρμόσει τις σχετικές εντάσεις των παρτών μέσω ενός μείκτη. Το MuseScore μπορεί να εξαγάγει γραφικές (SVG, PDF, PNG) και audio μορφοποιήσεις (MP3, OGG, FLAC, WAV) καθώς και να αναγνώσει, πέραν των MIDI μορφότυπων, και MusicXML κωδικοποιήσεις⁹.

Παρακάτω, στην Εικόνα 3-10 φαίνεται ένα στιγμιότυπο του περιβάλλοντος του προγράμματος:



Εικόνα 3-10: Στιγμιότυπο του περιβάλλοντος του MuseScore

3.2.2 LilyPond

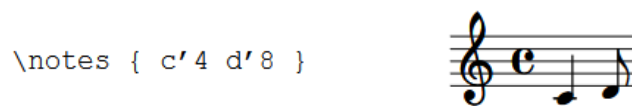
Το LilyPond είναι ένα αρθρωτό, επεκτάσιμο πακέτο λογισμικού ανεξάρτητο πλατφόρμας για την παραγωγή υψηλής ποιότητας μουσικής σημειογραφίας καθώς και ένας τύπος κωδικοποίησης μουσικής πληροφορίας. Το πρόγραμμα διανέμεται ελεύθερα σύμφωνα με τους όρους της Γενικής Άδειας Δημόσιας Χρήσης GNU και ο κώδικας του είναι ανοιχτός στην

⁹<https://musescore.org/>

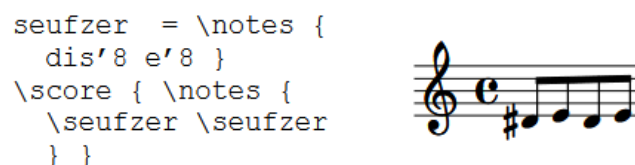
τροποποίηση. Ξεκίνησε να κυκλοφορεί το 1996 και έκτοτε συνεχώς βελτιώνεται από μια ενεργή κοινότητα εθελοντών προγραμματιστών. Το σύστημα είναι μερικώς υλοποιημένο στη γλώσσα προγραμματισμού Scheme, μέλος της οικογένειας προγραμματιστικών γλωσσών LISP. Στο πρόγραμμα είναι ενσωματωμένος ένας διερμηνέας της Scheme που δίνει στη χρήστρια δυνατότητες τροποποίησης υπαρχόντων εργαλείων του καθώς και προγραμματισμού νέων λειτουργικοτήτων.

Οι χρήστες εισάγουν την κωδικοποιημένη μουσική πληροφορία με το πληκτρολόγιο στη γραφική διεπαφή του προγράμματος. Η βασική μονάδα οργάνωσης σε αυτό το σύστημα κωδικοποίησης αποτελείται από μια νότα (εκφρασμένη στο σύστημα ABC) με καθορισμένη διάρκεια. Αυτές οι μονάδες μπορούν να συνδυαστούν για να σχηματίσουν μελωδικές φωνές ή συγχορδίες. Ένα παράδειγμα απεικονίζεται στην Εικόνα 3-11. Σε αυτό φαίνεται η χρήση, αφενός της δήλωσης “\notes” για την εισαγωγή νοτών, και αφετέρου των αγκύλων για την προσθήκη μιας μουσικής φράσης, δηλαδή δύο διαδοχικών νοτών. Με τη λέξη “\simultaneous” υποδηλώνεται η εισαγωγή νοτών παράλληλων στο χρόνο, είτε πρόκειται για διαφορετικές φωνές είτε για διαφορετικές πάρτες. Επίσης, δίνεται η δυνατότητα εκχώρησης, σε μια μεταβλητή, του κώδικα ενός τμήματος μουσικής, η οποία στη συνέχεια μπορεί να χρησιμοποιηθεί πολλαπλές φορές. Προσφέρεται, έτσι, μια λειτουργικότητα εξαγωγής πάρτης, καθώς αυτή μπορεί να εκχωρηθεί σε μεταβλητή, της οποίας η διαχείριση πραγματοποιείται διακριτά από το υπόλοιπο μουσικό έργο. Στην Εικόνα 3-12 φαίνεται ένα παράδειγμα. Τέλος, μπορεί ακόμα και να γραφεί κώδικας σε γλώσσα Scheme, για παράδειγμα συναρτήσεις με δυνατότητα πρόσβασης σε, και αλλαγής μουσικών φράσεων.

Το γεγονός αυτό διευκολύνει την ανάλυση, την επεξεργασία και την παραγωγή μουσικών δεδομένων με προγραμματιστικό τρόπο.



Εικόνα 3-11: Παράδειγμα γραφικής απόδοσης μιας μουσικής φράσης



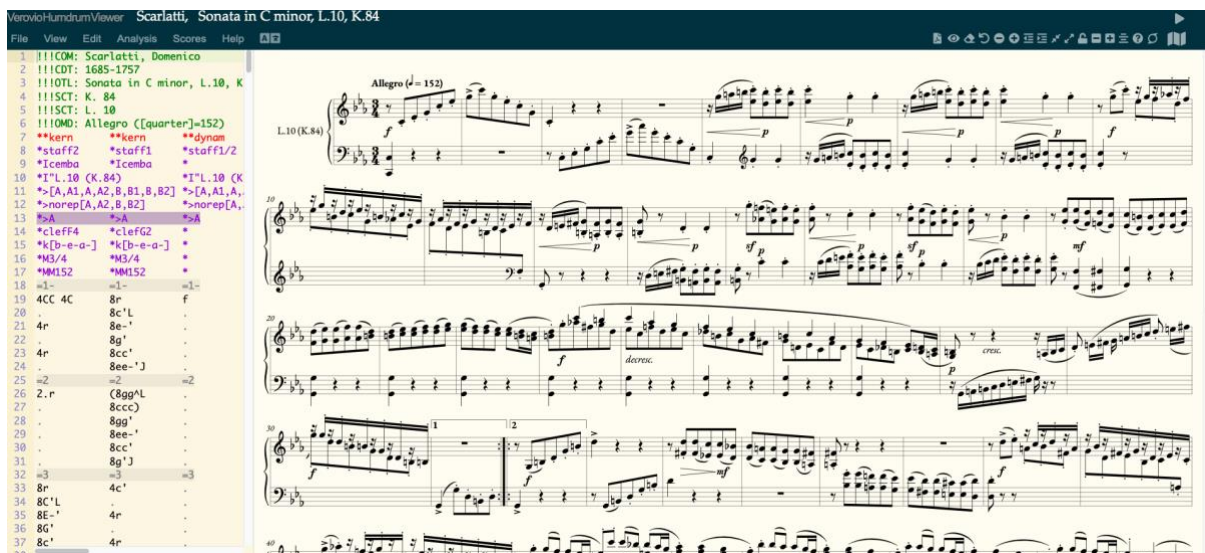
Εικόνα 3-12: Παράδειγμα εκχώρησης μιας μουσικής φράσης σε μεταβλητή και η γραφική της απόδοση

Το LilyPond, μετά την εισαγωγή των δεδομένων από το χρήστη, πραγματοποιεί ανάλυση και κατασκευή ενός συντακτικού δέντρου, παρομοίως με έναν διαδικτυακό περιηγητή που αναλύει κώδικα HTML. Στη συνέχεια τα μουσικά γεγονότα μεταφράζονται σε γραφικά αντικείμενα, τα οποία σχηματίζουν μια παρτιτούρα που μορφοποιείται περαιτέρω με έναν αυτοματοποιημένο τρόπο (Nienhuys et al, 2003). Τελικά, εξάγεται μια ψηφιακή παρτιτούρα σε μορφοποίηση αρχείου Postscript/PDF/SVG/PNG, η οποία μπορεί στη συνέχεια να

τυπωθεί ή να υποστεί περαιτέρω επεξεργασία για άλλες χρήσεις, πχ. τη διαδικτυακή κυκλοφορία. Δίνεται, επίσης δυνατότητα εξαγωγής αρχείου MIDI.

3.2.3 Verovio Humdrum Viewer

Το Verovio Humdrum Viewer είναι κατά βάση μια διαδικτυακή αλληλεπιδραστική εφαρμογή απόδοσης μουσικής σημειογραφίας, που χρησιμοποιεί το Veronio ως μηχανή παραγωγής γραφικών SVG. Εκτός του Veronio, ενσωματωμένη στον κώδικα του VHV είναι και η βιβλιοθήκη Humlib, που είναι υλοποιημένη στη γλώσσα C++ και προσφέρει δυνατότητες ανάλυσης της πληροφορίας αρχείων Humdrum, καθώς και μια σειρά εργαλείων για τον αυτόματο μετασχηματισμό της. Ο κώδικας του VHV διατίθεται ελεύθερα στο διαδίκτυο. Η γραφική διεπαφή της εφαρμογής αποτελείται από ένα τμήμα εισαγωγής κειμένου στα αριστερά και ένα κομμάτι στο οποίο εμφανίζεται η ψηφιακή παρτιτούρα στα δεξιά. Στην Εικόνα 3-13 απεικονίζεται ένα στιγμιότυπό της:

The image shows a screenshot of the Verovio Humdrum Viewer web application. The interface is split into two main sections. On the left, there is a sidebar with a menu (File, View, Edit, Analysis, Scores, Help) and a list of musical data elements such as 'I:ICOM: Scarlatti, Domenico', 'I:IDT: 1685-1757', and 'I:ITL: Sonata in C minor, L.10, K.84'. Below this list is a table of musical symbols and their corresponding Humdrum codes. On the right, the main area displays a musical score for 'Allegro (♩ = 152)'. The score is presented in a multi-staff format, showing both the treble and bass clefs. The notation includes various musical symbols like notes, rests, and dynamic markings such as 'p' (piano) and 'f' (forte). The score is labeled 'L.10 (K.84)' and includes performance instructions like 'Allegro' and 'Allegro'.

Εικόνα 3-13: Στιγμιότυπο από το Verovio Humdrum Viewer

Ο χρήστης μπορεί να εισάγει τη μουσική πληροφορία στο τμήμα του επεξεργαστή κειμένου μέσω drag ή drop ή απευθείας πληκτρολόγησης. Η πληροφορία προς εισαγωγή πρέπει να είναι κωδικοποιημένη στις μορφοποιήσεις MusicXML, MEI, Humdrum. Τελικά, όλες οι μορφοποιήσεις πριν τη γραφική τους απόδοση από το Veronio μετατρέπονται σε MEI. Πέραν της γραφικής απόδοσης, η μουσική πληροφορία ενός αρχείου μπορεί επίσης να μεταγραφεί σε MIDI κωδικοποίηση και να αναπαραχθεί ως Standard Midi File.

Στο VHV η επεξεργασία της μουσικής πληροφορίας πραγματοποιείται είτε άμεσα, μέσω του επεξεργαστή κειμένου, είτε αλληλεπιδρώντας με τα γραφικά. Πρέπει να σημειωθεί ότι δυνατότητα επεξεργασίας της μουσικής πληροφορίας με γραφικό τρόπο παρέχεται μόνο όταν αυτή είναι κωδικοποιημένη σε μορφότυπο Humdrum. Η εφαρμογή δίνει τη δυνατότητα επιλογής μουσικών γεγονότων, τροποποίησης του ύψους τους καθώς και προσθήκης/αφαίρεσης/επεξεργασίας αλλοιώσεων, καλλωπισμών, σημείων προσωδίας, ζεύξεων¹⁰.

¹⁰ <https://verovio.humdrum.org>

3.3 Σύνοψη

Οι προγραμματιστικές βιβλιοθήκες που παρουσιάστηκαν χρησιμοποιούνται σε εργασίες διαφορετικού σκοπού, αφού έχουν σχεδιαστεί βάσει διαφορετικών στοχεύσεων. Ως προς τη γραφική απόδοση μουσικής σημειογραφίας, η βιβλιοθήκη Music21 παρέχει δυνατότητα απόδοσης μικρών μόνο τμημάτων μουσικής, ενώ η Partitura μπορεί να εξάγει μόνο αρχεία κωδικοποιημένης πληροφορίας, συγκεκριμένα MIDI και MusicXML. Ωστόσο και οι δύο βιβλιοθήκες είναι αρκετά κατάλληλες για χρήση σε εργασίες Ανάκτησης Μουσικής Πληροφορίας. Επιπρόσθετα, η βιβλιοθήκη JMusic δημιουργήθηκε κυρίως για σύνθεση μουσικής και ήχου, αν και χρησιμοποιήθηκε και για δημιουργία παρτιτούρας, αφού όταν αρχικά εκδόθηκε δεν υπήρχαν πολλά εργαλεία για γραφική απόδοση μουσικής σημειογραφίας. Από την άλλη, οι πιο πρόσφατα εκδοθείσες βιβλιοθήκες Veronio και VexFlow σχεδιάστηκαν για αυτόν ακριβώς το σκοπό. Όπως προαναφέρθηκε, μεταξύ άλλων παρέχουν τη δυνατότητα εξαγωγής μορφοποιήσεων SVG, γεγονός που τις καθιστά κατάλληλες για χρήση σε προγραμματιστικές εργασίες, στις οποίες απαιτείται υψηλή ποιότητα γραφικής απόδοσης μουσικής σημειογραφίας καθώς και αλληλεπιδραστικότητα. Πιο κάτω, στον Πίνακα 3-1: Μορφοποιήσεις που εξάγονται ανά API φαίνονται οι μορφοποιήσεις που μπορούν να εξαχθούν ανά API:

	Music21	Partitura	JMusic	VexFlow	Veronio
Μορφότυποι μουσικής σημειογραφίας	MIDI, humdrum, musicXML, MEI	MIDI, musicXML	MIDI		
Μορφότυποι γραφικών	Μορφότυποι Bitmap			Μορφότυποι SVG	Μορφότυποι SVG
Μορφότυποι audio			Au		

Πίνακας 3-1: Μορφοποιήσεις που εξάγονται ανά API

Στον Πίνακα 3-2 που ακολουθεί, συνοψίζονται οι γλώσσες προγραμματισμού στις οποίες είναι υλοποιημένες οι βιβλιοθήκες που παρουσιάστηκαν:

API	Music21	Partitura	JMusic	VexFlow	Veronio
Γλώσσα υλοποίησης	Python, Javascript	Python	Java	Javascript	C++, Javascript

Πίνακας 3-2: Γλώσσες υλοποίησης των API's

Όπως φαίνεται στον Πίνακα 3-2, οι βιβλιοθήκες VexFlow, Veronio και Music21 είτε είναι υλοποιημένες απευθείας στη γλώσσα Javascript ή έχουν μεταγραφεί σε αυτή. Το γεγονός αυτό παρέχει εγγενές πλεονέκτημα όσον αφορά τη χρήση αυτών των API's σε διαδικτυακές εφαρμογές που εκτελούνται σε περιβάλλον περιηγητή.

Όσον αφορά τις εφαρμογές που παρουσιάστηκαν στην αντίστοιχη ενότητα, όλες εξειδικεύονται στην γραφική και ηχητική απόδοση μουσικής σημειογραφίας. Ενώ, όμως, το

Musescore και το Lilypond είναι πακέτα λογισμικού που εκτελούνται αυτόνομα, το VHV αποτελεί εφαρμογή που εκτελείται σε περιβάλλον περιηγητή. Επίσης, το VHV και το Musescore διαθέτουν αλληλεπιδραστικές γραφικές διεπαφές σε αντίθεση με το Lilypond, στο οποίο τα δεδομένα εισάγονται αποκλειστικά με το πληκτρολόγιο στον επεξεργαστή κειμένου της εφαρμογής. Διαφορές μεταξύ των εφαρμογών υπάρχουν και στους μορφότυπους που μπορούν να αναγνώσουν. Στον Πίνακα 3-3 απεικονίζονται οι εγγενώς αναγνώσιμοι μορφότυποι ανά πακέτο λογισμικού:

	Musescore	Lilypond	VHV
MIDI	✓		
Humdrum			✓
MusicXML	✓		✓
MEI			✓
Ειδικοί μορφότυποι	MuseData	Lilypond	

Πίνακας 3-3: Αναγνώσιμοι μορφότυποι ανά εφαρμογή

Στον Πίνακα 3-4 φαίνονται οι μορφότυποι που μπορούν να αναγνωστούν ανά API:

	Music21	Partitura	JMusic	VexFlow	Verovio
MIDI	✓	✓	✓		
Humdrum	✓	✓			✓
MusicXML	✓	✓	✓		✓
MEI	✓	✓			✓
Ειδικοί μορφότυποι	Musescore, Lilypond, VexFlow, Musedata				

Πίνακας 3-4: Αναγνώσιμοι μορφότυποι ανά API

Πρέπει να σημειωθεί ότι δύνανται μορφότυποι να είναι αναγνώσιμοι από τις βιβλιοθήκες/εφαρμογές μέσω πρόσθετων εργαλείων. Για παράδειγμα, στον επίσημο ιστότοπο του Lilypond¹¹ προτείνονται μετατροπείς για τους μορφότυπους MIDI/MusicXML/ABC. Ωστόσο, η χρήση ειδικών αναπαραστάσεων μουσικής πληροφορίας χωρίς πρόβλεψη για εγγενώς δομική αντιστοιχισή τους με πιο κοινούς μορφότυπους (πχ. VexFlow/Lilypond) αποτελεί μειονέκτημα καθώς, όπως προαναφέρθηκε, κατά τη διάρκεια μετατροπών ενέχεται κίνδυνος απώλειας πληροφορίας.

¹¹ <https://lilypond.org/doc/v2.22/Documentation/usage/convert-from-other-formats>

4. ΥΛΟΠΟΙΗΣΗ

4.1 Το VHV στη μουσική εκπαίδευση

Οι προσαρμογές που έγιναν στον κώδικα του VHV και αποτελούν το πρακτικό τμήμα αυτής της διπλωματικής εργασίας και εντάσσονται στην υλοποίηση του MusiCoLab, ενός συνεργατικού ακαδημαϊκού ερευνητικού έργου στο οποίο συμμετέχει το Τμήμα Μουσικής Τεχνολογίας και Ακουστικής του Ελληνικού Μεσογειακού Πανεπιστημίου. Το MusiCoLab στοχεύει στη δημιουργία μιας διαδικτυακής πλατφόρμας κατάλληλης για μουσική εκπαίδευση από απόσταση. Η ανάπτυξή της βασίζεται σε θεωρίες συμμετοχικής μάθησης, κατά τις οποίες η μάθηση ως στόχος προκύπτει από την συνεργατική αλληλεπίδραση των μαθησιακών υποκειμένων με τα τεχνουργήματα που από κοινού κατασκευάζουν και επεξεργάζονται. Στην περίπτωση του MusiCoLab τέτοια τεχνουργήματα μπορεί να είναι ψηφιακά αρχεία ήχου ή και αρχεία ψηφιακής μουσικής σημειογραφίας.

Στο πλαίσιο αυτής της εργασίας υλοποιήθηκαν τρεις επεκτάσεις του λογισμικού VHV, στις γλώσσες Javascript, HTML, CSS. Η πρώτη, ο επεξεργαστής συγχορδιών, προσθέτει τη δυνατότητα αλλαγής συγχορδίας σε ένα αρχείο μουσικής σημειογραφίας ώστε στο πλαίσιο μουσικής μάθησης, να παρέχεται η δυνατότητα πειραματισμού με εναλλακτικές εναρμονίσεις ενός μουσικού έργου και κατ'επέκταση να επιτρέπει την ανάπτυξη των αυτοσχεδιαστικών ικανοτήτων των μαθητευόμενων. Η δεύτερη δίνει τη δυνατότητα στο χρήστη να μεταβάλλει το τέμπο του έργου, εργαλείο χρήσιμο για τις περιπτώσεις που επιλέγεται η MIDI αναπαραγωγή του (πχ υποβοηθητικά στην εκτέλεσή του από μαθητευόμενο). Τέλος, η τρίτη επέκταση επιτρέπει τον υπολογισμό του χρόνου κάθε μουσικού συμβάντος από την έναρξη του μουσικού έργου που αναπαρίσταται στην συμβολική μορφή. Ο χρόνος υπολογίζεται βάσει του τέμπο και του ρυθμού, ώστε να αντιστοιχίζεται στο μουσικό συμβάν που επιλέγεται από το χρονοδείκτη του VHV. Η λειτουργία αυτή επιτρέπει τη δυνατότητα συγχρονισμού και ευθυγράμμισης (alignment) μιας προ-ηχογραφημένης ερμηνείας του μουσικού έργου με την παρτιτούρα του. Ο συγχρονισμός επιτελείται με αλγόριθμους επεξεργασίας ψηφιακού σήματος που υλοποιούνται στο πλαίσιο άλλων δραστηριοτήτων του έργου.

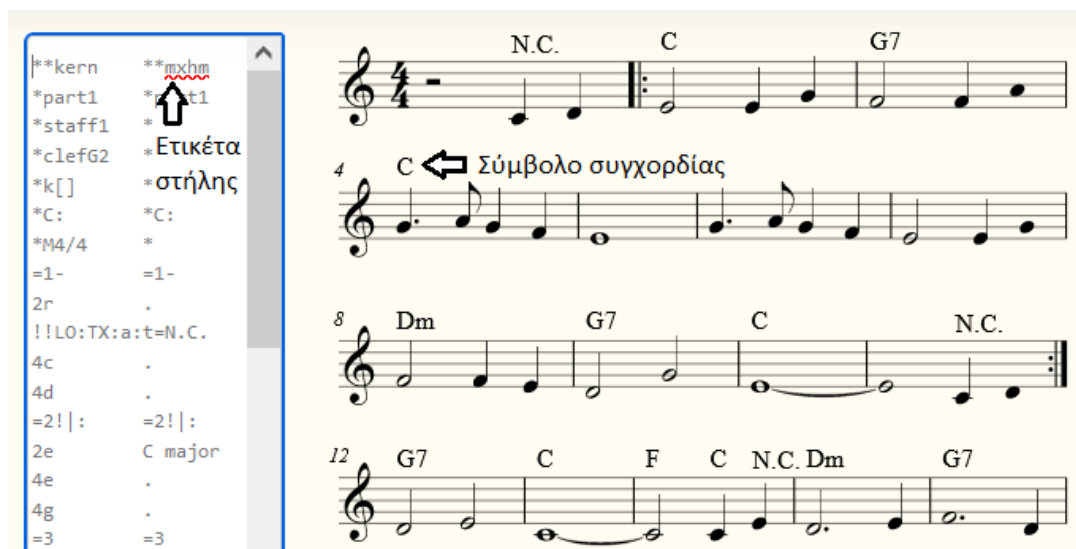
Στις επόμενες ενότητες παρουσιάζεται η υλοποίηση των τριών επεκτάσεων. Πρέπει να σημειωθεί ότι τα εργαλεία που προστέθηκαν παρέχουν δυνατότητες επεξεργασίας αρχείων κωδικοποιημένων μόνο σε μορφότυπο Humdrum.

4.2 Επεξεργαστής Συγχορδιών

Ο επεξεργαστής συγχορδιών είναι ένα εργαλείο, το οποίο δημιουργήθηκε για να διευκολύνει τη γραφική επεξεργασία των συγχορδιών του μουσικού έργου. Συγκεκριμένα ο χρήστης μπορεί να αντικαταστήσει μια υπάρχουσα συγχορδία με μια άλλη, αλληλεπιδρώντας με τη γραφική διεπαφή του VHV.

Σε ένα αρχείο Humdrum, τα σύμβολα συγχορδιών (chord symbols) κωδικοποιούνται με ένα προ-ορισμένο αναπαραστατικό σχήμα Humdrum σε μια ξεχωριστή στήλη (spine). Η ετικέτα του σχήματος μπορεί να είναι είτε η `**harm` είτε η `**mxhm` (συγχορδίες που εισήχθησαν από μορφότυπο MusicXML). Στην παρτιτούρα που παράγεται από το Veronio και

απεικονίζεται στη γραφική διεπαφή του VHV, η στήλη αυτή αποδίδεται γραφικά με σύμβολα συγχορδιών τύπου ταμπλατούρας πάνω από κάθε σειρά πεντάγραμμου. Στην Εικόνα 4-1: Τμήμα της γραφικής διεπαφής του VHV απεικονίζεται ένα στιγμιότυπο.



Εικόνα 4-1: Τμήμα της γραφικής διεπαφής του VHV

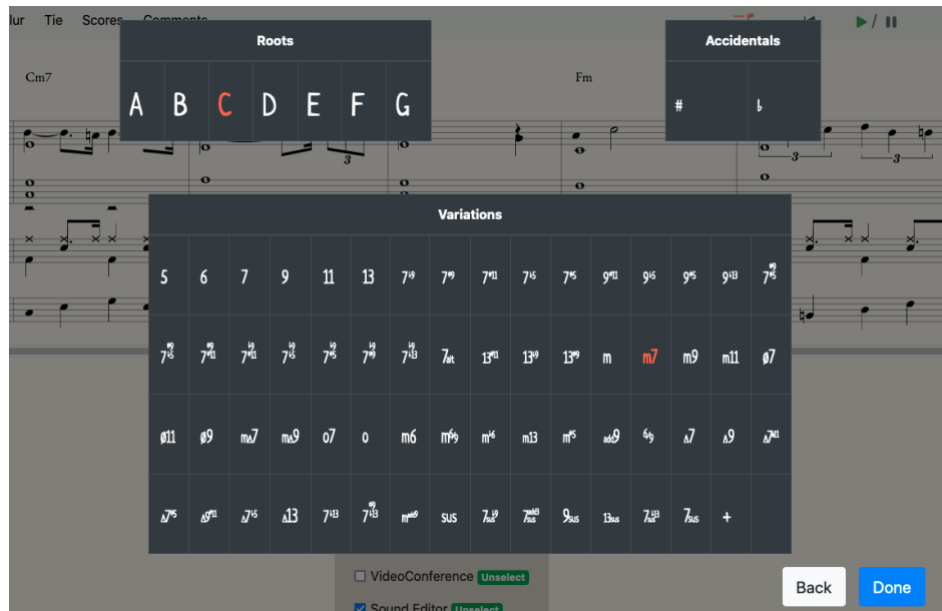
Με τον επεξεργαστή συγχορδιών πατώντας κλικ πάνω σε ένα σύμβολο συγχορδίας (chord symbol) στην παρτιτούρα εμφανίζονται δύο κουμπιά με τις επιλογές Edit και Suggest. Στην Εικόνα 4-2 φαίνεται τμήμα της γραφικής διεπαφής της εφαρμογής μετά το κλικ πάνω σε σύμβολο συγχορδίας.



Εικόνα 4-2: Τα κουμπιά που εμφανίζονται μετά το κλικ σε σύμβολο συγχορδίας

Αν πατηθεί κλικ στην επιλογή Edit εμφανίζεται στο προσκήνιο το κυρίως τμήμα του επεξεργαστή συγχορδιών που επιτρέπει στο χρήστη να επιλέξει τη νέα συγχορδία. Αποτελείται από τρία πλαίσια, στα οποία ο χρήστης επιλέγει το θεμέλιο φθόγγο (πχ Α για Λα, C για Ντο κλπ.), πιθανές αλλοιώσεις του θεμέλιου καθώς και τον τύπο της συγχορδίας (πχ ματζόρε πέμπτης, μινόρε τρίτης κλπ.). Κάτω από τα πλαίσια επιλογής της νέας συγχορδίας εμφανίζονται δύο κουμπιά, το Done και το Back για να καταχωρηθεί ή να

ακυρωθεί αντίστοιχα η αλλαγή. Στην Εικόνα 4-3 φαίνεται ένα στιγμιότυπο με επιλεγμένη τη συγχορδία Ντο μινόρε εβδόμης:



Εικόνα 4-3: Το κυρίως τμήμα του επεξεργαστή συγχορδίων μετά το κλικ στο κουμπί "Edit"

Από την άλλη, πατώντας κλικ στο κουμπί Suggest, προτείνεται μια εναλλακτική εναρμόνιση, παραγόμενη με αλγοριθμική διαδικασία.

Ο επεξεργαστής συγχορδίων υλοποιήθηκε στην γλώσσα προγραμματισμού Javascript και στις γλώσσες σήμανσης HTML και CSS. Η επέκταση αυτή αποτελείται από διάφορα στοιχεία τα οποία συνεργάζονται μεταξύ τους για την επίτευξη του αποτελέσματος.

Η γραφική διεπαφή του επεξεργαστή συγχορδίων, δηλαδή τα κουμπιά και οι πίνακες επιλογής νέας συγχορδίας, υλοποιήθηκε σε HTML και ενσωματώθηκε στο αρχείο του κώδικα της ιστοσελίδας που αφορά τη σύγχρονη διδασκαλία της πλατφόρμας μουσικής εκπαίδευσης Musicolab. Για τη στυλιστική διαμόρφωση των παραπάνω στοιχείων έγινε χρήση της πλατφόρμας CSS Bootstrap¹². Συγκεκριμένα για την υλοποίηση της γραφικής διεπαφής προστέθηκαν δύο περιέκτες div στο αρχείο HTML του κώδικα της ιστοσελίδας του VHV. Ο πρώτος περιέχει τα κουμπιά Edit και Suggest, ενώ ο δεύτερος τους πίνακες επιλογής νέας συγχορδίας.

Τα σύμβολα των συγχορδίων που εμφανίζονται στην Εικόνα 4-3 προσφέρθηκαν για τις ανάγκες υλοποίησης της παρούσας εφαρμογής από την εμπορική εφαρμογή Genius Jam Tracks¹³. Το GJT παρέχει μία σουίτα εφαρμογών εκμάθησης της Jazz μουσικής για το λειτουργικό iOS. Τα σύμβολα έχουν υλοποιηθεί ως ένα font family, του οποίου οι αντιστοιχίσεις παρατίθενται στο **Error! Reference source not found.**

Το κυρίως τμήμα της εφαρμογής υλοποιήθηκε σε Javascript. Γενικά, ο κώδικας του επεξεργαστή συγχορδίων εκτελείται στον περιηγητή του χρήστη της πλατφόρμας και έχει σαν τελικό αποτέλεσμα την αποστολή δεδομένων σε σχέση με την τρέχουσα και τη νέα

¹² <https://getbootstrap.com/docs/4.0/getting-started/introduction/>

¹³ <https://geniusiamtracks.com>

συγχορδία σε έναν εξυπηρετητή του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης (ΑΠΘ). Βάσει αυτών των δεδομένων ο εξυπηρετητής στέλνει πίσω ένα νέο αρχείο Humdrum με αλλαγμένη τη συγχορδία, το οποίο φορτώνεται αυτόματα στο VHV και αποδίδεται στη νέα παρτιτούρα. Τα δεδομένα που αποστέλλονται στον εξυπηρετητή του ΑΠΘ αποτελούνται από:

1. Τη θέση και το κείμενο της κωδικοποίησης της τρέχουσας συγχορδίας μέσα στο αρχείο Humdrum.
2. Μια παράμετρο που καθορίζει αν η επιλογή της νέας συγχορδίας πραγματοποιείται αλγοριθμικά ή από το χρήστη.
3. Τη νέα συγχορδία που επιλέγεται από το χρήστη (νότα ρίζας, αλλοιώσεις, τύπος).
4. Ολόκληρο το τρέχον αρχείο Humdrum.

Συγκεκριμένα, η διαδικασία ξεκινά με το κλικ σε ένα γραφικό αντικείμενο που εμπεριέχει σύμβολο συγχορδίας. Ένα παράδειγμα κώδικα τέτοιου αντικειμένου είναι:

```
<g id="harm-L25F8" class=" harm highlight">
```

Το παραπάνω αντικείμενο ταυτοποιείται ως περιέκτης συμβόλου συγχορδίας μέσω της προσθήκης του όρου “harm” στην τάξη (class) του. Επίσης, η ταυτότητα (id) ενός τέτοιου αντικειμένου εμπεριέχει την πληροφορία του σημείου κωδικοποίησής του μέσα στο αρχείο Humdrum. Στο παραπάνω παράδειγμα το σημείο είναι στη γραμμή 25 και στη στήλη 8 (L25F8). Στη συνέχεια καλείται ένα γεγονός onclick γενικής φύσης που προϋπήρχε στον κώδικα του VHV και αφορά το πάτημα κλικ εντός του σώματος της σελίδας. Στην περίπτωση που το αντικείμενο στο οποίο πατήθηκε κλικ απεικονίζει σύμβολο συγχορδίας ενεργοποιείται ο εξής κώδικας:

```
chordLocation.line = harmonyElem.id.split('L')[1].split('F')[0];  
chordLocation.column = harmonyElem.id.split('L')[1].split('F')[1];  
chordBtns.style.visibility = 'visible';
```

Με τον παραπάνω κώδικα, αρχικά ανακτάται η πληροφορία του σημείου στο αρχείο Humdrum, στο οποίο κωδικοποιείται το σύμβολο της τρέχουσας συγχορδίας. Η πληροφορία του σημείου αποθηκεύεται στο αντικείμενο “chordLocation”. Στη συνέχεια το πλαίσιο με τα κουμπιά Edit και Suggest (chordBtns) γίνεται ορατό στο χρήστη. Προς το τέλος του κώδικα του γεγονότος που πυροδοτήθηκε προστέθηκε η λειτουργικότητα ανάκτησης του κειμένου κωδικοποίησης της τρέχουσας συγχορδίας, το οποίο αποθηκεύεται στην παράμετρο “current” του αντικειμένου “chord”:

```
if (elmnt.classList.contains("harm") && elmnt.classList.contains("highlight")) {  
  let line = editor.session.doc.getLine(row-1);  
  let stopIndex = col;  
  while( stopIndex-col<line.length ) stopIndex++;  
  chord.current = line.substring(col, stopIndex);  
}
```

Με το πάτημα κλικ στο κουμπί Suggest, ενεργοποιείται ένα γεγονός onclick και καλείται μια συνάρτηση στην οποία εκτελούνται οι εξής λειτουργικότητες:

1. Ανακτάται το τρέχον αρχείο Humdrum και αποθηκεύεται στη μεταβλητή “kernfile”:

```
let edtr = getAceEditor();
if (!edtr) {
  throw new Error('Ace Editor is undefined');
}
let kernfile = edtr.session.getValue();
```

2. Τίθεται αληθής η παράμετρος “chord.reharmonize” που καθορίζει την παραγωγή της εναλλακτικής εναρμόνισης με αλγοριθμικό τρόπο:

```
chord.reharmonize = 'true';
```

3. Δημιουργείται ένα αντικείμενο “jsonFile” που περιέχει συγκεντρωτικά το αρχείο Humdrum, το κείμενο που κωδικοποιεί την τρέχουσα συγχορδία καθώς και τη θέση του μέσα στο αρχείο Humdrum καθώς και την παράμετρο που καθορίζει την αλγοριθμική εναρμόνιση:

```
let jsonRequest = {
  kernfile,
  chordLocation,
  chord,
};

console.log(jsonRequest);
let jsonFile = JSON.stringify(jsonRequest);
```

4. Αποστέλλεται το αντικείμενο “jsonFile” στον εξυπηρετητή του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης με την τεχνική AJAX (Asynchronous Javascript and XML):

```
let xhttp = new XMLHttpRequest();
xhttp.onload = function () {
  let jsonResponse = JSON.parse(xhttp.response);
  let newKern = jsonResponse[0];
  edtr.setValue(newKern);
};
xhttp.open("POST", url);
xhttp.setRequestHeader('Content-Type', 'application/json');
xhttp.send(jsonFile);
```

Από την άλλη, με το κλικ στο κουμπί Edit πυροδοτείται ένα άλλο γεγονός, το οποίο έχει σαν αποτέλεσμα την εμφάνιση του πλαισίου με τους πίνακες επιλογής νέας συγχορδίας (“chordEditor”) στη γραφική διεπαφή:

```
chordEditor.style.display = 'block';
```

Μετά την επιλογή με κλικ κάποιου κελιού των τριών πινάκων με τις νέες ρίζας, τις πιθανές αλλοιώσεις και τους τύπους της νέας συγχορδίας καλείται ένα γεγονός που έχει ως τελικό αποτέλεσμα την αποθήκευση της επιλογής στις μεταβλητές “chord.new.root”, “chord.new.accidental” και “chord.new.variation” αντίστοιχα:

```
chord.new.root = event.target.innerText;
```

```
chord.new.accidental = event.target.innerText;
```

```
chord.new.variation = event.target.innerText;
```

Η διαδικασία της επεξεργασίας ολοκληρώνεται με το κλικ στο κουμπί Done οπότε και καλείται η εξής “συνάρτηση βέλους” (arrow function):

```
sendBtn.addEventListener('click', function (event) {
  chordEditor.style.display = 'none';

  chordSelected.root.style.color = 'white';
  if (chordSelected.accidental) chordSelected.accidental.style.color = 'white';
  chordSelected.variation.style.color = 'white';
  Object.keys(chordSelected).forEach((i) => (chordSelected[i] = null));

  let edtr = getAceEditor();
  if (!edtr) {
    throw new Error('Ace Editor is undefined');
  }
  let kernfile = edtr.session.getValue();

  let jsonRequest = {
    kernfile,
    chordLocation,
    chord,
  };

  let jsonFile = JSON.stringify(jsonRequest);

  console.log(jsonRequest);
```



```

let xhttp = new XMLHttpRequest();
xhttp.onload = function () {

    let jsonResponse = xhttp.response;

    let newKern = jsonResponse[0];
    edtr.setValue(newKern);
};

xhttp.open("POST", 'http://155.207.188.7:6001/sending_kern?row=17&column=8&chord=Cm&kern=lalala');
xhttp.setRequestHeader('Content-Type', 'application/json');
xhttp.send(jsonFile);

Object.keys(chord.new).forEach((i) => (chord.new[i] = null));
sendBtn.style.display = 'none';

```

Η παραπάνω συνάρτηση εκτελεί μεταξύ άλλων παρόμοιες λειτουργίες με αυτές του κλικ στο κουμπί Suggest. Δηλαδή αρχικά αποθηκεύεται το τρέχον αρχείο Humdrum στη μεταβλητή “kernfile”. Στη συνέχεια κατασκευάζεται ένα αντικείμενο “jsonFile” που περιέχει το αρχείο Humdrum, το κείμενο που κωδικοποιεί την τρέχουσα συγχορδία καθώς και τη θέση του μέσα στο αρχείο Humdrum, όπως επίσης και τη νότα ρίζας, κάποια πιθανή αλλοίωση και τον τύπο της καινούργιας συγχορδίας. Το αντικείμενο “jsonFile” αποστέλλεται στον εξυπηρετητή του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης με την τεχνική AJAX (Asynchronous Javascript and XML).

4.3 Αλλαγή ταχύτητας αναπαραγωγής

Αυτή η επέκταση αφορά την αλλαγή της ταχύτητας αναπαραγωγής του μουσικού έργου. Η χρήστρια εισάγει το επιθυμητό τέμπο στο κατάλληλο πεδίο και πατά κλικ στο κουμπί Change!.

Στην Εικόνα 4-4: Η γραφική διεπαφή της μεταβολής τέμπο απεικονίζεται το τμήμα της γραφικής διεπαφής της εφαρμογής που αφορά την αναπαραγωγή του μουσικού έργου καθώς και τη μεταβολή του τέμπο του:



Εικόνα 4-4: Η γραφική διεπαφή της μεταβολής τέμπο

Το πεδίο εισαγωγής νέου τέμπο καθώς και τα κουμπιά της αναπαραγωγής και της επιβεβαίωσης μεταβολής τέμπο υλοποιήθηκαν σε HTML. Συγκεκριμένα, στο αρχείο του κώδικα της ιστοσελίδας του VHV προστέθηκε ένα πλαίσιο div με το πεδίο εισαγωγής τέμπο και το κουμπί Change!.

Θα πρέπει να αναφερθεί ότι στα αρχεία Humdrum, η πληροφορία του τέμπο είναι κωδικοποιημένη ως συμπληρωματική ετικέτα (ξεκινά με μονό αστερίσκο). Αποτελείται από έναν ακέραιο αριθμό, του οποίου προηγούνται οι χαρακτήρες ASCII “MM”. Στην Εικόνα 4-5 απεικονίζεται ένα παράδειγμα κωδικοποίησης τέμπο σε αρχείο Humdrum:

```

7  **kern      **kern      **dynam
8  *staff2    *staff1    *staff1/2
9  *Icamba   *Icamba      *
10 *I".10 (K.84) *I".10 (K.84)
11 *>[A,A1,A,A2,B,B1,B,B2] *>[A,A1,A,A2,B,B1,B,B2]
12 *>norep[A,A2,B,B2] *>norep[A,A2,B,B2]
13 *>A      *>A      *>A
14 *clefF4  *clefG2    *
15 *k[b-e-a-] *k[b-e-a-] *
16 *M3/4    *M3/4      *
17 *MM152   *MM152    ← Τέμπο: 152bpm
18 =1-     =1-     =1-
19 4CC 4C   8r      f
20 .       8c'L   .
21 4r      8e-'   .
22 .       8g'    .
23 4r      8cc'   .
24 .       8ee-'J  .

```

Εικόνα 4-5: Κωδικοποίηση του τέμπο σε αρχείο Humdrum (βελάκι)

Ο κώδικας που αφορά τις λειτουργικότητες των γραφικών αυτών στοιχείων υλοποιήθηκε σε Javascript. Αρχικά, προστέθηκε στον κώδικα του VHV ένα γεγονός onload που πυροδοτείται κάθε φορά που φορτώνεται ένα νέο αρχείο Humdrum:

```
getAceEditor().getSession().on('change', function() {
```

Το γεγονός καλεί μια συνάρτηση, στο πρώτο τμήμα της οποίας πραγματοποιείται η εξαγωγή του τρέχοντος τέμπο. Αυτό γίνεται σε δύο βήματα. Στο πρώτο ανακτάται το τρέχον αρχείο Humdrum και αποθηκεύεται ως συμβολοσειρά στη μεταβλητή “kernFile”:

```
let kernFile = getAceEditor().getSession().getValue();
```

Στο δεύτερο βήμα πραγματοποιείται πρώτα η εύρεση του μοτίβου “MM” στο αρχείο Humdrum και μετά η εξαγωγή του ακέραιου που ακολουθεί. Αυτός ο ακέραιος αποτελεί το τέμπο και αποθηκεύεται στην καθολική μεταβλητή “window.TEMPO”:

```

let startIndex = kernFile.indexOf('MM') + 2;
let stopIndex = startIndex;

while( !(kernFile[stopIndex] == '\t') ) stopIndex++;
window.TEMPO = parseInt( kernFile.slice(startIndex, stopIndex) );

```

Στη συνέχεια της συνάρτησης το τρέχον τέμπο απεικονίζεται ως προκαθορισμένη τιμή στο πεδίο μεταβολής τέμπο για να είναι ορατή στο χρήστη. Το στοιχείο HTML που περιέχει το πεδίο μεταβολής τέμπο ανακτάται μέσω της ταυτότητάς του (“tempo-input”):

```
document.getElementById('tempo-input').placeholder = window.TEMPO;
```

Κάθε φορά που ο χρήστης εισάγει μια νέα τιμή τέμπο στο κατάλληλο πεδίο και πατάει κλικ στο κουμπί Change! πυροδοτείται ένα γεγονός onclick το οποίο καλεί την εξής “συνάρτηση βέλους” (“arrow function”):

```

document.getElementById('change-tempo').addEventListener('click', () => {
  let kernFile = getAceEditor().getSession().getValue();
  let newTempo = document.getElementById('tempo-input').value;

  let newKern = ( kernFile.replaceAll(/MM\d+/g, `MM${newTempo}`) ).replaceAll(/t=\[quarter\]=\d+/g,
`t=[quarter]=${newTempo}`);

  getAceEditor().getSession().setValue(newKern, 0);
})

```

Στη συνάρτηση αυτή αρχικά αποθηκεύεται το τρέχον αρχείο Humdrum στη μεταβλητή “kernFile”. Στη συνέχεια ανακτάται το νέο τέμπο που εισήγαγε ο χρήστης και αποθηκεύεται στη μεταβλητή “newTempo”. Με μεθόδους επεξεργασίας συμβολοσειράς, πραγματοποιείται αναζήτηση του μοτίβου “MM[ακέραιος]” στη μεταβλητή “kernFile” και αντικατάστασή του ακέραιου. Τελικά το τροποποιημένο αρχείο Humdrum φορτώνεται στο VHV και αποδίδεται αυτόματα σε παρτιτούρα.

4.4 Υπολογισμός χρονικής αλληλουχίας

Η λειτουργικότητα αυτή, όπως προαναφέρθηκε, περιλαμβάνει τον υπολογισμό του χρόνου στον οποίο αντιστοιχεί κάθε επιλεγμένο μουσικό συμβάν από την έναρξη του μουσικού έργου. Στόχο έχει να διευκολύνει διαδικασίες συγκριτικής ανάλυσης διαφόρων παραμέτρων μιας προ-ηχογραφημένης ερμηνείας του μουσικού έργου με τη MIDI εκδοχή του. Σε αντίθεση με τις προηγούμενες επεκτάσεις, εδώ δεν υφίσταται γραφικό τμήμα. Ο κώδικας υλοποιήθηκε αποκλειστικά σε Javascript.

Ο υπολογισμός του χρόνου πραγματοποιείται βάσει της ακόλουθης σχέσης:

$$t = (TSN * CMN) / T$$

όπου t ο χρόνος σε sec, TSN (Time Signature Nominator) ο αριθμητής του χρονοδείκτη του μουσικού έργου, CMN (Current Measure Number) ο αριθμός του επιλεχθέντος μέτρου και T το τέμπο της παρτιτούρας σε bpm.

Στα αρχεία Humdrum η πληροφορία του χρονοδείκτη κωδικοποιείται με το μοτίβο “MM[ακέραιος]/[ακέραιος]”. Ένα παράδειγμα κωδικοποίησης χρονοδείκτη σε αρχείο Humdrum απεικονίζεται στην Εικόνα 4-6:

8	**kern	**kern	**kern	**kern
9	*ICvox	*ICvox	*ICvox	*ICvox
10	*Ibass	*Itenor	*Ialto	*Isoprno
11	*I"Bass	*I"Tenor	*I"Alto	*I"Soprano
12	*>[A,A,B]	*>[A,A,B]	*>[A,A,B]	*>[A,A,B]
13	*>norep[A,B]	*>norep[A,B]	*>norep[A,B]	*>norep[A,B]
14	*>A	*>A	*>A	*>A
15	*cleff4	*cleffv2	*cleff2	*cleff2
16	*k[f#c#g#]	*k[f#c#g#]	*k[f#c#g#]	*k[f#c#g#]
17	*A:	*A:	*A:	*A:
18	*M4/4	*M4/4	*M4/4	*M4/4
19	*met(c)	*met(c)	*met(c)	*met(c)
20	*MM100	*MM100	*MM100	*MM100
21	8AL	4c#	4e	4a
22	8G#J	.	.	.
23	=1	=1	=1	=1
24	4F#	4c#	4f#	4a
25	4C#	8c#L	4e	4a
26	.	8BJ	.	.
27	4D	8AL	4f#	4a
28	.	8G#J	.	.
29	4D#	4F#	.	4b

Εικόνα 4-6: Κωδικοποίηση του χρονοδείκτη και του 1ου μέτρου σε αρχείο Humdrum

Η εξαγωγή του χρονοδείκτη του μουσικού έργου πραγματοποιείται αμέσως μετά την εξαγωγή του τέμπο με το εξής τμήμα κώδικα:

```
startIndex = kernFile.indexOf('M') + 1;
stopIndex = startIndex;

while( !(kernFile[stopIndex] == '\t') ) stopIndex++;
window.BEATSPERMEASURE = parseInt(timeSignatureText[0]);
```

Ο κώδικας αυτός εμπεριέχεται στη συνάρτηση που καλείται με την ενεργοποίηση του γεγονότος που αφορά την φόρτωση ενός νέου αρχείου Humdrum. Στη συμβολοσειρά "kernFile", αναζητάται η πληροφορία που κωδικοποιεί το χρονοδείκτη βάσει του ακόλουθου μοτίβου: M[ακέραιος]/[ακέραιος]. Στη συνέχεια, ανακτάται ο πρώτος ακέραιος, δηλαδή ο αριθμητής του κλάσματος του χρονοδείκτη, και αποθηκεύεται στην καθολική μεταβλητή "window.BEATSPERMEASURE".

Σε αντίθεση με τις προηγούμενες δύο παραμέτρους που εξάγονται όταν κάποιο αρχείο Humdrum φορτωθεί στην εφαρμογή, η διαδικασία εύρεσης του αριθμού του μέτρου ενεργοποιείται κάθε φορά που ο χρήστης επιλέγει κάποιο γραφικό στοιχείο της παρτιτούρας (πχ. νότα). Γενικά, ο αριθμός του μέτρου σε αρχεία Humdrum κωδικοποιείται με έναν ακέραιο, του οποίου προηγείται ο χαρακτήρας "=", όπως καταδεικνύεται στην Εικόνα 4-6.

Η εξαγωγή του αριθμού του μέτρου που έχει επιλεχθεί πραγματοποιείται μέσω του εξής κώδικα:

```
let currentLine = editor.session.getLine(row-1);
let kernText = editor.session.getValue();

if (currentLine[0] == '=') {
    window.MEASURENO = parseInt(currentLine.match(/\d+/)[0]);
}
```

```

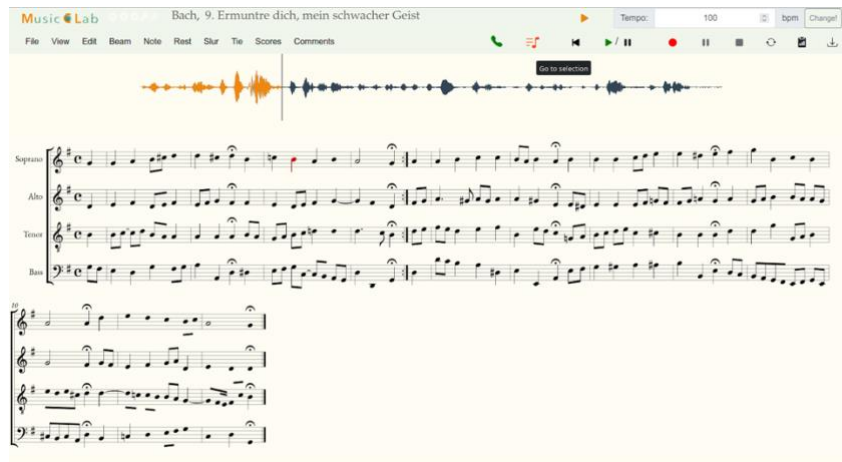
} else {
  let currentLineIndex = kernText.indexOf(currentLine);
  let matches = kernText.slice(0,currentLineIndex).match(/t=\d+/g);

  if(!matches) {
    window.MEASURENO = 1;
  } else {
    let lastMatch = matches[matches.length - 1];
    window.MEASURENO = parseInt(lastMatch.match(/d+/)[0]);
  }
}

```

Ο κώδικας προστέθηκε στην προϋπάρχουσα συνάρτηση `highlightIdInEditor` που καλείται όταν πυροδοτηθεί το γεγονός του πατήματος κλικ μέσα στην παρτιτούρα. Γενικά, η λειτουργία ανάκτησης του επιλεγμένου μέτρου βασίζεται σε έναν μηχανισμό που ενεργοποιείται όταν επιλέγεται ένα γραφικό στοιχείο στην παρτιτούρα και είναι ήδη ενσωματωμένος στην εφαρμογή του VHV. Δηλαδή, με την επιλογή στοιχείου, ένας κέρσορας μετακινείται εντός του αρχείου `Humdrum` στο σημείο που κωδικοποιεί το στοιχείο αυτό. Μάλιστα, στην περίπτωση που έχει επιλεγεί το γραφικό στοιχείο του πεντάγραμμου, η γραμμή στην οποία μετακινείται ο κέρσορας είναι εκείνη στην οποία δηλώνεται η έναρξη του μέτρου (πχ γραμμή 23 στην Εικόνα 4-6) και ξεκινά με το χαρακτήρα “=”. Στις υπόλοιπες περιπτώσεις ο κέρσορας μετακινείται σε γραμμή που βρίσκεται μετά τη γραμμή έναρξης. Με τον κώδικα που προστέθηκε, αρχικά πραγματοποιείται ανάκτηση της γραμμής στην οποία έχει μετακινηθεί ο κέρσορας και αποθήκευσή της στη μεταβλητή “`currentLine`”. Στη συνέχεια ανακτάται το τρέχον αρχείο `Humdrum` (“`kernText`”). Στην περίπτωση που η μεταβλητή “`currentLine`” ξεκινά με “=” (σκέλος “`if`” της εντολής “`if-else`”), πραγματοποιείται απευθείας η ανάκτηση του ακέραιου που ακολουθεί το “=”, δηλαδή του αριθμού του τρέχοντος μέτρου. Ο ακέραιος αποθηκεύεται σε μια καθολική μεταβλητή με το όνομα “`window.MEASURENO`”. Σε όλες τις υπόλοιπες περιπτώσεις (σκέλος “`else`” της εντολής “`if-else`”), η γραμμή δήλωσης του μέτρου προηγείται της γραμμής μετακίνησης του κέρσορα. Βάσει αυτού του γεγονότος, η ανάκτηση του αριθμού του μέτρου πραγματοποιείται με μεθόδους επεξεργασίας συμβολοσειράς.

Η λειτουργικότητα υπολογισμού της χρονικής ακολουθίας χρησιμοποιείται από άλλες επεκτάσεις του VHV, οι οποίες υλοποιούνται σε άλλες δράσεις του έργου `MusiCoLab` οι οποίες μπορούν να διαβάσουν την τιμή των καθολικών μεταβλητών “`window.TEMPO`” “`window.BEATSPERMEASURE`” και “`window.MEASURENO`”. Για παράδειγμα μια τέτοια επέκταση αφορά στο συγχρονισμό μιας παρτιτούρας με την ηχογράφηση της αναπαραγωγής του μουσικού, όπως αποτυπώνεται στην Εικόνα 4-7.



Εικόνα 4-7: Στιγμιότυπο της εφαρμογής συγχρονισμού παρτιτούρας-ηχογράφησης

4.5 Διάθεση κώδικα στο GitHub

Ο κώδικας του VHV με τις επεκτάσεις διατίθεται σε ένα δημόσιο αποθετήριο στην ιστοσελίδα του GitHub στη διεύθυνση <https://github.com/b1m1n1/vhv.git>. Στο αποθετήριο παρέχονται σαφείς οδηγίες για την εγκατάσταση της εφαρμογής.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

5.1 Σύνοψη

Στο πλαίσιο της παρούσας εργασίας, αρχικά αναλύθηκαν και συγκρίθηκαν οι μορφότυποι αρχείων μουσικής σημειογραφίας MIDI, Humdrum, ABC, MusicXML, MEI, ώστε να προσδιοριστούν οι δυνατότητες κωδικοποίησης μουσικής πληροφορίας που παρέχουν. Στη συνέχεια, παρουσιάστηκαν οι προγραμματιστικές βιβλιοθήκες Music21, Partitura, JMusic, Vexflow, Veronio και οι εφαρμογές Musescore, Lilypond, Veronio Humdrum Viewer. Καθώς το σύνολο αυτών έχει χρησιμοποιηθεί ευρέως για σκοπούς γραφικής και ηχητικής απόδοσης μουσικής σημειογραφίας, η παρουσίαση τους έγινε με στόχο τη διερεύνηση των διαφόρων λειτουργικότητων που προσφέρουν. Τέλος, πραγματοποιήθηκαν κάποιες επεκτάσεις στην ανοικτού κώδικα εφαρμογή Veronio Humdrum Viewer, οι οποίες επαυξάνουν την αλληλεπιδραστικότητα της εφαρμογής προσφέροντας χρήσιμες λειτουργίες για τη μουσική εκπαίδευση. Αντικείμενο των εργαλείων που προστέθηκαν είναι η γραφική επεξεργασία των συγχορδιών μιας παρτιτούρας, η ολοκληρωμένη ηχητική απόδοση και η μεταβολή ταχύτητας αναπαραγωγής ενός μουσικού έργου, καθώς και ο υπολογισμός χρονικής αλληλουχίας μουσικών συμβάντων.

Η εφαρμογή ενσωματώθηκε στην πλατφόρμα μουσικής εκπαίδευσης MusiCoLab. Οι επεκτάσεις υλοποιήθηκαν σε HTML, JavaScript, CSS. Για τη στυλιστική διαμόρφωση των γραφικών στοιχείων της εφαρμογής χρησιμοποιήθηκε το Bootstrap, μια ανοικτού κώδικα πλατφόρμα CSS.

5.2 Μελλοντικές επεκτάσεις

Σε αυτή την ενότητα παρουσιάζονται κάποιες επιπρόσθετες επεκτάσεις, των οποίων η υλοποίηση θα βοηθούσε στη χρηστικότητα του εργαλείου VHV.

5.2.1 Δημιουργία παρτιτούρας

Το VHV προσανατολίζεται περισσότερο στην απόδοση/επεξεργασία, παρά στη δημιουργία παρτιτούρας, δηλαδή τη σύνθεση ενός μουσικού έργου από την αρχή. Θα μπορούσε να ενσωματωθεί μια επέκταση που παρέχει δυνατότητα δημιουργίας απλών αρχείων μουσικής σημειογραφίας, για παράδειγμα επαυξάνοντας τη γραφική διεπαφή χρήστη, ώστε να παρέχονται δυνατότητες μουσικής σύνθεσης με κουμπιά, μενού και άλλα γραφικά αντικείμενα.

Εναλλακτικά, θα μπορούσε να παρέχεται η δυνατότητα μουσικής σύνθεσης στη μορφή αρχείων ABC. Ο μορφότυπος ABC αποτελεί ίσως την κατάλληλη κωδικοποίηση για αυτό το σκοπό, αφού είναι απλός και εύκολα κατανοητός από χρήστες με περιορισμένες τεχνολογικές/προγραμματιστικές δεξιότητες. Στη συνέχεια, ο μορφότυπος ABC θα μεταγραφόταν σε κάποια από τις κωδικοποιήσεις που μπορεί να αναγνώσει το VHV, με τελικό αποτέλεσμα τη γραφική απόδοσή του.

Βάσει αυτής της επέκτασης θα διευκολυνόταν η σύνθεση μουσικής μέσω του VHV. Εφόσον η σύνθεση αποτελεί βασικό τμήμα της μουσικής εκπαίδευσης, θα μπορούσαν να

προστεθούν στο VHV πρότυπες κωδικοποιημένες φόρμες σύνθεσης, με δυνατότητα σύγκρισης τους με τις συνθέσεις των εκπαιδευόμενων. Σε επίπεδο κωδικοποίησης η σύγκριση βάσει συγκεκριμένων παραμέτρων θα αποτελούσε μια σχετική απλή διαδικασία.

5.2.2 Sequencer

Αυτή η πιθανή επέκταση θα είχε στόχο τη διεύρυνση των δυνατοτήτων ηχητικής απόδοσης μιας παρτιτούρας μέσω ενός απλού μεϊκτη, που στην περίπτωση της συμβολικής μουσικής αποδίδεται πιο σωστά με τον όρο sequencer. Ο χρήστης θα μπορούσε να επιλέγει με γραφικό τρόπο ποιες πάρτες του μουσικού έργου επιθυμεί να ακροαστεί (λειτουργία solo/mute). Επίσης, θα διευκολυνόταν ο πειραματισμός πάνω στις σχετικές εντάσεις των παρτών με πιθανό αποτέλεσμα την εξαγωγή χρήσιμων συμπερασμάτων από τους εκπαιδευόμενους σχετικά με τις σχετικές δυναμικές των οργάνων μιας μουσικής σύνθεσης.

5.2.3 Συνεργατική αλληλεπίδραση με παρτιτούρες

Μία σημαντική επέκταση του VHV που θα ενίσχυε σημαντικά τη χρησιμότητά του και θα πρόσφερε προστιθέμενη αξία στο εργαλείο αυτό αφορά στην υλοποίηση του ως συνεργατική εφαρμογή. Γενικά, ανάλογα με το βαθμό συνεργατικότητας μιας ομάδας, το συνεργατικό λογισμικό μπορεί να στοχεύει (Casalino et al, 2010):

1. στην απλή επικοινωνία της ομάδας, δηλαδή τη μη δομημένη ανταλλαγή πληροφοριών,
2. στη συνδιάσκεψη της ομάδας, δηλαδή στο δομημένο επικοινωνιακό έργο που απαιτείται για την επίτευξη κοινής στοχοθεσίας ή
3. στο συντονισμό της ομάδας, δηλαδή στη σύνθετη αλληλοεξαρτώμενη εργασία με κοινούς στόχους.

Η περίπτωση της συμμετοχικής μάθησης στα πλαίσια μουσικής εκπαίδευσης εντάσσεται στην τρίτη κατηγορία. Για παράδειγμα, στα πλαίσια ενός μαθήματος μουσικής, η δημιουργία μιας μουσικής σύνθεσης θα μπορούσε να πραγματοποιείται από ομάδα εκπαιδευόμενων, η καθεμία από τις οποίες συνθέτει μια διαφορετική πάρτη υπό ένα κοινό μουσικό θέμα. Πράξεις όπως η εναλλαγή εναρμονίσεων ή η σύνθεση των διακριτών μελωδικών φωνών απαιτούν την ύπαρξη εργαλείων συνεργατικής επεξεργασίας παρτιτούρας.

Μια πρώτη προσπάθεια της επέκτασης του VHV ως συνεργατικό λογισμικό παρέχεται από την πτυχιακή εργασία με τίτλο “Συνεργατική διαχείριση μουσικής σε πραγματικό χρόνο: Μια μελέτη περίπτωσης του Veronio και του Yjs shared editing framework”, η οποία εκπονήθηκε από το Δημήτρη Μήλιο, απόφοιτο του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ελληνικού Μεσογειακού Πανεπιστημίου. Η περαιτέρω επέκταση της συνεργατικότητας, με βάση τις προσθήκες που έγιναν στην εργασία αυτή, μπορεί να καταστήσει το VHV ένα εξαιρετικά χρήσιμο εργαλείο για τη μουσική εκπαίδευση.

6. ΑΝΑΦΟΡΕΣ

- Alexandraki, C., Akoumianakis, D., Kalochristianakis, M., Zervas, P., Kaliakatsos-Papakostas, M. and Cambouropoulos, E., 2022. MusiCoLab: Towards a Modular Architecture for Collaborative Music Learning. Web Audio Conference 2022 (WAC 2022), Cannes, France. <https://doi.org/10.5281/zenodo.6770559>
- Akoumianakis, D., Alexandraki, C., Milios, D. Nousias, A., 2022, Synchronous Collaborative Music Lessons and their digital materiality. Web Audio Conference 2022 (WAC 2022), Cannes, France. <https://doi.org/10.5281/zenodo.6768537>
- Khan, N.U., Lee, J.-C., 2014. Development of a Music Score Editor based on MusicXML. Journal of the Korea Society of Computer and Information 19, 77–90. doi:10.9708/jksci.2014.19.2.077
- Devaney, J., 2020. Using Note-Level Music Encodings to Facilitate Interdisciplinary Research on Human Engagement with Music. Transactions of the International Society for Music Information Retrieval 3, 205–217. doi:10.5334/tismir.56
- Hankinson, A., Roland, P., Fujinaga, I., 2011. The music encoding initiative as a document-encoding framework, in: Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011. pp. 293–298.
- Müller, M., 2015. Fundamentals of Music Processing, Fundamentals of Music Processing. Springer International Publishing. doi:10.1007/978-3-319-21945-5
- Huron, D., 2002. Music information processing using the humdrum toolkit: Concepts, examples, and lessons. Computer Music Journal 26, 11. doi:10.1162/014892602760137158
- Good, M., 2001. MusicXML: An internet-friendly format for sheet music. XML Conference and Expo 1–12.
- Khan, N.U., Lee, J.-C., 2014. Development of a Music Score Editor based on MusicXML. Journal of the Korea Society of Computer and Information 19, 77–90. doi:10.9708/jksci.2014.19.2.077
- Crawford, T., Lewis, R., 2016. Review: Music Encoding Initiative. Journal of the American Musicological Society 69, 273–285. doi:10.1525/jams.2016.69.1.273
- Watson, M., 2018. MuseScore. Journal of the Musical Arts in Africa.
- 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. International Society for Music Information Retrieval, pp. 107–112.
- Slovák, D., Látal, P., 2022. Main functional parts of Multilevel supporting system for special forms of learning and their usage, in: WSEAS TRANSACTIONS on ACOUSTICS and MUSIC
- Nienhuys, H.-W., Nieuwenhuizen, J., 2003. Lilypond, a System for Automated Music Engraving. Colloquium, X I V Informatics, Musical Cim, X I V 1–6.

- Cuthbert, M.S., Ariza, C., 2010. Music21: A toolkit for computer-aided musicology and symbolic music data, in: Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010. pp. 637–642.
- Grachten, M., Chacón, C.C., & Gadermaier, T., 2022. partitura: A Python Package for Handling Symbolic Musical Data. ArXiv, abs/2201.13144.
- Sorensen, A., Brown, A.R., 2000. Introducing jmusic, in: Australasian Computer Music Conference. ACMA, pp. 68–76.
- Casalino, N., Draoli, M., 2010. Governance and organizational aspects of an experimental groupware in the Italian public administration to support multi-institutional partnerships, in: Information Systems: People, Organizations, Institutions, and Technologies - ItAIS: The Italian Association for Information Systems. Physica-Verlag, pp. 81–89. doi:10.1007/978-3-7908-2148-2_11
- Μήλιος Δ., 2022. Συνεργατική διαχείριση μουσικής σε πραγματικό χρόνο: Μια μελέτη περίπτωσης του Veronio και του Yjs shared editing framework, προπτυχιακή διπλωματική, Ηράκλειο: Ελληνικό Μεσογειακό Πανεπιστήμιο


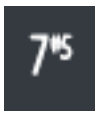

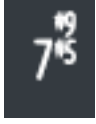


7. ΠΑΡΑΡΤΗΜΑ Α












Στον Πίνακα 7-1 παρατίθεται οι οικογένεια γραμματοσειρών που χρησιμοποιήθηκε στον Επεξεργαστή Συγχορδιών. Έχει αναπτυχθεί από την ομάδα της εφαρμογής Genius Jam Tracks¹⁴, η οποία πραγματοποίησε την αντιστοίχιση κάθε συμβόλου συγχορδίας με έναν χαρακτήρα ASCII. Αυτά τα σύμβολα συγχορδιών χρησιμοποιούνται σε συνδυασμό με τη βασική νότα της συγχορδίας και ενδεχόμενες αλλοιώσεις (δίεση ύφεση). Για παράδειγμα η συγχορδία “Ντο δίεση ελάσσονα με έβδομη” αντιστοιχίζεται στο σύμβολο C#m7.





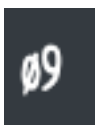







Πίνακας 7-1: Αντιστοίχιση συμβόλου συγχορδίας με τον αλφαριθμητικό χαρακτήρα.















A/A	Σύμβολο Συγχορδίας	Κωδικοποίηση	Περιγραφή
1	5	“5”	Fifth
2	6	“6”	Sixth
3	7	“7”	Seventh
4	9	“9”	Ninth
5	11	“Q”	Eleventh
6	13	“U”	Thirteenth
7	7 ^b	“S”	Seventh flat ninth








¹⁴ <https://genusiamtracks.com/>

8		"s"	Seventh sharp ninth
9		"t"	Seventh sharp eleventh
10		"p"	Seventh flat fifth
11		"q"	Seventh sharp fifth
12		"r"	Ninth sharp eleventh
13		"T"	Ninth flat fifth
14		"n"	Ninth sharp fifth
15		"2"	Ninth sharp thirteenth
16		"M"	Seventh sharp fifth sharp ninth
17		"O"	Seventh flat fifth sharp ninth
18		"N"	Seventh sharp ninth sharp eleventh

19		“P”	Seventh flat ninth sharp eleventh
20		“L”	Seventh flat ninth flat fifth
21		“J”	Seventh flat ninth sharp fifth
22		“K”	Seventh flat ninth sharp ninth
23		“I”	Seventh flat ninth flat thirteenth
24		“F”	Altered seventh
25		“I”	Thirteenth sharp eleventh
26		“u”	Thirteenth flat ninth
27		“o”	Thirteenth sharp ninth
28		“a”	Minor
29		“b”	Minor seventh

30		"h"	Minor ninth
31		"i"	Minor eleventh
32		"W"	Half diminished seventh
33		"Y"	Half diminished eleventh
34		"X"	Half diminished ninth
35		"f"	Minor chord major seventh
36		"g"	Minor chord major ninth
37		"8"	Diminished seventh
38		"E"	Diminished triad
39		"j"	Minor sixth
40		"Z"	Minor sixth ninth
41		"R"	Minor flat sixth

42		"3"	Minor thirteenth
43		"v"	Minor sharp fifth
44		"D"	Added ninth
45		"K"	Sixth ninth
46		"c"	Major seventh
47		"d"	Major ninth
48		"x"	Major seventh sharp eleventh
49		"z"	Major seventh sharp fifth
50		"y"	Major ninth sharp eleventh
51		"1"	Major seventh flat fifth
52		"e"	Major thirteenth
53		"0"	Seventh sharp ninth flat thirteenth
54		"%"	Minor added ninth
55		"4"	Suspended

56		"v"	Suspended seventh flat ninth
57		"H"	Suspended seventh added thirteenth
58		"B"	Suspended ninth
59		"C"	Suspended thirteenth
60		"w"	Suspended seventh flat thirteenth
61		"A"	Suspended seventh
62		"G"	Augmented